Safety Aware Place and Route for On-Chip Redundancy in Safety Critical Applications

Romuald Girardey, Michael Hübner, Jürgen Becker
Institut für Technik der Informationsverarbeitung – ITIV
Karlsruhe Institute for Technology (KIT), Germany
http://www.itiv.kit.edu/
{girardey, huebner, becker}@itiv.uni-karlsruhe.de

***Abstract*—This paper proposes a new solution dealing with functional safety in safety critical applications, especially with concern to the second edition of the standard IEC 61508. Actually, this new edition defines quite stringent requirements for the on-chip redundancy, such that its use in FPGAs may be compromised. Based on a previous study, which presents an on-chip coarse grained mixed-signal Triple Modular Redundancy architecture in FPGAs, this paper proposes a method to implement on-chip redundancy in FPGAs which complies with the new edition of the standard. Firstly, the paper will discuss the standard, thereafter the rules and constraints for the implementation of the on-chip redundancy, and finally it will evaluate the compliance of the method and suggest some improvements. The paper shows that the use of on-chip redundancy for FPGAs is achievable.**

I. INTRODUCTION

The first edition of the standard IEC 61508 [1] from 1998, deals with functional safety. It provides methods for the development of electric, electronic and programmable electronic functional safety-related systems to control random hardware failures or avoid systematic failures. The need of functional safety was always present. However, over the last years, this need has increased with the increased usage of electronic systems in various products of different application domains, such as within the medical, aerospace, railway, automotive, nuclear power plants and process automation industries. Of course, a failure of an electronic system in these application domains could have immense consequences on the general population and environment, which is unacceptable.

The second edition of the IEC 61508 [2] provides requirements for ASICs and FPGAs which were not present in the first edition. It provides requirements for on-chip redundancy which have a specific impact on redundancy techniques on FPGAs, such as the triple modular redundancy (TMR). This paper explains both this impact as well as an approach for the realization of on-chip redundancy technique. The goal is to implement an on-chip redundancy in an FPGA in such a way that fulfills all the requirements. Moreover, this solution should still leave the possibility to use the partial dynamic reconfiguration supported by current FPGA architectures from Xilinx. The advantages of this feature are flexibility and increased safety through the use of the self-healing process as described in [6]. This process enables failure correction during runtime.

II. BACKGROUND AND RELATED WORKS

There have been various publications in recent years dealing with fault tolerance, especially for RAM-based FPGAs which are particularly susceptible to radiation. Indeed, energetic particles can change the state of RAM cells. This could produce Single Event Upsets (SEU), i.e. a change in the configuration of the FPGA. Most of the reported approaches use a redundancy in order to avoid these failures. There are many methods for realizing a redundancy. The fine grained triple modular redundancy (TMR) (see [3], [4] and [5]) is the most commonly used. Here, the function block is integrated three times and a majority voter is used to pass the correct output.

The before mentioned approaches target the fault tolerance due to radiation effects. The standard IEC 61508 does not only consider these radiation effects, but takes into account all possible failures: random hardware failures such as radiation effects or broken transistors and interconnections; as well as the systematic failures like those done at the specification stage. In [6] an architecture designed to comply with the standard IEC 61508 ed. 1 is presented. The presented work is based on a process automation sensor. It uses a coarse grained mixed-signal diverse TMR with three diverse measuring paths: one DSP, one processor and one analog module. All these elements provide the same functionality. They compensate, calibrate and linearize a process signal. Fig. 1 shows the architecture. A majority voter is added to compare the three measuring path outputs and pass the correct one to the system output. The system also takes advantage of the ability of the FPGA to partial dynamic reconfiguration, by means of the self-healing scenario.

The redundant functional blocks used in most implementations for fault tolerance in FPGAs, as presented above, are homogenous, i.e. a multiple implementation of the

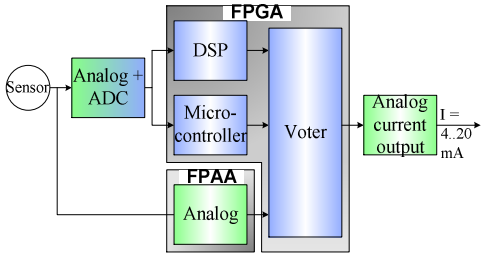

Figure 1. A coarse grained mixed-signal TMR architecture

978-0-7695-4076-4/10 $26.00 © 2010 IEEE
DOI 10.1109/ISVLSI.2010.51

same hardware block. In [6] a diverse redundancy, also called diversity, is used, i.e. a multiple implementation of different blocks which do the same work. The diversity offers the ability to detect the systematic failures. These failures are mostly made during the specification/development phase, such as coding errors. Those errors cannot be detected by a homogenous redundancy due to the fact that the errors are then identically present in the three modules. With a diverse redundancy, these coding errors will only occurs in one module since the development process of these modules was fully decoupled. With the use of an analog module the diversity will be highly increased, since not only are the technologies completely different, but also the thought processes behind them.

The approach presented in this paper is based on this architecture, but it can also be applied on all systems using redundancy in FPGAs and which have to fulfill the standard IEC61508 Ed.2.

III. THE STANDARD IEC 61508

A. First Edition

The standard IEC 61508 [1] is a basis standard, i.e. it is application independent and also referenced by other application standards. It applies to safety critical systems which incorporate electrical, electronic and programmable electronic devices. Its goal is to define measures and techniques to reduce the risk for the general population and environment, but also to reduce the economical impact of a failing device.

The standard provides many techniques and measures for the control of random hardware failures. These failures occur at random times and result from one or more possible degradation mechanisms in the hardware: an internal short-circuit of a component, a connection break, a change in value, etc. The major techniques include the diagnosis and the homogenous redundancy. The standard also presents methods for avoiding systematic failures. These failures are caused mainly by errors made during the specification and development phase, such as software coding errors. The techniques against systematic failure are mainly organizationally orientated, for example the use of a V-model during product development or complete and consistent documentation. For the control of systematic failures the diverse redundancy, which is used in the architecture described in [6], is the most suitable solution.

B. Second Edition

The second edition of the standard IEC 61508 [2] fills a major gap left by the first edition: the lack of requirements for ASICs and FPGAs. These requirements do not differ much from those for “discrete” components, i.e.: documentation, failure avoidance and failure control.

For the avoidance of systematic failures, an adapted version of the V-model shall be used during the ASIC development time. All design and simulation activities, results and tools should be documented. All tools, libraries and production procedures should be proven in use over a period of 2 years. All activities and their results should be verified by simulation, equivalence checks, timings analyses etc. Utilized soft-cores and hard-cores should be validated. A high-level design description language should be used, such as VHDL or Verilog. The standard also recommends the use of techniques for a satisfactory testability for the production phase. Test-structures should be implemented in the ASIC, such as Scan-Path or Build-in Self Test (BIST). The ASIC manufacturer should provide sufficient quality control and quality management, for example ISO 9000. A proven in use process technology should be utilized. The techniques to control random hardware failures in ASICs and FPGAs are similar to those used in other systems without ASICs. However, there are special measures, e.g. the use of hardware with an automatic check, testing using a test pattern, testing using redundant hardware or on-chip redundancy.

C. On-Chip Redundancy

A safety critical system with more than one functional safety channel, i.e. a redundant system, can be realized using one single IC semiconductor substrate; it makes use of on-chip redundancy. Annex E of the standard IEC 61508-2 Ed. 2 specifies special architectural requirements for integrated circuits with on-chip redundancy. The aim of these requirements is to decrease the impact of failures of one block on a second block of the IC. The effects can be electrical or thermal.

First of all, the standard defines that each channel and each monitoring element shall be on a separate physical block on the substrate. Each block shall have its own separate bond wires, pin-outs, inputs and outputs. The minimum distance between each block shall be sufficient to avoid short circuit and cross talk between these blocks. Since the effects of increasing temperature, e.g. due to a short circuit of an output, shall also be considered, this distance will also serve as a thermal decoupling, in order to avoid common cause failures. For this reason the standard provides a list of measures that should be applied, e.g. a temperature sensor between blocks with a shut-down function. Since the impact of a local short circuit within a logic circuit is regarded as negligible in comparison to power circuits, an analysis, simulation and testing of the effects of faults is sufficient. The power supply failures shall be controlled either by providing each block with its own power supply pins and separate wells or by using voltage monitors to detect the power supply faults. Short circuit and cross-talk at adjacent lines of different blocks shall not impact the safety function. An example of an on-chip redundancy according to the standard is shown in Fig. 2 and Fig. 3.

All these requirements have a large impact on the Integrated Circuit (IC) architecture. Additionally requirements which impact the design of the IC are provided. A safety critical system using an on-chip redundancy can only reach up to SIL3. The diagnostic coverage, i.e. the percentage of failures detected by a diagnosis, shall be at least 60%. One channel cannot be the watchdog of another, except for when the two channels are functionally diverse.

The preceding requirements are mandatory, but the standard also provides measures and techniques which can be used to decrease the β-factor (β_{IC}). This factor defines the

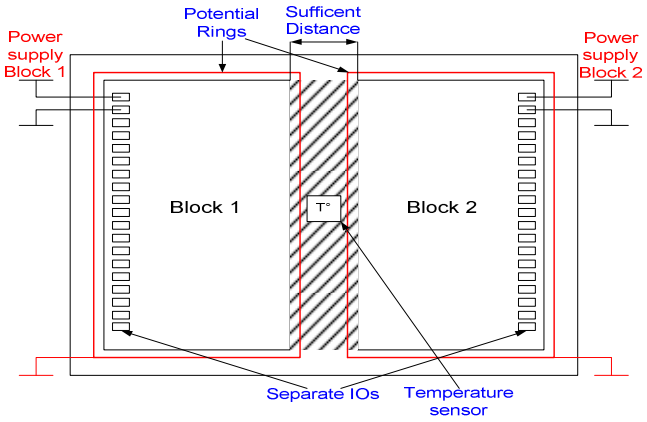

Figure 2. Example of on-chip redundancy: top view

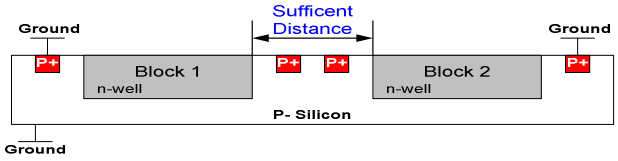

Figure 3. Example of on-chip redundancy: sectional view

susceptibility of an IC with on-chip redundancy to common cause failures, i.e. failures that are the result of one or more events, causing coincident failures of two or more separate channels. The initial β-factor ($\beta_{B\text{-}IC}$) is 33% and can be reduced by many measures and techniques such as. a high diagnostic coverage of each channel, the use of structures that isolate and decouple the channels such as potential rings, the use of temperature sensors between blocks, providing each block with its own power supply, the use of diverse redundancy, etc. The β_{IC} factor is increased by techniques and measures that increase the susceptibility to common cause failures. The final β_{IC} shall not exceed 25%.

IV. The Proposed Solution

In note 12 of annex E it is stated that: "In general these requirements do restrict the use of on-chip redundancy to full-custom ASICs and semi-custom ASICs with standard cells. Other designs such as Gate Arrays, FPGAs etc. may not allow to fulfill all requirements". In this section, it will be shown that it is possible to use an FPGA to implement an on-chip redundancy according to IEC 61508 Ed.2. The concept is to define rules and constraints for the place and route phase, so that the result complies with the standard. The presented techniques and rules are defined for Xilinx FPGAs and more precisely for the Spartan 3E or 3A, since they are architecture dependant. Not all FPGAs can be used. To utilize other FPGAs (from Xilinx or other manufacturers), the study must be remade.

A. The rules and constraints

The design is divided into blocks which are the channels of the system, diagnosis blocks and in case of the architecture described in [6], the majority voter. All function blocks are placed on the FPGA using the tool PlanAhead from Xilinx. Forbidden blocks are also defined and inserted between each defined function blocks as shown in Fig. 4. They act as a physical separation. All resources: interconnections and Configurable Logic Blocks (CLB) [7], present in the forbidden blocks are connected to ground. All communications between blocks are routed off-chip and through resistors to reduce the risk of common cause failure due to high voltage or high current in a block in fault case.

The slices [7] in the CLBs are configured such that all possible internal lines and all internal functions are also connected to ground. In the SLICEL, the Look-Up Table (LUT) is configured to output "0" for every input combination. In the SLICEM, the distributed RAM function is used, since it uses the most of the internal interconnection lines and components. The content of the RAM does not need to be set since the default value after the FPGA configuration is "0". Due to the structure of the slices, not all internal lines can be set to "0", e.g. the reset signal (SR) of the Flip-Flop (FF) needs to be "1" to reset the FF-output to "0". However the priority is to have "0" at the outputs of the slices.

In order to guarantee that no connection between two different function blocks is possible, a minimum width of the forbidden block of 7 CLBs is needed. This width is defined by considering the structure of the interconnection lines of the Spartan 3 FPGA. These lines are divided into four groups, defined by their length: the Direct Connections, the Double Lines, the Hex Lines and the Long Lines [7]. Each of them can lie either horizontally or vertically. The Direct Connection can additionally lie diagonally, and connects one CLB to its eight neighbors, i.e. two consecutive CLBs are connected together by a Direct Connection line. The Double Line connects three consecutive CLBs together and the Hex Line seven. The Long Lines span the die both horizontally and vertically and connect every sixth CLB together. With a forbidden block width of 7 CLBs it is guaranteed that all lines of the type Direct, Double and Hex, that are completely integrated into the forbidden block, are connected to ground. It is also guaranteed that there are no lines of these types which pass through the forbidden blocks. With regard to the Long Lines, it is not possible to define a forbidden block which includes it. For this reason the use of Long Lines which cross the forbidden blocks is prohibited. It is not a timing issue, as the Hex Lines are also trimmed for timing. IO Pads, block-RAMs and multipliers within the Forbidden Blocks are also completely connected to ground.

There are further lines and components to consider. The first being the Global Lines [7], which span the die like the

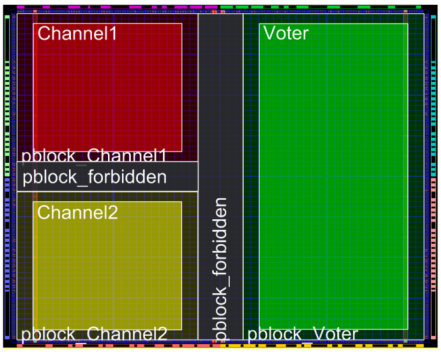

Figure 4 On-chip redundancy in FPGA

Long Lines, and the Digital Clock Managers (DCM) [7], which generates a clock signal. The Global Lines are used to transport the clock signal from the DCM to the CLBs and other components. They are also optimized to have a low skew. Due to the structure of the Global Lines and the position of the DCM in the Spartan 3E & 3A, the architecture shown in Fig. 4 must be modified as visualized in Fig. 5. The hatched area delimits the forbidden blocks. The arrows represented are the main Global Lines, which connect the DCM to the smaller Global Lines (not shown here). The function blocks are disposed in a manner such that each of them has a minimum of one DCM and one main Global Line, which are then separated from the other function blocks. The DCM and the Global Lines used by a function block is represented using the same color as the block. The form of the forbidden blocks results from the requirement to have a minimum of 7 CLBs/DCMs between two function blocks. With this configuration, the horizontal Long Lines can be used in the voter.

There are two area within this representation where two sets of Global Lines from two different function blocks are in close proximity to each other. One of them (left) is illustrated in Fig. 6. The Global Lines approaching from the left DCMs can be connected to the Global Lines running up or down. The same is possible for the Global Lines approaching from the middle which are connected to the upper middle or lower middle DCMs. The solution to having no connection between the Global Lines from different blocks is to define Global Lines in each set that are prohibited as shown in Fig. 6 (hatched area). The Global Lines coming from the right are connected to those running down, and the Global Lines coming from the left are connected to those running up. As it can be seen in Fig. 6, the Global Lines from different function blocks have no contact together.

The last interconnections and components to consider are those used for the configuration of the FPGA. These are the components used for the configuration e.g. the JTAG Boundary-Scan block [7][8] and the configuration lines which connect the configuration components to every configuration cell. Finally they are many global logic control signals, such as GSR, GTS, GHIGH_B, GRESTORE, GCAPTURE,

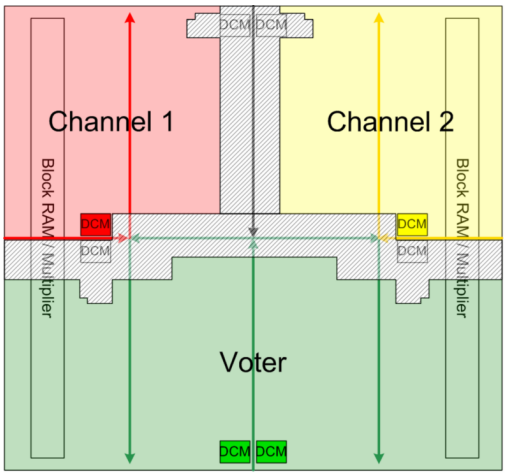

Figure 5 On-chip redundancy considering Global Lines structure

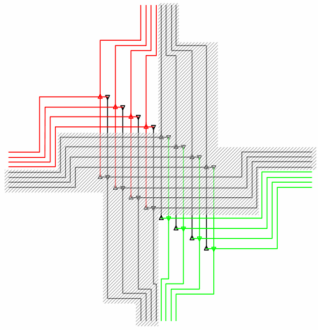

Figure 6 Global Lines interconnection in detail

DALIGN [7][8]. These signals control the state of all internal resources such as for CLBs, IOs and interconnections during start-up and configuration. All these resources are unique on the FPGA. They are not divided and distributed into the whole FPGA like the clock resources; consequentially, all function blocks are connected together. This could be an issue for the compliance with the standard IEC61508 Ed.2. In a following section, a possible improvement of current FPGAs regarding the on-chip redundancy will be discussed.

The complete design of the forbidden block is realized with the tool FPGA Editor from Xilinx. A Relationally Placed Macro (RPM) is created which is the basic part of the forbidden block. The size of the macro is 7 x 7 CLBs. The macros are combined vertically or horizontally together to realize the whole forbidden block. The macros are then connected together to ensure that no interconnection lines are left unconnected to ground. Finally the DCMs, IO pads, block RAMs and multipliers are also completely connected to ground. This process can also be automated. With regard to the Long Lines they are left unused during the Place and Route phase, whereafter they are manually connected to ground in the Forbidden Blocks with FPGA Editor.

B. Evaluation of compliance to IEC 61508 Ed. 2

As already mentioned, the forbidden blocks can be considered as a physical separation, as required by the standard. Of course, there are no different wells for each block, but this is also not mandatory. The use of different wells each with their own power supply is one appropriate measure to avoid dangerous failure caused by faults of the power supply, but it is not the only one. Another measure is the external monitoring of the power supply by use of voltage monitoring. This solution is adopted in this work.

The forbidden blocks can also be considered as a structure that isolates and decouples physical blocks as is defined in the Table E.2 of the standard. They can be considered as potential ring since they are completely connected to ground. Another requirement of the standard is that no interconnection lines from one block pass through other blocks, except for the Long Lines. However the Long Lines are not used in this approach. At least, one transistor on each side disconnects the Long Line from the function blocks, i.e. two transistor faults must occur on the same line so that two function blocks are connected together.

Of course, there are also Global Lines from two different blocks which are close together. However, as the standard states: "Short circuit and/or cross-talk at adjacent lines of different blocks shall not lead to a loss of a safety function or an undetected loss of a monitoring function (Table E.2, no. 5)"; this means that adjacent lines are allowed. The Global Lines are designed as to not produce cross-talk at full frequency and full length. Furthermore, there is always an unused Global Line, which is connected to ground, between two sets of Global Lines from different function blocks. This measure prevents short circuits between Global Lines as described for the pin-out in Table E.2, no. 5: "Ground pin between pin-out of different blocks; if not implemented, short circuit between adjacent lines of different blocks shall be carried out to test for effects of tear-off of bond wiring".

Concerning the configuration resources, the requirements of physical separated blocks and interconnects cannot be satisfied. However, these resources are used to configure the FPGA, and methods like the read-back of the configuration can be used to monitor the whole process. Furthermore, the improvement presented in the following section can also solve this possible issue. The standard specifies that the distance between two different blocks shall be sufficient to avoid short circuit and cross-talk. In an earlier version a minimum distance of 50µm was specified. The present consensus is about 100 times the structure size. With a width of 7 CLBs, which is, for example 12% of the width of a Spartan 3E 1600, the distance is far greater than 50µm or 90µm (100 times the structure: 90nm). Three temperature sensors are integrated into the forbidden blocks, between the function blocks. They have an external direct connection to the alarm output function which sets the output to a defined safe state: alarm. They are realized using an on-chip ring oscillator as described in [9].The minimum diagnostic coverage of 60% imposed by the standard is realized in the architecture described in [6].

The standard also requires the calculation of the common cause factor (β_{IC}). As already mentioned, this factor defines the susceptibility of the IC with on-chip redundancy to common cause failures, i.e. failures that are the result of one or more events, causing coincident failures of two or more separate channels. The percent values are defined in the standard. Starting with 33%, it is increase by 5% through the on-chip use of watchdogs as monitoring elements. The function blocks monitor each other; this is allowed by the standard since the function implementations are diverse. β_{IC} is then decreased by 6% due to the use of diversity in the functions and measures to control failures in different channels. This criterion comes from the architecture described in [6] which takes advantage of the diverse redundancy, and on which this work is based. The system is also designed not to be interfered by electromagnetic disturbance with an additional safety margin. This decreases the factor by 5%. In addition, the forbidden blocks and IO pins connected to ground decrease β_{IC} by 2% each and furthermore the temperature sensors decrease it by an additional 2%. The resultant β_{IC} is 21%, which is lower than the 25% imposed by the standard.

The correct implementation of the rules and methods is visually verified after the Place and Route phase with FPGA Editor. This tool simplifies the inspection of the Forbidden Blocks, since it shows all used interconnections lines. A used line connected to a Forbidden Block can therefore be easily detected. This work assumes a correct bitstream generation. However, as already mentioned the target architecture described in [6], it is a diverse triple redundancy. If an error happens during this last phase, it will be detected by the voter since the faulty block will deliver different results as the other blocks. That is to say the failures made during the bitstream generation are included in the architecture fault tolerance.

C. Evaluation of the impacts

Another aspect to consider is the heaviness of the proposed methods. Of course the width of the forbidden blocks can be an issue for small FPGAs. Nevertheless, the Spartan 3E 1600 used for the implementation of the architecture in [6], with two Microblaze and one DSP, tolerates the overhead. Certainly this method is costly in terms of critical components like DCMs and global lines, and also in terms of the CLBs. For given applications it could be necessary to switch to a larger FPGA, but this method enables the use of on-chip redundancy in FPGAs. Without it, it would be necessary to use two or three FPGAs. The restriction of the use of the Long Lines for his part restricts the use of a Microblaze, since the Microblaze uses hard-macros in order to meet the timing requirements. But, other processors can be used. Moreover, the frequency used in the implementation does not exceed 25MHz, i.e. it is realizable without the use of the Long Lines.

Regarding the toolchains, the described methods and rules are transparent for Xilinx Tools since the forbidden blocks are macros and thus either generated from a specific tool by combining the 7x7 CLB base macros or manually produced. The rest of the method relies on the standard Partial Reconfiguration (PR) toolflow. The function blocks are delimited using the constraint generated by PlanAhead: *AREA_GROUP "..." RANGE = SLICE_...:SLICE_...;*. This forces the placement and the routing of a function block and its interconnections into the specified area, preventing therewith an interconnection to be routed from one Block into another and other perverse routing.

As already stated, this work was done for the Spartan 3E/A from Xilinx. One main rule for the transfer of this work to other FPGAs is that all global resources that are needed, e.g. DCMs, Global lines, multipliers, block-RAMs, Global Lines should be distributed in as many separate places as the number of function blocks. As presented in this work, if there are three function blocks, a minimum of three places for the global resources are needed. Each global resource should be present in each a function block so far as required.

This work can be compared to other papers [10][11], both of which define a forbidden block, called respectively a fence or a moat. However the focus differs. The first difference to the present work is that this fence or moat is left unused. In this work the forbidden blocks and the forbidden lines such as the Long Lines are connected to ground. This is essential in the case of the accumulation of failures as required by the standard. A failure in one function block could connect it to the forbidden block, but it cannot connect it to other function blocks due to the grounding of the forbidden block. This stems

from Common Cause Failure. All these measures are used to approach the ideal realization defined by the standard: no transistors and lines shall be present in the forbidden blocks. One CLB wide forbidden block as defined in [10] is incompatible with the requirement to connect all lines in the forbidden block with ground. In [11] a 2 CLB wide forbidden block is defined by disabling the use of longer interconnection lines such as Hex and Long Lines. This is a compromise between performance and area. In the present work, it will be necessary not only to connect the long lines to ground but also the Hex Line. This would be much more complicated to realize. The second difference results from the fact that these papers do not consider the global lines and the configuration lines.

D. *Possible improvements of current FPGAs considering the on-chip redundancy*

One of the main improvements would be to include more than one set of configuration components and lines, and to distribute them in the FPGA, for example based on the clock resources distribution (DCM and Global Lines). The number of configurations set shall be equal to or greater than the number of function blocks to be implemented (three in the case of [6]). An improvement of the Global Lines could also be helpful. It can provide more flexibility than the presented solution, since the function block are limited with regard to the form, size and position by the structure of the Global Lines. Of course, a Function Block can only be placed in a quadrant of the FPGA and can use only one or two. By changing the tree structure of the Global Lines in a manner that all CLBs are fed from all main Global Lines, the size and the position of a Function Block could be made variable. For this, the implementation of more Programmable Interconnect Points (pips) which connect or disconnect two Global Lines together is required. Indeed, these pips will be used to separate two different function blocks. The use of pips can also enable the use of Long Lines. The implementation of one pips in every three CLBs makes the separation of two different Function Blocks possible. Of course, if a Long Lines cross a Forbidden Block, which is 7 CLBs wide, there will be a minimum of 2 pips which separate the two sides of the Long Lines. The two last improvements could be used for the configuration lines as well as for all lines which span the die. The Spartan 3 FPGA provides a VCC signal at the slice inputs, and can be connected at every input. A final improvement could be to provide a ground signal, which would be useful in the Forbidden Blocks.

All these improvements may seem unrealistic, but they could really extend the range of use of FPGAs in safety critical applications and open new markets. The first improvement which proposes to implement and distribute more configuration components and lines is the most important. It can be compared to DCM and Global Lines which are also distributed in the chip and should not dramatically increase the cost of the device.

V. REALISATION AND FUTURE WORK

The forbidden blocks were designed in a Spartan 3E XC3S1600E with the tool FPGA Editor. The next step is to implement the whole architecture as described in [6]. In parallel, contact must be made with the certification authority in order to test the acceptance of the proposed solution. This whole study was realized assuming that the representation of the FPGA from FPGA Editor coincide with the reality since of course the FPGA Editor shows a two-dimensional representation of a three-dimensional object. Two superimposed interconnections lines must be represented side by side. This representation brings with it a blurriness which is however much smaller than the relevant size of the forbidden block: 7 CLBs. The only location where this haziness could be relevant is where the main Global lines are close together. Therefore, contact will be made with Xilinx to confirm that the representation is very near to the reality.

VI. CONCLUSION

This paper shows that the use of an FPGA under certain conditions is not necessarily incompatible with the on-chip redundancy with respect to the standard IEC 61508 Ed. 2. Of course there are some constraints and limitations, and also some improvements which are needed, but the possibility to use a commercial FPGA would provide an immense advantage: no ASIC development, higher flexibility and partial dynamic reconfiguration. This approach opens the door to safety critical applications for FPGA devices.

REFERENCES

[1] International Electrotechnical Commission, “IEC 61508 First Edition: Functional Safety of Electrical/Electronic/Programmable Electronic Systems”, 1998.

[2] International Electrotechnical Commission, “IEC 61508 Second Edition: Functional Safety of Electrical/Electronic/Programmable Electronic Systems”, Committee Draft for Vote (CDV), 2008.

[3] Xilinx, “Triple Module Redundancy Design Techniques for Virtex FPGAs”, XAPP197, Jul. 2006.

[4] S. D’Angelo, C. Metra, S. Pastore, A. Pogutz, G. Sechi, “Fault-tolerant voting mechanism and recovery scheme for TMR FPGA-based systems”, Int. Symposium on Defect and Fault Tolerance in VLSI Systems, pp. 233-40, 1998.

[5] K. Paulsson, M.Hübner, M. Jung, J. Becker: "Methods for Run-Time Failure Recognition and Recovery in Dynamic and Partial Reconfigurable Systems Based on Xilinx Virtex-II Pro FPGAs ", ISVLSI2006, Karlsruhe, Germany

[6] R. Girardey, M. Hübner, J. Becker, “Dynamic Reconfigurable Mixed-Signal Architecture for Safety Critical Applications”, FPL2009, Prague, Czech Republic, 2009

[7] Xilinx, “Spartan-3 Generation FPGA User Guide - Extended Spartan-3A, Spartan-3E, and Spartan-3 FPGA Families”, UG331, Jan. 2009

[8] Xilinx, “Spartan-3 Generation Configuration User Guide - Extended Spartan-3A, Spartan-3E, and Spartan-3 FPGA Families”, UG332, March 2009

[9] S. Lopez-Buedo, J. Garrido, E. Boemo, “Thermal Testing on Reconfigurable Computers”, IEEE Design and Test of Computers, vol. 17, no. 1, pp. 84-91, Jan.-Mar. 2000

[10] M. McLean, J. Moore, “FPGA-Based Single Chip Cryptographic Solution”, Military Embedded Systems, March 2007, http://www.milembedded.com/articles/id/?2069

[11] T. Huffmire, B. Brotherton, G. Wang, T. Sherwood, R. Kastner, T. Levin, T. Nguyen, C. Irvine, “Moats and drawbridges: An isolation primitive for reconfigurable hardware based systems”, IEEE Symposium on Security and Privacy, pages 281–295, 2007