A Novel Partial Dynamic Reconfiguration Based Fault Tolerant System on Chip

Deepa Jose
Nirmala Devi
Vidhya B
KCG College of Technology, Karapakkam
(deepa.ece@kcgcollege.com)

This paper presents a reconfiguration algorithm driven intrinsic repair as a practical and efficient generic solution for Field Programmable Array Based based Software Defined Radio (SDR) systems. This research deals with the implementation of a real-time adaptive system for fault tolerant FT mission critical applications that can change automatically based on the performance and reliability constraints of the application. The hardware implementation on Virtex 6E FPGA offers greater than 40% power savings at reliability (R) values as high as 0.75 and the gain reduces to 30% for high-reliability systems (R > 0.90) while retaining similar fault-protection capabilities associated with triple modular redundancy. The PDR implementations provide an average a 14.61% improvement in terms of power and 32.6% improvement in terms of resource utilization

Keywords: Software Defined Radio; Partial Dynamic Reconfiguration; Fault Tolerant; Low Power; Reliability; Signal Processing; FPGA

1. Introduction

Software Defined Radios (SDR) are currently used for various applications such as navigation, geodetic science, atmospheric science and other space applications. Since SDR contains software functions, it can change the parameters at any time according to the situations. FPGA based SDRs are used in space missions due to the combination of their computational power and the ability to reconfigure, enabling a system to support new protocols while in orbit. For the correct function, the FPGA has to be configured by loading configuration data into its configuration memory. FPGAs are equipped with an SRAM configuration memory which allows for fast and frequent reconfiguration, but which also makes them very susceptible to SEU errors. Such an error can have serious consequences and has to be mitigated before it causes an incorrect function in the system. If the FPGA is used outside the Earth’s atmosphere or the FPGA based designs are produced in massive numbers, the probability of an SEU attack will rapidly increase. The issue of SEU mitigation is an actively researched problem, which remains somewhat unsolved.

Most real-time systems are based either on Application Specific Integrated Circuits (ASICs) or reconfigurable hardware. Modern society, has become reliant on complex SOC systems which must provide reliable service. Mission critical signal processing systems often have stringent reliability and performance requirements. Failures in these systems can result in data corruption and lower performance, leading to catastrophic failures. The reliable operation of these systems is vital, not only for mission critical applications but also for regular mass-market applications. For the sub-micron SOC systems the effects of process variations (random dopant fluctuations/sub-wavelength lithography) and lower voltage/ current threshold, make these designs susceptible to various types of faults [1,2]. Maintenance and repair are usually very expensive and time consuming for these systems. Hence, besides performance, reliability and FT have become important design metrics for these systems. Due to the higher probability of faults, there is a fundamental need for addressing reliability and design for testability (DFT) issues, while designing FT VLSI systems.

FT systems require increased capability for autonomous fault tolerance and self-adaptation at run-time, especially as system complexities and interdependencies increase. The signal processing systems should desirably have an optimal balance between performance and reliability requirement of the FT system and an automated and adaptive online flexible repair algorithm. Redundancy is the provision of the functional capabilities, that would be unnecessary in a fault-free environment. This can be backup components, which automatically ‘kick in’ should one component fail. However, the associated redundancy brings in a large number of penalties: increase in weight, size, power consumption, cost, as well as time to design, verify, and test. Hence in this research work redundancy is incorporated using partial dynamic reconfiguration (PDR). Normal hardware redundancy architectures are based on TMR or duplex systems [3-6]. The present approach is fundamentally different from the TMR, because it selects between the stages of redundancy (simplex, duplex, TMR), based on the importance of balance in performance and reliability. TMR is a wise choice in the latter case, whereas a simplex configuration is a better option in the former. This paper implements various signal processing circuits for SDR systems and intends to compare different criteria; such as size, critical path delay, power and analyze the optimum fault tolerant combination which can be used. Additionally, to control the entire error detection and recovery from various types of faults, a novel fault handling reconfiguration algorithm for FPGA based SDR systems, is proposed and implemented in this paper.

2. Fault Handling Reconfiguration Algorithm - Control Flow of Detection and Repair Process

This section describes the proposed fault handling control algorithm, whose procedure is depicted in Figure 1. A fault (SEU or stuck-at-0/1) is active when it produces an error. To achieve an FT system, the proposed fault handling algorithm implements both system failure prevention mechanisms - aimed to increase the mean time between errors (MTBE), and
module error correction methods - aimed to reduce mean time to repair (MTTR), in order to improve the availability. In our proposed algorithm shown in Figure 1, for system failure prevention, the TMR or duplex scheme with online checkers is used in order to detect and mask errors, while the corrective action takes place.

### 2.1 Module Error Correction

Assuming that the operational mode is simplex, once an error is detected, the system is changed to the duplex mode, and the recovery process for the faulty module is initiated. In the duplex mode if an online checker detects an error in a module, hardware reconfiguration is done. If an error persists, the error persistence is assumed to be due to SEU and the healing action consists in partially reconfiguring the faulty module with its corresponding configuration bit files. For example, in the duplex mode, the online checker will locate the error and initiate the error signal to the partial reconfiguration controller (PRC), which loads the bit file of the affected module. Due to this two phase recovery process, the MTTR is reduced, when compared to partially reconfiguring the bit files, each time an error is detected. In the present algorithm the bit files are reloaded, using the PDR only if a hardware reconfiguration fails to recover the fault. A watchdog timer is designed that detects the persistence of an error even after the hardware reconfiguration. In such a case, the watchdog timer generates a control signal to initiate the second recovery phase. Previously described two phase correction techniques should ensure that a faulty module will be again operative within the MTTR time interval [7, 8]. But if a second module fails before the aforementioned module is repaired, the system will fail (when MTTR>MTBE). If the before mentioned deterministic recovery strategies fail, a GA based recovery [9] can be applied to the faulty modules and the architecture is switched to the TMR mode. When a TMR fault becomes active, the voter is immediately aware of it, with mean time to detect (MTTD) = 0, since it is asynchronous and the same recovery algorithm continues until the limited number of iterations is reached. If all the strategies fail, we assume the error to be a hard error.

### 2.2 System Framework for the Proposed Fault Handling Algorithm

The system architecture, in which the fault handling algorithm has been tested together consists of two parts: the partially reconfigurable functional region (PRFR), and the static reconfiguration controller region (SRCR). The SRCR consists of the PRC incorporated with the control algorithm. Error signals from all PRMs are the input into the PRC. As the fault recovery should be initiated during the application run, a reliable tool to discover any failure immediately after its occurrence is needed. It is desirable to design circuits to be self-checking for as many faults as possible, even though more hardware is required. In case a function unit has a problem, or a fault has occurred in a function unit, the checker sends out a corresponding error signal. The checker performs the function of checking whether the output from the function unit is the correct one.

![Proposed Fault Handling Algorithm - System Self-X Properties and Transitions](image)

**Figure 1** Proposed Fault Handling Algorithm - System Self-X Properties and Transitions
3. Implementation Steps and Results of the Partial Dynamic Reconfiguration

Xilinx provides development platforms for the FPGA and SOC users. Xilinx ISE-14 Design Suite Plan Ahead supports a single environment (or platform) to manage the PDR. The methodology requires meticulous implementation to ensure success. In this work, bus macros are inserted between the modules that need to be swapped out (called PRMs) and the rest of the design (static logic). These are the channels, or ports, through which modules communicate and pass data. Synthesis guidelines are carefully followed to generate a partially reconfigurable netlist. The PRMs are floor planned and static modules are clustered. PR-specific design rules are followed. PDR design rule checks are run on the partial reconfiguration design, before implementing the configurations and generating the bit files.

The purpose of creating a black box module is to generate a blanking bit file in the bitgen step. This step includes adding a black box module to self-healing FFT_1 and FFT_2. Each reconfigurable partition must have an AREA_GROUP range constraint, to designate which physical resources are part of the reconfigurable partition. In this work, the reconfigurable partitions are FFT_1 and FFT_2. FFT_1 contains the first FFT block and its corresponding checker. FFT_2 contains the second FFT block and its corresponding checker. All physical resources not part of the AREA_GROUP range constraint associated with a reconfigurable partition are part of the static logic. Floor planning of the modules are shown in Figure 2.

The reduction in power and area with and without PDR for the various case study circuits we have implemented is studied. The percentage reduction achieved is given in Figure 3 separately for the tmr3ch and duplex2ch implementations. The PDR implementations provide on an average a 14.61% improvement in terms of power and 32.6 % improvement in terms of resource utilization over the corresponding non-PDR implementations.

3.1 Evaluation of Power Savings due to Adaptive Approach

This design results in maximum power savings under the no-fault scenario and also when the reliability requirement is less, which usually constitutes the average of an application’s lifetime. To analyze the power savings of the adaptive system over TMR, we define the variables as follows

- \( P_{FE} \) represents the dynamic power consumption of one FU and checker,
- \( P_V \) represents the dynamic power consumption of the voter,
- \( P_{TMR} \) represents the dynamic power consumption of the conventional TMR,
P_{ADAP} represents the dynamic power consumption of the adaptive system, P_s represents the power savings by using adaptive system over conventional TMR and is given by Equation (1)

\[ P_s = \frac{P_{TMR} - P_{ADAP}}{P_{TMR}} \]  

Equation (2) represents the power savings of using the adaptive over the conventional TMR with the proposed assumptions. However, if the system is assumed to be in all the three operational modes, in equal time intervals and the power savings are calculated and given by Equation (3)

\[ P_s = \frac{(2 - R)P_{PE}}{3P_{TMR}} \]

\[ P_s = \frac{R_{PV} + 2P_{PE}}{3P_{TMR}} \]

It is understood from Equation (2&3) that, if the system reliability requirement is low or average, it results in increased power gain for the adaptive system, because the system can benefit from prolonged operation under the low power profile. To experimentally qualify the analytical results, we employed the Xilinx X Power analyzer tool (XPA), to measure the dynamic power consumption for the different system components. The XPA is part of the Xilinx ISE design suite, and this has an advantage over the alternative tool, the Xilinx power estimators (XPE), which relies only on mapping reports and ignores the details of placement and routing in estimating the power consumption. The adaptive FT system obtains an advantage over the TMR in power consumption at moderately high reliability ratings as seen in Table 1. The cutoff point when the proposed adaptive system becomes advantageous in terms of power saving is until R = 0.75, which is a comparatively above average reliability value in real-life mission-critical systems, because it implies that the mission is under repair 30% of the time. The adaptive FT system offers greater than 40% power savings at R values as high as 0.75, and the gain reduces to 30% for high-reliability systems (R > 0.90) while retaining similar fault-protection capabilities to those associated with TMR. Here we can conclude that a balanced system which works without much performance degradation for lesser to above average reliability requirements is designed. However, in the previous works a better improvement in power savings is observed for higher reliability requirements alone. But high reliability requirement is not the case, for most of the working time of the system.

<table>
<thead>
<tr>
<th>Reliability</th>
<th>P_s (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1%</td>
<td>60.14</td>
</tr>
<tr>
<td>21%</td>
<td>50.32</td>
</tr>
<tr>
<td>50%</td>
<td>45.22</td>
</tr>
<tr>
<td>70%</td>
<td>40.19</td>
</tr>
<tr>
<td>90%</td>
<td>35.33</td>
</tr>
<tr>
<td>97%</td>
<td>30.33</td>
</tr>
<tr>
<td>99%</td>
<td>29.99</td>
</tr>
<tr>
<td>100%</td>
<td>28.86</td>
</tr>
</tbody>
</table>

4. Conclusion

A balanced system which works without much performance degradation for lesser to above average reliability requirements is implemented on Xilinx Virtex6E FPGA. Unlike the previous approaches the present work exhibit low power requirements during low reliability requirement also. The present study concludes that the proposed design can improve the reliability and availability parameters of the FT system, while simultaneously providing less performance overhead. The reconfiguring hardware architecture gives engineers the ability to provide adaptive self healing without the need to rework an existing design or to only design for worst-case scenarios.

5. References


