Advance in Electronic and Electric Engineering.
ISSN 2231-1297, Volume 3, Number 5 (2013), pp. 593-600

http://www.ripublication.com/aeee.htm

Detection of Faults Causing Data Corruption on Serial Communication Lines in MIPS-RTOS-Embedded Systems

Saurabh Kumar

Sreenidhi Institute of Science and Technology, M.Tech (DSCE),
Department of ECE, JNTU-Hyderabad
Risala Bazar, Secunderabad, A.P, India.

Abstract

Embedded Real Time application uses multi threading, a key concept of any conventional OS. The advantage of multi-threading include greater throughput, more efficient use of CPU so that it cannot remain idle for long time, better system reliability, improved performance on multiprocessor computer. The use of Real Time Operating Systems (RTOSs) became an attractive solution to simplify the design of safety-critical real-time embedded systems. Due to their demanding strict attention to rules and procedures constraints such as high-speed, low voltage operation and battery-power dependence, due to sudden up and down of voltages, flow of current these systems are often subject to transient faults. External conditions, such as Electromagnetic Interference (EMI), Heavy-Ion Radiation (HIR) as well as Power Supply Disturbances (PSD) may also cause transient faults. As the major consequence, the system's reliability degrades.

In this paper, the main focus will be on the monitoring the faults or detection of the faults on the serial communication lines that corrupt the tasks execution flow in embedded systems based on preemptive RTOS. During execution of these programs, the proposed system was exposed to conduct EMI according to the international standard IEC-61.000-4-29 for voltage transients, voltage dips and short interruptions on the serial communication lines of electronic systems. The obtained result shows that the proposed approach is able to provide higher fault coverage and reduced fault latency.

Index Terms: Hardware-Based Approach, Real-Time Operating System, Reliable Embedded System, Electromagnetic Interference (EMI).

1. Introduction

The OS meant for RTS is referred as RTOS where the time at which results are produced is of major concern. Basically, real-time systems are classified in to two types Hard and Soft real-time systems. Now days, all real time embedded systems for safety measures uses timing constraints. In general terms, real-time systems have to provide not only logically correct results [1] but also in how much time they are producing it. The necessity to adopt Real-Time Operating Systems (RTOSs) is increased as the real time systems are getting complex day by day in order to simplify the design. Though, embedded systems based on RTOSs exploit some important facilities associated to RTOSs' native intrinsic mechanisms to manage tasks, concurrency, memory as well as interrupts. In other words, RTOSs serve as an interface between software and hardware.

At the same time, the environment's always increasing hostility caused substantially by the ubiquitous adoption of wireless technologies, such as mobile telephones, represents a huge challenge for the reliability [2] of real-time embedded systems as these equipments causes' radiations and affects the behavior of any embedded system which can leads to many fatal results. In detail, external conditions, such as Electromagnetic Interference (EMI) [3], Heavy-Ion Radiation (HIR) as well as Power Supply Disturbances (PSD) [4] may cause transient faults. Currently, the consequences of transient faults represent a well-known concern in microelectronic systems. Though, embedded systems based on RTOS are subject to Single Event Upsets (SEUs) causing transient faults, which can affect the application running on embedded systems as well as the RTOS executing the application. Affecting the RTOS, this kind of fault can generate scheduling dysfunctions that could lead to incorrect system behavior.

The suggested approach in this paper is to use the Plasma MIPS architecture [10] with the combination of UCOS-II (Platform) [6] with Cyclic Check Redundancy (CRC) [5] to monitor or detect the faults affecting the tasks execution flow in embedded systems using Preemptive RTOS. To test and achieve the required performance it is run on FPGA kit [9].

2. Implementation

The implementation of this concept consists of both Hardware and Software Approach.

2.1 Hardware Approach

To evaluate the hardware based approach it consists of RTOS running on Plasma MIPS architecture [10]. The Plasma MIPS [10] is implemented in VHDL with exception of the load/store instruction, an instruction set compatible to the MIPS architecture. The Plasma RTOS uses the Preemptive scheduling algorithm with priority support composed of the following three states: *blocked*, *ready* and *executing*.

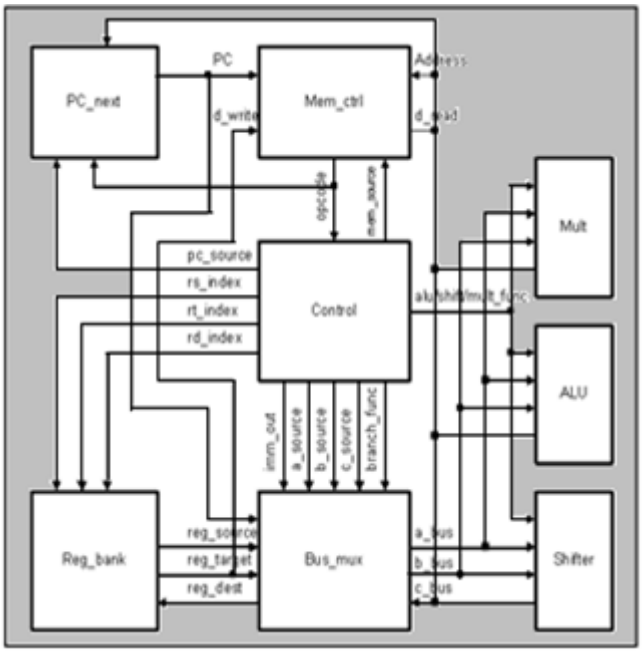

Figure 1: MIPS Processor Architecture.

The Plasma's RTOS provides a basic mechanism able to monitor the task's execution flow and manage some particular situations when faults cause misbehaviors of the RTOS's essential services, such as stack overflow and timing violations. This hardware approach i.e. Plasma MIPS architecture is linked with software approach to produce or to obtain the desire results.

2.2 Software Approach

The software approach consists of UCOS-II [6] platform with the application of monitoring or detecting the faults which causes deviation in the outputs or generates dysfunctions that could lead to incorrect system behavior using cyclic redundancy check [5]. The main operation of software approach is to detect the fault which leads to system dysfunctions.

To implement this approach GNU tool chain [7] is used to compile the ucos-ii [6] files with the application to detect the faults. From this we will get *.o file* which have to be converted in *.axf file* and now *.axf file* with the help of *convert_bin.exe* have to be converted in *.txt and .bin file* after this we have to generate the *Ram_image.vhd* from ***.txt*** and ***.bin*** using *make toimage* command.

And the complete project process flow consists of linking the ***Ram_image.vhd*** which is generated by the process flow (Software approach) with the VHDL files of Plasma MIPS architecture, and then this linked file is compiled or executed in Xilinx IDE software to generate the ***Bit Map file*** and then this generated bit map file is to be dump on SPARTAN 3E 1600 FPGA kit [9] to perform the required operations.

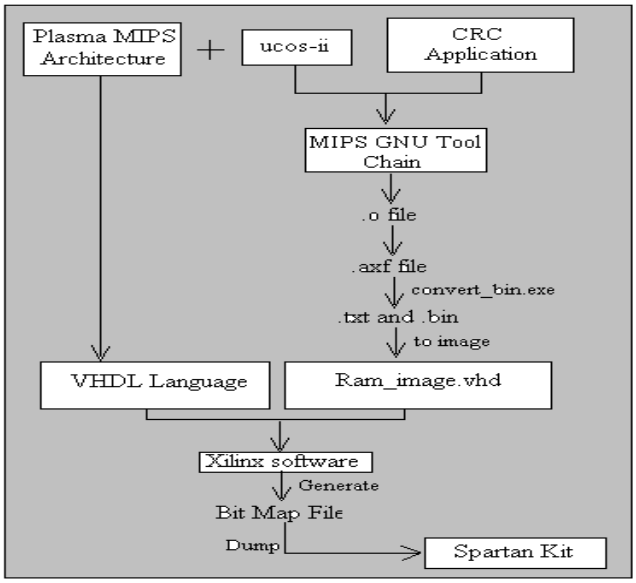

Figure 2: Project Process Flow.

3. Experimentation

SPARTAN 3E 1600 FPGA [9] development was used to run the architecture with UCOS-II [6] and test the application. Before dumping the architecture in to the FPGA the *.bin* file corresponding to the architecture is being generated and is dumped in to the board using the iMPACT software which is part of XILINX ISE .GCC tool is used for compiling the UCOS-II with application and at the same time generating the *hex file* corresponding to the required application. The operating system and application executable were loaded into the FPGA's block RAM [9] and executed from there. And then an application is to be run on architecture and the corresponding output is to be observed in HyperTerminal by means of UART.

4. Simulation Results and Outputs

As mentioned earlier, the application (CRC) has to be developed with the combination of RTOS (ucos-ii) with MIPS architecture and has to be dumped on FPGA kit [9]. The application tests the successful operation of the proposed approach and produces the required results on the HyperTerminal. As is Figure 4, there is no corruption of data on serial communication lines. Our proposed approach i.e. CRC has produced the result as NO FAULT DETECTED on the HyperTerminal.

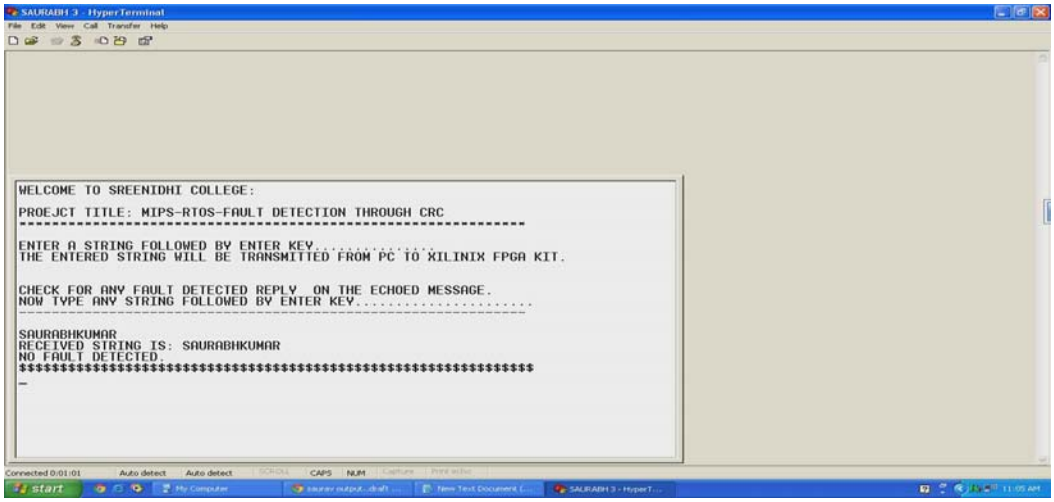

Figure 4: Serial Debug Log for Test application i.e.CRC (NO FAULT DETECTED)

As in Figure 5, we have introduced the faults using solenoid, and by generating Electromagnetic Interference (EMI) and Power supply Disturbances on serial communication lines. Our proposed approach i.e. CRC has produced the result as CRC DETECTED THE FAULT ON THE RECEIVED BYTE on HyperTerminal successfully.

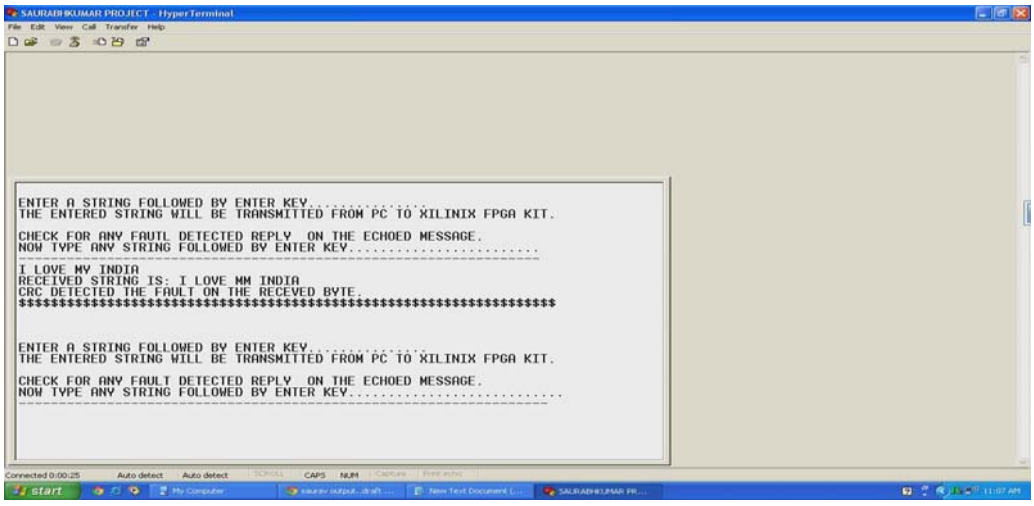

Figure 5: Serial Debug Log for Test application i.e.CRC (CRC DETECTED THE FAULT)

5. Conclusion

The main contribution of this paper consists of providing significantly more robust way of detecting the faults on serial communication lines on MIPS architecture with ucos-ii RTOS, implemented and tested on Xilinx FPGA kit [9]. In this paper we proposed a CRC approach able to detect fault occurrences on serial communication lines causing the data corruption. In general terms, the proposed approach targets transient faults affecting the data on serial bus, where it can be used for the scheduling process of the RTOS. It has been developed using cyclic check redundancy (CRC) [5] to perform on-line detection of such type of faults. The main contribution of this paper consists of drastically improving the robustness of MIPS architecture based processor and ucos-II RTOS embedded systems operating in harsh environments like those where the electronics is exposed to conducted EMI noise. The proposed approach provides nearly 100% of fault coverage.

To conclude, we are convinced that the CRC-based approach proposed herein represents an important improvement to the state-of the- art of designing hardened embedded systems running on RTOSs. Future work includes correction of errors apart from the error detection, using error correction techniques such as hamming code etc. And also this can be extended to other serial communication buses like SPI, USB etc.

References

[1] N. Ignat, B. Nicolescu, Y. Savari, G. Nicolescu, “Soft-Error Classification and Impact Analysis on Real-Time Operating Systems”, IEEE Design, Automation and Test in Europe, 2006.

[2] E. Touloupis, J. A. Flint, V. A. Chouliaras, D. D. Ward, “Study of the Effects of SEU Induced Faults on a Pipeline Protected Microprocessor”, IEEE TC, 2007.

[3] J. Freijedo, L. Costas, J. Semião, J. J. Rodríguez-Andina, M. J. Moure, F. Vargas, I. C. Teixeira, and J. P. Teixeira, “Impact of power supply voltage variations on FPGA-based digital systems performance”, Journal of Low Power Electronics, vol. 6, pp. 339-349, Aug. 2010.

[4] J. Arlat, Y. Crouzet, JU. Karlsson, P. Folkesson, E. Fuchs, G. H. Leber, “Comparison of Physical and Software-implemented Fault Injection Techniques”, IEEE Trans. on Computer, Vol. 52, N. 9, Sept., 2003.

[5] Peterson, W. W. and Brown, D. T. (January 1961). "Cyclic Codes for Error Detection". *Proceedings of the IRE* **49** (1): 228–235. doi:10.1109/JRPROC.1961.287814.

[6] Micrium Expands RTOS Family with µC/OS-II" (Press release). Micriμm, Inc. 2009-03-24. Retrieved 2010-02-14.

[7] Programming Languages Supported by GCC". GNU Project. Retrieved 2011-11-25.

[8] Denton J. Dailey (2004), *Programming Logic Fundamentals Using Xilinx ISE and CPLDs*, Pretince Hall, 203 pages. Introduction to PLDs using Xilinx ISE.
[9] Embedded Technology Journal, "Introducing the Xilinx Targeted Design Platform: Fulfilling the Programmable Imperative." Retrieved June 10, 2010.
[10] MIPS32 Architecture". MIPS Technologies. Retrieved 27 May 2009.

Author's Biography

Saurabh Kumar, Pursuing M.Tech in Digital systems and Computer Electronics (DSCE), Department of ECE Sreenidhi Institute of Science and Technology, JNTU-Hyderabad.