SAFETY CORE APPROACH FOR THE SYSTEM WITH HIGH DEMANDS FOR A SAFETY AND RELIABILITY DESIGN IN A PARTIALLY DYNAMICALLY RECONFIGURABLE FIELD-PROGRAMMABLE GATE ARRAY (FPGA)

Leso Martin, Musil Tomáš[*]

Abstract: This paper deals with a new approach to designing the micro-electronic system suitable for mass-parallel and neuronal structures realizations in which the high demand on safety and reliability is given. The presented concept is based on the FPGA platform. Authors point out various kinds of faults which can possibly occur during system cycle. Furthermore, authors introduce the Safety Core principle and define systems for which it is applicable. There are possibilities of using partial dynamic reconfiguration shown in this paper in the context of FPGA fabric testing, faults catching and correcting.

Key words: *Mass-parallel computing, neural networks, FPGA, Virtex-4, Safety Core, safety, reliability, partial dynamic reconfiguration*

Received: June 22, 2011
Revised and accepted: September 14, 2011

1. Introduction

Nowadays, there is an increasing number of designs of applications of mass-parallel and neural microelectronic structures with high demands for safety and reliability based on FPGAs. The natural parallelism of some applications directly implies using the FPGAs. But there are more reasons for using them. In comparison with microprocessor systems, which are more commonly used, FPGAs offer a simpler and regular/symmetric structure; this can become significantly profitable both in the design validation and the real time online testing. Moreover, the FPGA structure

[*] Leso Martin, Musil Tomáš
Faculty of Transportation Sciences, Czech Technical University, Prague, Czech Republic, E-mail: leso@fd.cvut.cz, musilt@fd.cvut.cz

allows to set off the resources on which the test procedure is executed during an ordinary function of the other resources or at least without paralyzing them.

Triple Modular Redundancy (TMR) or other types of the hardware redundancy are commonly used for design hardening. But hardening all the safety and reliability sensitive functions by TMR consume more fabric[1] resources, and so a bigger FPGA is necessary. If the FPGA configuration data or loaded logic structure stay without any change for a rather long time, then the probability of a fault occurrence increases and leads to the necessity of a periodical memory scrubbing. Structure degradation cannot be detected earlier than during the processing of sensitive functions. The static structure does not allow the system to get regenerated, the faults are masked by TMR.

2. The Overview of the FPGA Fabric

FPGA Structure FPGAs are based on the structure of the Configurable Logic Blocks [CLBs], the Input Output Blocks [IOBs], routing matrices and miscellaneous auxiliary components like the Block RAMs [BRAMs], multipliers, etc.

The CLBs are organized into a two-dimensional array (Fig. 1) and connected with each other by the routing matrices. The routing matrices are typically formed by lines, wires with programmable connections (switch matrix, multiplexers) of different lengths. The CLB is composed of the so-called SLICEs and then of Look Up Tables (LUTs).

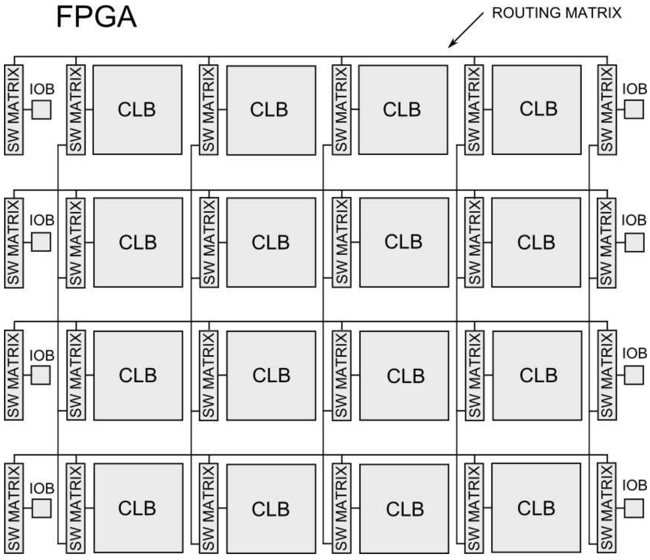

Fig. 1 *FPGA structure.*

[1] fabric: basic structure of FPGA formed by combinational logic, interconnect, input/output pins

FPGA (re)configuration The FPGA configuration is stored in the configuration memory, mostly realized by SRAM. Data stored in the configuration memory determine functions of the LUTs, the routing matrices and the auxiliary components.

The FPGA can be configurable as a whole at once or partially. It depends on an internal structure implementation, configuration memory and configuration port ability. Furthermore, the partially dynamically reconfigurable FPGAs could be reconfigured in parts while the rest of the FPGA is still running.

XILINX Virtex-4 structure – main advantages From our point of view, only the partial dynamic reconfigurable FPGAs are interesting.

Xilinx XC6200 family was the first one with this ability. And still, Xilinx provides the most appropriate features in the area of the partially dynamically reconfigurable FPGAs, and so we have decided to aim at them especially at the Virtex-4 family.

Within the Virtex-4 family a significant change in the structure and the configuration memory has been done. Due to a reduced length of the configuration memory frame it is simply possible to create a reconfigurable module with less height than only full height of CLB array, as it was possible in the old FPGA families. Particularly, the height of the Virtex-4 reconfigurable module can be a multiple of 16 CLBs. We should mention the fact that there is a possibility to create a module with a lower height in previous Xilinx FPGA families like Virtex-II, too. This technique is called “Merge partial reconfiguration”, described in [1], but this approach is not as straight as the direct using of the Virtex-4 features. The reconfigurable module width ranges from a minimum of four slices (1CLB equals 4 slices) to a maximum of the full-device width in four-slices increments.

The Virtex-4 CLBs array is divided into domains with a local clock distribution, e.g. Virtex-4 LX15 has, as the smallest family member, eight local clock domains. The symmetry of general fabric among the domains allows designing a module independent of the absolute position in the FPGA fabric [2]. Of course, the module design has to reflect fabric resources as clock domains and special resources e.g.: BRAMs, multipliers, fix cores or IOBs.

3. Types of Faults and Relevant Testing

Although the FPGAs (Xilinx FPGAs, for example) are completely tested by the manufacturer, faults of various kinds can occur during the system life cycle. Basically, there are two structures where faults can appear, it is a configuration SRAM and the FPGA fabric. The faults can be both transient and permanent.

Faults can be caused by degradation or because of an internal or external impact. Time Dependent Dielectric Breakdown, Electromigration or Thermal cycling and Stress Migration are examples of structure degradation, which can cause permanent faults.

Single Event Upset (SEU) or Single Event Transition (SET) caused by a heavy ion impact can produce faults in the configuration memory or directly in the user logic. If a bit in the configuration memory was altered, then for example LUT functions, routing matrices or clock routings could be changed. In the user logic, combinational logic can be affected and then this fault can be caught by the next Flip Flop (FF) or this FF can be directly affected. Half latches are more sensitive to SEUs than other logic. Higher temperature or a heavy ion impact can cause a latch-up[2] with or without permanent degradation (some chips are designed with high resistance to the latch-up or with the latch-up die-down ability, like Xilinx Virtex-4), this problem increases with a higher transistor density and is linked with a finer grain silicon technology.

Tests based on a different kind of fault models have been designed and presented in [3] and [4]. These tests assume symmetry of the FPGA structure. However, this hypothesis does not need to be strictly true. The FPGA structure in general is symmetric, however, in addition to this structure, FPGAs often contain another auxiliary resources. The symmetry of the structure is also broken on borders of logic blocks array or on borders of designed modules.

The static test detects faults in LUTs (the erroneously generated logic function), or in routing matrices like Permanent connection, Permanent Disconnection, Stuck-At Zero, Stuck-At One, Wire-Open. Dynamic test can detect the beginning of degradation when impedances change and delays in logic and routing matrices increase.

The dynamic test requires measuring the time delay of a particular signal and it is necessary to compare it with the reference time delay data. The reference data should be generated on the basis of a simulation and tested in more samples of the FPGA target chip within the whole range of conditions.

4. Safety Core Approach

The main scope of our usage presented FPGA reconfigurable architecture is dedicated to railway interlocking systems.

For these systems the highest safety and reliability parameters are required, but the demands for high-speed system response is not extreme – in comparison with FPGA speed ability. Particular demands are discussed in [5]. Our approach is highly advantageous for systems separable into independent functions which process input or state data and which can be utterly used in parallel.

Our approach is derived from the requirements that the system is to provide correct results as long as possible on the highest level of safety as much as possible.

We divide FPGA fabric into two parts (Fig. 2). The first one, called the Safety Core (SC), is a partial dynamic reconfigurable area. In the SC, functions with high demands for safety and reliability are located and bordered. The other part serves for maintenance functions, generally without any demands for high safety and reliability and is designed as a static area. We can presume that a smaller and a well-defined area used for SC is better testable than the whole FPGA.

After the system startup, the static part configuration bitstream is loaded. If the FPGA configuration data bitstream stays without any change for a rather long time, then the probability of a fault occurrence increases and it leads to

[2] Latch-up: parasitic PNPN structure in CMOS integrated circuit similar to thyristor which, if triggered, can cause short circuit

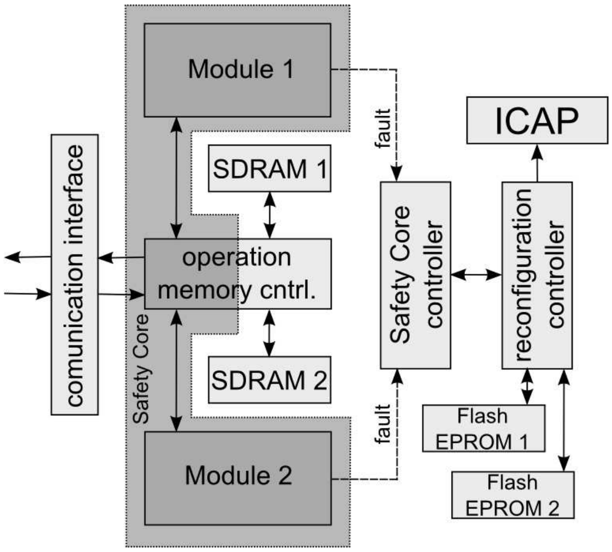

Fig. 2 *Safety Core.*

the necessity of a periodical memory scrubbing. In order to prevent faults in the configuration memory, relevant to the static part, configuration memory is periodically scrubbed.

For a higher reliability, the configuration bitstream is saved in two memories in inverse coding and secured by CRC also used by Xilinx to provide for a secure configuration bitstream. These two mirrors are compared before each configuration memory loading or scrubbing. The static part contains a module for communication with the external system outside the FPGA, a partial dynamic reconfiguration controller, an SC module controller, an SC operation memory controller and auxiliary functions, like a watchdog, etc.

We have a set of functions with high demands for safety and reliability, each designed in two different ways; it leads to two different structures and mitigation of probability of fault which can arise during design process.

In the SC, we actively use two partially reconfigurable areas, each in a separate local clock domain (e.g. Xilinx Virtex-4 FPGAs allow it) for proceeding both functions' representation simultaneously. The module controller of the SC periodically sets off so-called Time Slots (TSs). Each function from the set has the same time for an action, a part of the TS, and each of them is processed in the discrete TS. The size of the local clock domain implies the maximum time size for the function. Larger functions must be decomposed into the appropriate size and so proceed in more than only one TS.

The SC is connected to the operation memory controller. The operation memory controller is physically designed with two memories where the data are hardened by inverse coding and CRC.

Before functions are executed in the SC, data must be loaded from the operation memory and checked. Of course, check process has the same demands for safety and reliability as critical functions do, so consequently, the checking process is hardened by the same way as critical functions – proceeding in the SC. The same method is used for data comparing, deciding if they are valid and saving them into the operation memory after critical functions have been performed.

Therefore, each TS consists of a possible fabric test procedure, a data loading from the operation memory procedure, a function processing on loaded data and the saving of data back into the operation memory. Before data saving, the operation memory controller supervises the data validity. In case the data are not valid, the SC controller is aware of it. Fig. 3 shows the typical structure of two simultaneously proceed TSs.

The operation memory represents the interface for communication within the safety functions and the surroundings. From the other side of the interface, the data are loaded into the memory and read by a communication module. The communication module prevents loss of data integrity by appropriate coding and securing by CRC.

Thanks to the fact that the critical functions are designed as independent from their absolute position in the FPGA, it is possible to use another domain for critical functions processing when a permanent fault in the structure is detected, of course the FPGA must be sufficiently big to have reserve fabric in another local clock domain. A bitstream relocation technique has been mentioned in [2].

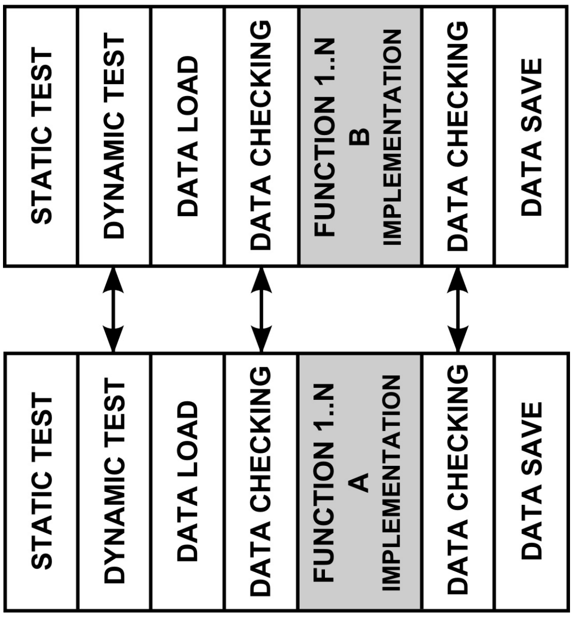

Fig. 3 *Two simultaneously proceed Time Slots in the Safety Core.*

So, we can have an alternative area for the SC where the clocks are switched off (e.g. by BUFR in Virtex-4) as a prevention of fabric degradation. When a permanent fault (structure defect) or degradation starts in the original SC domain, this alternative domain is used.

During the periodical fabric test we need to cover as many potential permanent faults as possible.

In order to ensure the reliability we use both static and dynamic tests (Fig. 3). Certain independence of the local clock domain in Virtex-4 FPGA allows us to use different domains for cross testing of the time delay. The results of the time delay gained by the test structure and of the input signals in one domain are compared with the results of the same structure and input signals in the other domain. The same conditions for both domains during this cross test are naturally guaranteed.

If a transient fault occurs in the function with high demands for safety and reliability, the "2 out of 2" system, which our design represents, should prevent the failure owing to its redundancy - in the data checking process, the way it is mentioned above.

When the fault is kept by "2 out of 2" redundancy, the type of fault must be differentiated whether it is transient or permanent (structure degradation). It is realized solely by the fabric test.

Because there are more functions with different resources requirements, periodically loaded into the SC application, independent testing is more advantageous than applying dependent tests which would cause a number of tests equal to the number of different functions.

5. Conclusion

In this paper, we have presented a new approach to the design of micro-electronic systems, suitable for mass-parallel and neuronal structures realizations with high demands for safety and reliability, which are based on FPGA platforms. The Safety Core (SC) of these structures, into which the modules with different functions are loaded by a partial dynamic reconfiguration, has been designed. This approach is of special importance for design of railway interlocking system, for which it was practically tested. However, it is in general applicable also to the other micro-electronic mass-parallel and neural systems with high requirements on reliability and operation safety (see [6-8] e.g.).

References

[1] Sedcole P., Blodget B., Becker T., Anderson J., Lysaght P.: Modular Partial Reconfiguration in Virtex FPGAs, IEE Proc.-Comput. Digit. Tech., **153**, 3, 2006.

[2] Flynn A., Gordon-Ross A., George A. D.: Bitstream Relocation with Local Clock Domains for Partially Reconfigurable FPGAs, DATE'09, Nice, 2009.

[3] Dutton B. F., Stroud C. E.: Built-In Self-Test of Embedded SEU Detection Cores in Virtex-4 and Virtex-5 FPGAs. In: Proc. ESA, 2009, pp. 149-155.

[4] Renovell M.: A Specific Test Methodology for Symmetric SRAM-Based FPGAs, Lecture Notes in Computer Science, Heidelberg, 2000.

[5] Leso M.: Redundant and combined interlocking systems and their implementations, dissertation thesis, CTU Prague, 2004, (in Czech).

[6] Votruba Z., Novak M.: Alliance approach to the modeling of interfaces in complex heterogeneous objects, Neural Network World, **20**, 5, 2010, pp. 609-619.