Prague Embedded Systems Workshop

June 12-13, 2014
Roztoky u Prahy,Czech Republic

Foreword

The Prague Embedded Systems Workshop is a prior student forum, which is intended for the presentation of Ph.D. and master students' research results and partial progress, including interesting practical designs and implementations, all in the field of all aspects of embedded systems design, testing and applications. The workshop is organized mostly by members of Digital Design and Dependability Research Group (DDD) from the youngest faculty (Faculty of Information Technology) of the oldest technical university in Central Europe, the Czech Technical University in Prague. The idea of PESW originated on the basis of cooperation with Tel Aviv University and the main aim is to expand the mutual collaboration in the field of embedded systems research not only inside EU.

PESW 2014 was held in Academic hotel & Congress Centre in Roztoky (the very nice place close to Prague) in June 12-13, 2014. 11 papers and one keynote were presented, rich and fruitful discussions continued during the social event and dinner. There were more than 20 participants from Izrael, Poland and Czech Republic.

This year's topics were:

- Programmable/re-configurable/adaptable architectures based on FPGA
- Architectures and hardware for security applications
- On-line and off-line error detection and correction
- Fault-tolerant control systems design methods
- Testability analysis
- Logic synthesis and optimization
- Applications of (embedded) digital systems

Last but not least we would like to thank to our sponsors (CTU in Prague, EaToN company and ASICentrum).

See you at PESW 2015!!!

Hana Kubátová and Petr Fišer

Committees

Workshop Chairs

Hana Kubátová

Petr Fišer

Programme Committee

R. Blažek, CTU in Prague, Prague (CZ)

R. Dobai, BUT, Brno (CZ)

P. Fišer, CTU in Prague, Prague (CZ)

J.-L. Gaudiot, University of California, Irvine (USA)

J. Kašpar, CTU in Prague, Prague (CZ)

H. Kubátová, CTU in Prague, Prague (CZ)

I. Levin, Tel-Aviv University (Israel)

P. Mróz, University of Zielona Gora (PL)

M. Novotný, CTU in Prague, Prague (CZ)

A. Pławiak-Mowna, University of Zielona Gora (PL)

M. Pohronská, STU, Bratislava (SK)

S. Racek, UWB, Pilsen, (CZ)

S. Ratschan, CTU in Prague and AS CR Prague (CZ)

J. Schmidt, CTU in Prague, Prague (CZ)

M. Skrbek, CTU in Prague, Prague (CZ)

W. Zając, University of Zielona Gora (PL)

Organizing Committee

H. Kubátová, CTU in Prague, Prague (CZ)

P. Fišer, CTU in Prague, Prague (CZ)

R. Kinc, AMCA (CZ)

E. Uhrová, AMCA (CZ)

Editor

J. Borecký, CTU in Prague, Prague (CZ)

Contents

Taxonomy of Research in Digital Design

ILYA LEVIN

The subject of the talk is research on field of Digital Design. A new conceptual framework for the subject is presented. According to the proposed framework, the academic studies in Digital Design are considered as a combination of aims and methods of the corresponding researches. Three main types of the research are proposed and discussed: so-called root, body and branch researchers. The root researches actually belong to the pure math; they form the Digital Design as an academic discipline. The body researches comprise fundamental studies of the Digital Design itself, enriching the discipline. The branch researches include applied engineering studies caused by and connected with emerging industrial challenges. The talk includes a number of illustrative examples of the above research types. The talk formulates a taxonomically justified approach for evaluating future directions in Digital Design studies.

Time Patterns Relation in Assisted Automated Scheduling

Krzysztof Odwrot
Department of Electrical Engineering, Computer Science and Telecommunications, University of Zielona Gora
Zielona Gora, Poland
K.Odwrot@weit.uz.zgora.pl

Wojciech Zając
Department of Electrical Engineering, Computer Science and Telecommunications, University of Zielona Gora
Zielona Gora, Poland
W.Zajac@iie.uz.zgora.pl

***Abstract* — In the paper there is discussed problem of time patterns relations in the process of assisted automated scheduling. There is described a relationship between events time description method and computational cost of scheduling process. There is presented analysis of the impact of time patterns definition to flexibility of processing algorithms. Discussion of description method effectiveness is presented and conclusion is drawn.**

Keywords — assisted scheduling, time patterns, timetabling

I. INTRODUCTION

Assisted automated scheduling and timetable creation is an important issue in various aspects of human life, because of wide need of controlling the schedule and optimization of logistic relations in time and space. It would be difficult to imagine for example efficient functioning of any logistic process or school education without existing of properly designed schedules, taking account all mutually exclusive dependences of processes and ensuring optimal usage of available resources.

Computational complexity of timetabling problem makes impossible to use the classic algorithms to solve it in polynomial time. Previous attempts of approach to automate the timetable creation process were based on various kinds of heuristics so there is a good chance to receive an acceptable result, however there is always a risk that we were unable to obtain a result, even if it exists. Furthermore every modification of base parameters causes significant changes in received result which involves necessity to create completely new timetable for every set of constraints and conditions.

As the result of requirement of matching the method of data and conditions description to work with particular algorithm there is no unified way of description allowing to define any kind of constraints and conditions. Unfortunately most of the algorithms are based on very simplified schedule models what prevents their universal application, and any change of model structure results the need of modification of the algorithm itself to handle additional dependencies. Too high level of model complexity increases the computational cost often to a level above viability point of use of such method.

The issue of study of time patterns relations in the assisted automated scheduling process and their impact on the effectiveness of the aided planning algorithm is the subject of numerous publications [1, 2, 3, 4]. In this paper there are presented results of studies on formal description methods of optimal inference path for the assumed class of applications.

II. TIME PATTERNS AND PATTERNS RELATIONS

The analysis of issue can conclude that the time patterns can be represented as the form of discrete linear functions, which allows to get a convenient description in terms of usage flexibility and reduction of calculations cost. Treating schedule as the deployment of resources and tasks in time and space it is easy to notice that the main factor affecting both the acceptability and degree of optimality of schedule is the location in the time dimension. The dominating type of events in complex, long-term schedules are cyclical events, which involve the same resources in the same or another location according to own sub-schedule. The analysis of the events time dependences including the cyclical events increases significantly complexity of the algorithm and the computational cost due to the need of treating single cyclical event as a series of individual subsidiaries events.

Let's define an Individual Event *IE* as a single and unique process definition along with definition time period of its progress and involving specific, identical resources.

Let's define a set of all possible events in the modeled system as

$$SE = \sum_{i}^{I} \sum_{j}^{Ji} \sum_{k}^{Kj} \left(IE_{x,k} + EI_{x,k} \right) \tag{1}$$

where:

SE – sum of all events in modeled system,

I – number of different Individual Events *IE*,

Ji – number of cycles in a cyclic event i,

Kj – number of Individual Events *IE* in a cycle j,

$IE_{x,k}$ – Individual Event of type x in a cycle j,

$IE_{x,k}$ – Event Interval for Individual Event of type x in a cycle j.

As one can notice, the single event location is a point in two-dimensional discrete space. Those dimensions are series of

available locations and set of time slots fulfilling conditions of particular events.

Verification of created schedule requires checking whether each single point in space corresponds to one and only one event. In case of cyclic events it is required to identify the original time windows corresponding to the definition of the cycle and then to analyze relations between every single individual event *IE*. Detection of a single conflict for any individual event in a given cyclic event means a conflict situation for this cyclic event.

In schedules based mainly on cyclic events, such a situation causes significant increase of computation complexity proportional to the number of individual elements per one cycle definition. In addition frequently one cyclic event consists of more than one sub-schedule contains separate cycle definition what results further increase of computation cost.

Definition of a single cycle of event includes the time constraints which limits the individual event execution and the event interval. The interval can be defined either as the distance between the individual events and by the coordinates based on calendar time, such as the specific day of the month. Detection of cycles conflicts is achieved by looking for common points that meet the definition of all cycles compared with each other. The comparisons are made on all cycles related to events occupying the same localization.

III. SUMMARY

As shown, the issue of modeling the time patterns used in process of assisted automated scheduling is essential in domain of support of planning and the logistics processes modeling and optimization. Proper preparation and verification of the model allows to achieve significant savings in the computational cost of realization and it also leads to shortening of time needed to achieve desired result. What's important, achieving of the desired result makes sense only if it will be available timely – exceeding that limit often makes whole results to be useless and the whole process must be started from the very beginning because of the change of input conditions.

Further research will be aimed at developing efficient way to use existing methods of description of time patterns relations to solve the problem of describing the optimal inference path for the assisted automated scheduling for assumed range of applications. A promising method seems to be an attempt to obtain a formal description of set of the time patterns connected to a single event in the form of structure, allowing application of analytical mechanisms for increased efficiency and flexibility than the previously used analysis of common points of functions defined by the time patterns. For this purpose, there are plans to use the mechanism of heuristic analysis.

REFERENCES

[1] Tim B. Cooper, Jeffrey H. Kingston: The complexity of timetable construction problems. Practice and Theory of Automated Timetabling Lecture Notes in Computer Science Volume 1153, 1996, pp 281-295

[2] Rupert Weare, Edmund Burke, Dave Elliman: A Hybrid Genetic Algorithm for Highly Constrained Timetabling Problems. Computer Science Technical Report No. NOTTCS-TR-1995-8

[3] Leo G. Kroon, Leon W. P. Peeters: A Variable Trip Time Model for Cyclic Railway Timetabling. Transportation Science Volume 37 Issue 2, May 2003, pp. 198-212

[4] Gyuri Lajos: Complete University modular timetabling using constraint logic programming. Computer Science Volume 1153, 1996, pp 146-161

Exploration Robot with Stereovision

Vladislav Richter, Miroslav Skrbek
richterv2@fit.cvut.cz, skrbek@fit.cvut.cz
Faculty of Information Technology, Czech Technical University in Prague

***Abstract*—This article presents a four-wheeled robot equipped with two cameras for stereoscopic vision. The robot is remotely controlled from a mobile device running the Android operating system. On the display, an operator sees 3D anaglyph image from stereoscopic cameras and the depth map that allows determining the distance to nearby objects in front of the robot. The robot has rich set of advanced sensors like inertial sensors, infrared low distance sensors, microphone array for sound localization, speech recognition and voice output. Wi-Fi is used for connection between the robot and the mobile device. Control commands are sent via the TCP/IP protocols to the robot. A robot control application on the mobile device has been developed for the Android operating system. This is a layered application consisting of the communication and robot control API and the graphical user interface. On start, the application retrieves a list of available commands in the XML format describing capabilities of the robot. The application can adapt its user interface according to this list. We also tested the Kinect sensor as a replacement of cameras and have developed a control application receiving all data available on Kinect.**

I. Introduction

At present the mportance of exploration robots grows. They are used in many fields especially in aerospace, waste disposal after disasters, military, police, but also in common life. Depending on deployment they vary in degree of their autonomy. For instance fully autonomous pipe explore robot is presented in [1].

Embedding 3D vision in a robot provides significant space information for a robot controller in case of fully autonomous robot, or it provides a 3D stereoscopic image to the operator. Several approaches were used for viewing 3D space in front of the robot. In [4] and [5] two cameras mounted at certain distance are used to map 3D space. Another way is to use 3D Laser Scanner as described in [3]. Tablets and mobile phones are cheap and easy to use peripherals that the operator can use to control the robot. A solution applied to a spider robot can be found in [2].

In this article we present a four-wheeled robot equipped with two cameras for stereoscopic vision. The robot is remotely controlled from a mobile device running Android operating system as shown in Fig. 1. On the display, an operator sees 3D anaglyph image from stereoscopic cameras and the depth map that allows determining the distance to nearby objects in front of the robot. The robot has rich set of advanced sensors like inertial sensors, infrared low distance sensors, microphone array for sound localization, speech recognition and voice output.

Wi-Fi is used for connection between the robot and the mobile device. Control commands are sent via the TCP/IP protocols to the robot. The sensor data including images are sent back to the mobile device and shown on its display in numerical and graphical form.

Both software and hardware of the robot is constantly improved. We also tested the Kinect sensor as a replacement of cameras. Latest version of the server running on a small notebook on the robot transmits all Kinect data including the high quality depth map to the mobile device.

II. Hardware

The four-wheeled robot has two pairs of motors, each pair is connected in parallel. The module with two pulse-width modulated H-bridges controls the left and the right side pair of motors. Wheels with 11cm diameter provide the speed of up to 60cm per second. The robot is powered by NiMH battery packs with tens cells and nominal 12V voltage.

A. Control boards

Electronic control boards are split into several layers. A PWM module and a sensor board are on the lowest level. A control board is on the intermediate level. A main board is on the top level. The computational power grows from the bottom to the top of the hierarchy. The sensor board has ATmega88 controller installed that is the heart of the odometer module. The control board, which is responsible for collecting sensor data and motor control, has 16-bit dsPIC33FJ256GP506 microcontroller. The main board is based on the Roboard RB110 which is the pentium based x86 compatible computer system with 256MB RAM running at 1GHz. The main board has Gentoo Linux installed and provides sufficient power for basic image and voice processing and communication.

B. Sensors

The robot is equipped with a rich set of sensors. The rear wheels have incremental quadrature encoders mounted on its motors. Impulses from the encoders are processed by the odometer unit that provides the distance and the velocity via the I^2C bus. Battery voltage and current sensors provide analog voltages measured by the A/D converter of the control board microcontroller. The 3-axis accelerometer and 3-axes gyroscope is mounted in the center of the robot. They allow to observe the terrain profile. The system temperature sensor, the odometer module and gyroscope share the same I^2C bus.

Near front and back obstacles are detected by two IR sensor with the range limited to 80 cm distance. They are mounted on the rear and front side of the robot chassis. For longer distance measurement, two cameras with 640x480 resolution are mounted on the top of the robot. The cameras create a

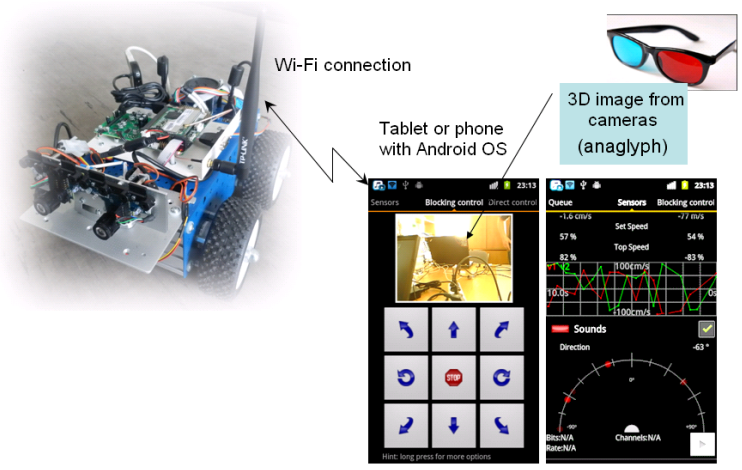

Figure 1. Exploration robot with stereoscopic vision

stereo pair that enables the depth map calculation. The cameras are rotably mounted and vertical angle is controlled by a servo drive.

The cameras have four-microphone array integrated. It enables sound source localization and can provide high quality voice for speech recognition.

C. Voice output

The robot is equipped with a USB connected sound card and speaker. In combination with Festival speech synthesizer, it provides voice output in Czech and English languages. It is an important feature for user-robot interaction and robot diagnostics because no display is mounted on.

III. Software

A. Robot side

The main board operates as a Wi-Fi access point. The server application running under the Linux operating system accepts connetions from remote control applications. For easy debugging we use textual communication protocol defining a collection of commands for wheel control (for example `set_wheel_speed`, `forward`, `backward`, `rotate`), reading camera images (`get_eye_image`, `get_depth_map`), reading status information (`get_info`). The server integrates a module for stereo vision and sound localization that process audio and video data from cameras directly attached to the main board. The server reads the sensor data from the control board which is designed as an USB CDC device and mapped as a Linux tty device. The communication protocol is mostly composed from the single letter commands decreasing the communication overhead.

Images from the cameras are processed on the robot and depth map is generated on demand the control application. Depending on algorithm and output quality selected, the depth map calculation takes from 0.3 to 3 seconds. Therefore the server is implemented as a multi-thread application to avoid unnecessary blocking of camera image and audio streams by depth map calculation. The stereo vision is based on OpenCV algorithms including an integrated camera calibration procedure.

B. Android Control Application

To remotely control the robot, three distinct application were developed. Each at a different development stage of the robot with different goal and more experience. But none of the later two applications is simply an upgrade. There are significant difference among them hence they are described separately.

All developed applications have a set of common features. They are Android based applications capable to run on small mobile devices like smartphones or tablets. Mobile devices in general provide convenient way to remotely control the robot moving around. A stationary control center seems to be very impractical in this case. Additionally the Android operating system allows to run applications on various devices. All applications shows the robot's telemetry to the operator in textual and graphical form and provides touch elements on the display to directly control the robot motors in real-time.

C. First Control Application

The earlier application is based on the Android 2.3 which is compatible with any number of devices but it cannot take advantage of advanced Android features of later versions. The structure of the application is shown in Fig. 2. This application is designed in two main layers. The robust robot's commands and data processing API (written in pure Java) are at the bottom. The data processing API provides an interface for accessing all functions of the robot. At the start this API establishes TCP/IP communication with the robot and requests a XML file describing all robot sensors and commands. After parsing this file, the API exposes appropriate interfaces to the higher GUI layer.

The GUI layer is based on Android SDK and exposed interfaces. The GUI provides 3D model of the Robot rotated

in space according to the robot's accelerometer data, current reading and history graph of the accelerometer data, gyroscope data, wheel speed, motor speed, distance from obstacles, battery voltage and current and system temperature. It can also receive and localize sound from robot's microphone and display the sound source direction on a handy display. It also receives and displays images from robot's cameras, depth map. The anaglyph provide 3D view when operator has blue-red glasses. A virtual joystick and motion keypad both drawn on the display are provided to control robot's movement. The virtual joystick is handled by user's fingers which directly sets the speed of motors in real-time. The motion keypad consists of a set of buttons representing clearly defined movements like move forward, backward, rotate left/right etc. These commands are defined at the robot side and can be chained and executed in a sequence. If a command has not a corresponding graphical control on the display, it may be also executed. The GUI shows a full list of available commands parsed form the XML file. An operator may select any command, enter its parameters and execute it.

D. Second Control Application

The second application was developed with slightly different goals. Although the earlier application has been excellent in presentation of telemetry data and very flexible for all aspects of control, it was somewhat cumbersome for remote robot navigation. A large number of display pages and control elements caused cluttered GUI and the operator was not able to move robot and see relevant data at the same time. The second application is based on Android 4.0 and expects the robot with the default configuration for the sake of simplicity. It is highly modular so that new functions may be easily added. All the control elements and telemetry information are now displayed at a single screen with the big camera image in the center. The telemetry data overlaid the image as a head-up display (HUD). Each side of the display has a slider for one thump that controls speed of one side the robot's wheels. On the top of the image, there are small icons which can switch to anaglyph display, disable HUD and perform other commands. Android 4 allows this part to be very easily editable and extended for other configurations. In order to improve the performance, the camera images received in YUY2 format are decoded directly on the graphics chip by means of the OpenGL 2.0. This approach has proved very successful since it was tested by a wide range of users with minimal experience, including small children.

E. Kinect

The user interface of the last application is based on the second one with minor improvements and different color theme. But it is differently implemented. This application has been developed from grounds up to cooperate with the robot equipped by a Kinect sensor. The Kinect sensor is now focal point of the entire application which can receive and display all Kinect sensor data including color image stream, infrared image stream, depth map stream, skeleton tracking data, audio data, and accelerometer data. For the first time this application also comes with it's own server application which runs on robot. Communication has been improved by introducing the UDP transport protocol which is more efficient for uni-directional data steams and increases throughput. Besides simple implementation it allows to stream the same data to multiple receivers at once.

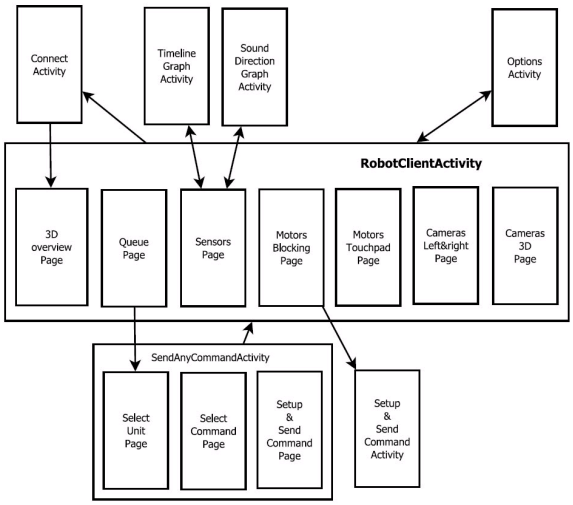

Figure 2. Internal structure of earlier application

ACKNOWLEDGMENT

Authors of this article thanks to many students who worked on hardware and software of this robot in the past and created the background for this final work. Namely we thank to Robert David, Petr Petrouš and Tomáš Klinský that implemented the video, audio and server modules as their bachelor thesis.

This research has been supported by grant
SGS12/096/OHK3/1T/18 Advanced perceptive methods for robotic platform.

REFERENCES

[1] Jong-Hoon Kim, G. Sharma, and S.S. Iyengar. Famper: A fully autonomous mobile robot for pipeline exploration. In *Industrial Technology (ICIT), 2010 IEEE International Conference on*, pages 517–523, 2010.

[2] Sung Wook Moon, Young Jin Kim, Ho Jun Myeong, Chang Soo Kim, Nam Ju Cha, and Dong Hwan Kim. Implementation of smartphone environment remote control and monitoring system for android operating system-based robot platform. In *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*, pages 211–214, 2011.

[3] J. Ryde and Huosheng Hu. 3D laser range scanner with hemispherical field of view for robot navigation. In *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*, pages 891–896, 2008.

[4] Kwang soo Kim, Wook Bahn, Changhun Lee, Tae jae Lee, M.M. Shaikh, and Kwang soo Kim. Vision system for mobile robots for tracking moving targets, based on robot motion and stereo vision information. In *System Integration (SII), 2011 IEEE/SICE International Symposium on*, pages 634–639, 2011.

[5] Zhao Yong-guo, Cheng Wei, Jia Lei, and Ma Si-le. The obstacle avoidance and navigation based on stereo vision for mobile robot. In *Optoelectronics and Image Processing (ICOIP), 2010 International Conference on*, volume 2, pages 565–568, 2010.

Comparision of various approaches in Fault-Tolerant and Attack-Resistant system design

Filip Štěpánek, Martin Novotný

Faculty of Information Technology
Czech Technical University in Prague
{filip.stepanek, martin.novotny}@fit.cvut.cz

Abstract. Fault-tolerance and attack-resistance are often discussed properties of embedded systems but are rarely achieved at the same time. The deployment of fault-tolerant systems demands some kind of reliability in hazard environment or the possibility of recovery in case of failure of the system to protect human lives or to prevent damage to property. The attack-resistant devices on the other hand protect the secrets/money or some other sensitive information of others from being misused or stolen. But as the number of attacks on software systems become more frequent and as the required education of attackers keeps decreasing, the question is – "When the safety-critical systems become target of malicious attacks?" The aim of this presentation is to discuss various fault tolerant and attack resistant system design approaches, to find common properties and to compare them to the ordinary design flow of the embedded systems. The goal of this work is to discuss the possibility of having both fault-tolerance and attack-resistance in embedded systems at the same time.

Effects of Arbitrary Hardware Faults on Multicore Scheduling in Safety-critical Applications

Evaluation by enhanced Markov models and discrete event simulation

Stefan Krämer

University of West Bohemia, Faculty of Applied Sciences
Univerzitní 8, 306 14 Plzen, Czech Republic
OTH Regensburg, Faculty of Electronics and Information Technology
Seybothstr. 2, D-93053 Regensburg, Germany
stefan.kraemer@hs-regensburg.de

Abstract

We present a discrete event simulation-based approach for reliability analysis in combination with schedulability analysis of safety-critical multicore real-time embedded systems. In such a safety-critical system the software execution does not only have to be hardened against sporadic hardware faults, e.g., by means of coded processing or symmetric redundancy, but also the real-time requirements still have to be met in the presence of such faults to guarantee a safe operation of the system. To verify the simulation environment, basic task sets that already include these safety mechanisms are evaluated by an enhanced Markov model. This Markov model is enriched by determination of timing characteristics, such as deadlines. It is shown that the behavior – regarding real-time and safety metrics – of this theoretical model can be transferred into an abstract system timing model which then can be analyzed by a discrete event simulation approach. Therefore it is possible to evaluate the influence regarding the real-time characteristics of a given sporadic fault with a certain fault rate by means of a discrete event simulation. By discrete event simulation the scope of analyzing a system can be extended compared to the Markov model. Now it is possible not only to evaluate single safety and timing metrics for a simplified functionality of the system, but also to evaluate the whole system. Currently there is no analytical approach available to proof the feasibility of global dynamic scheduling on a multicore system. The complete system model – including the mention safety mechanism and their impact on the scheduling itself – should be described. This model also includes the hardware faults that affect the system and is realized by simulated fault injection. In this work we present an approach to evaluate reliability metrics, real-time behavior and schedulability of multicores applications in a holistic view. Furthermore it is shown that the need arises to implement new safety-aware multicore scheduling algorithms.

Keywords: Markov model; Stochastic simulation; reliability analysis; real-time operating system; multicore scheduling; discrete event simulation; fault injection

Pessimistic Dependability Models Based on Hierarchical Markov Chains

Martin Kohlík and Hana Kubátová
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Email:{martin.kohlik, hana.kubatova}@fit.cvut.cz

The mission-critical systems with guaranteed levels of safety and reliability parameters are used in many different applications (e.g. aviation, medicine, space missions, and railway applications, etc.) with different impact to people and environment in case of their failure.

Such systems are composed of blocks based on different types of hardware (e.g. multi-core and many-core systems, programmable hardware like FPGA, etc.). Due to heterogeneous structure and different types of possible faults in different architectures and technologies, the realistic model, which has to be a base for necessary certifications of such systems, is mostly complicated. The state-explosion of such models leads to difficulties in construction at first, and secondly it leads to the inability to compute realistic values of dependability characteristics.

Therefore, the main aim of this paper is to propose a simplified dependability model and methods for easier dependability parameters computation, which will guarantee their required levels.

Dependability of a system is the ability to avoid *service failures* (situations where the behavior of the system deviates from the correct behavior) that are more frequent and more severe than acceptable.

Dependability is an integrating concept that includes the following attributes:

- *Safety* – absence of catastrophic consequences on the user(s) and the environment.
- *Availability* – readiness for correct service.
- *Reliability* – continuity of correct service.
- *Integrity* – absence of improper system alterations.
- *Maintainability* – ability to undergo modifications and repairs.

One of the most important techniques allowing improvement of dependability is redundancy. This means that if one part of the system fails, there is an alternate functional part. However, redundancy can have a negative impact on a system performance, size, weight, power consumption, and others.

There are many redundancy techniques including hardware, information, time, software redundancy, etc. We focus on hardware redundancy made by replication in this paper.

An event causing violation of safety of a system will be called a *hazard event*. The frequency of hazard events is called *hazard rate*.

Dependability models are models designed to calculate the hazard rate of a system. Models of complex systems consisting of cooperating dependable blocks may be created as coarse-grained or fine-grained. Coarse-grained models are small and simple models allowing exact calculations of hazard rate in a short time. On the other hand, they are inaccurate and do not reflect the internal structure of the system. Fine-grained models are accurate, but they can be too large, and thus the hazard rate calculation is time-consuming. They reflect the internal structure, but they grow rapidly in size when the complexity of a system (e.g. the number of blocks) increases.

Inexact models may be used to speed up the calculations. Accuracy is not crucial in case we prove that the inexact result is pessimistic. In other words, we must prove that the real system will be safer than the system modeled by the inexact model(s).

The presented *reduction* allows inexact pessimistic dependability models to be built. The method is based on reducing dependability models based on Markov chains. The reduced model contains one transition with one hazard rate only. This transition corresponds to a hazard event of the modeled part of the system.

The reduction of the Markov chain is the key step to calculate the hazard rate of the modeled system. It allows approximation of dependability models, so that hierarchical models can be built. The hierarchical models use multiple linked models to reflect the structure of a system. Multi-level hierarchy may be used to describe each level of redundancy independently. The hazard rates of the reduced low-level models are used in higher-level models. Higher-level models are also reduced and their hazard rates are used in top-level models.

The proposed hierarchical models allow us to

1) calculate the Safety Integrity Level (SIL),
2) determine, whether the hazard event can be tolerated/omitted safely (the hazard rate is lower than a limit value specified by SIL),
3) calculate hazard rates of systems containing multiple levels of redundancy.

Hierarchical models consisting of multiple small models

1) are easier to read/understand,
2) are easier to modify/manipulate,
3) allow the exponential number of states of the model to be avoided (the dependability parameters are calculated significantly faster).

Hardware implementation of the CloudBus protocol using FPGA

Kazimierz Krzywicki
Faculty of Electrical Engineering, Computer Science and Telecommunications, University of Zielona Gora
Zielona Gora, Poland
k.krzywicki@weit.uz.zgora.pl

Grzegorz Andrzejewski
Faculty of Electrical Engineering, Computer Science and Telecommunications, University of Zielona Gora
Zielona Gora, Poland
g.andrzejewski@iie.uz.zgora.pl

***Abstract*— In this paper, a CloudBus protocol for distributed embedded systems is presented. It provides a control mechanism for a number of processing units distributed in a network. The paper demonstrates a hardware implementation of a CloudBus protocol using FPGA. Furthermore, it considers the resource usage depending on the FPGA platform.**

***Index Terms*—Embedded Systems, Distributed Systems, FPGA, CloudBus protocol**

I. Introduction

Large and complex distributed embedded systems are difficult to design and implementation [1, 3, 6]. Moreover process synchronization in such systems are also complicated which often results in high load of the communication interfaces. CloudBus protocol [2, 7] proposed by authors allows to significant saving in the amount of the transmitted data between end modules of the distributed embedded system [4, 5] (especially when compared with the Modbus RTU or Profibus-DP protocol [7]). So far, the CloudBus protocol was implemented and tested on the distributed microcontroller platforms. This paper presents implementation of the CloudBus protocol on the FPGA platform. The implementation and the synthesis [3] was made using Hardware Description Language (HDL) – Verilog, under two different development tools: Xilinx ISE Design Suite 14.7 – synthesis to the following devices: Kintex-7 (XC7K70T), Spartan 3E (XC3S1600E), Virtex-6 (XC6VLX75T); Quartus II 13.1 – synthesis to the following devices: Arria II GX (EP2AGX45DF29C4), Cyclone IV E (EP4CE 115F23I8L), Cyclone V (5CGXFC7D7F27C8).

Comparison of the resource usage was made for the different destination devices and development tools.

In Section II CloudBus protocol is presented, Section III presents CloudBus protocol implementation with internal modules description. Section IV discuss research results for different FPGA devices. Section V, concludes the paper.

II. CloudBus Protocol

Presented CloudBus [2, 7] protocol (Fig.1) is one of the methods of the data exchange and concurrent process synchronization in the distributed systems. It realizes decentralized (distributed) control method, where all of the devices (modules), in designed system, are equal to each other. CloudBus is based on the dependence, that, data are only transferred between modules, when one of modules needs information (shared resource) from the outside of its own resource variables. Module sends (broadcasts) question to other modules about the state of the specified variable, e.g. (*if y == 1*?). The module which is responsible for this variable, sends the answer to the system, if and only if quested variable get previously quested state. This model of the communication, allows to significant savings in the amount of the transmitted data.

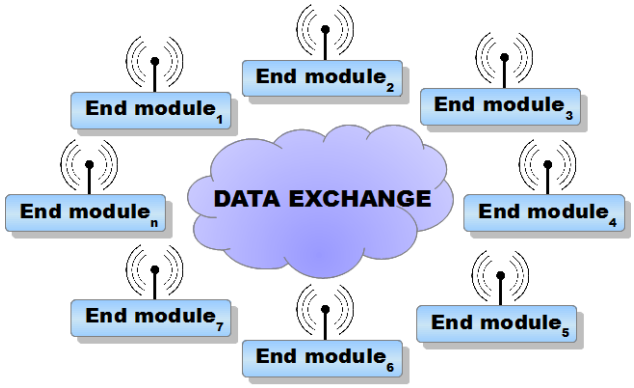

Fig. 1. Schematic diagram of the CloudBus protocol.

Basic data frame of the CloudBus protocol is shown in Table I.

TABLE I. CloudBus protocol frame

CNT	FUNC	VARS	DATA	CRC

Fields of the protocol corresponds to: *CNT* – 1 byte for entire frame length; *FUNC* – command code (e.g. question about condition or simple answer to other of the modules); *VARS* and *DATA* represents binary array of the variables and their states (values); CRC – 1 byte of the CRC error checksum.

III. Implementation

The implementation of the CloudBus protocol in the FPGA devices required dividing system functionality into dedicated modules. Designed modules are shown in Fig. 2.

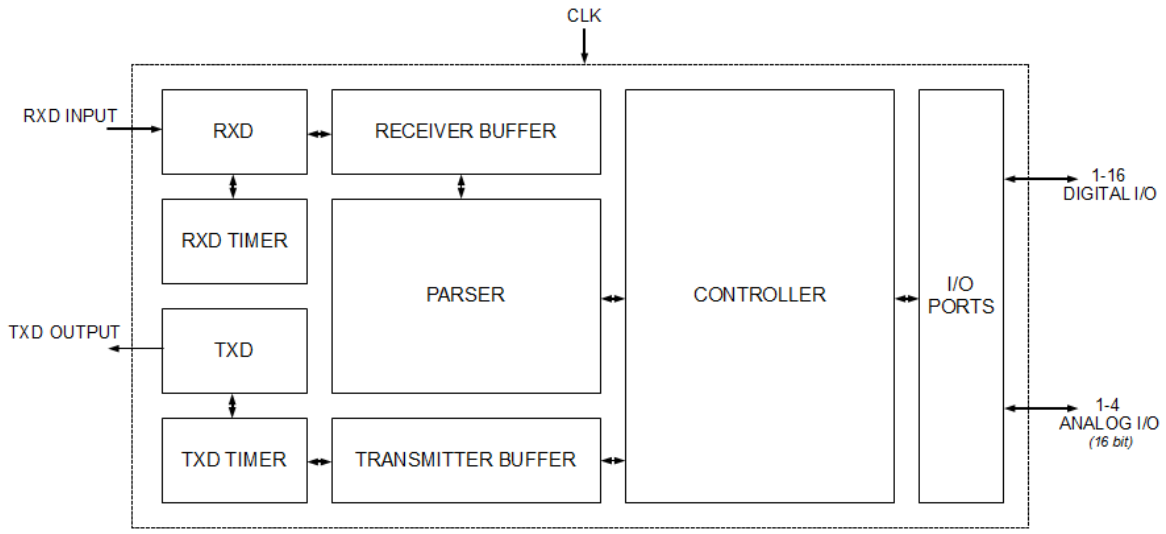

Fig. 2. Schematic diagram of the end module unit.

End module features:

- 16 digital inputs/outputs
- 4 analog inputs/outputs (16 bit)
- serial communication interface
- external clock input

A. *RXD TIMER module*

RXD TIMER module (Fig. 3) is double timer/counter. First counter (*clkHi*) with higher frequency is used for the sampling RXD INPUT line, second (slower) is used for the bit read from RXD INPUT line. *clkMain* input is external CLK clock input. *hiRst* and *loRst* are timer reset inputs.

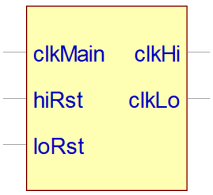

Fig. 3. Schematic diagram of the RXD TIMER module.

B. *RXD module*

RXD module (Fig. 4) is responsible for incoming communication from the outside of the end module. It retrieves data from RXD INPUT by input *RxD* line. Input line is sampled with *clkHi* clock frequency to check for incoming transmission. When it detects incoming data, *clkLo* clock counter is started for data read from *RxD* line and data bits are saved to *data* register.

Furthermore, RXD module can reset timer by loRst output or inform by received line about completed successful packet receiving. Error line, delivers information about corruption of the received data (parity/CRC checksum error).

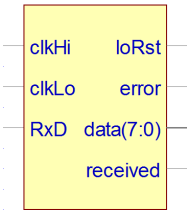

Fig. 4. Schematic diagram of the RXD module.

C. *RECEIVER BUFFER module*

RECEIVER BUFFER module (Fig. 5) is 128-bit data buffer for received data by RXD module. It merges all single bytes to entire frame of the CloudBus protocol. Module is synchronized by *clk* clock. Data incoming by 8-bit *in* input and outgoing to PARSER module by 128-bit *out* output. Additional *reset* input for resetting module and *ready* output which gives information when buffer is full.

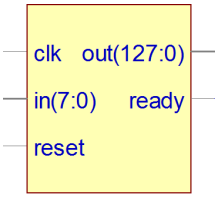

Fig. 5. Schematic diagram of the RECEIVER BUFFER module.

D. *PARSER module*

PARSER module (Fig. 6) is synchronized by *clk* input. Data are received from RECEIVER BUFFER by 128-bit *in* input. PARSRER module is responsible for parsing received frame of the CloudBus protocol from RECEIVER BUFFER. When parsing is done, without any errors (frame length check and CRC checksum check) valid CloudBus data are set

to outputs: *func*, *vars*, *digitalIO*, *anagalogIO* else *error* output is driven high, which means that received CloudBus frame is corrupted.

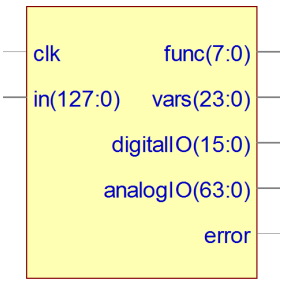

Fig. 6. Schematic diagram of the PARSER module.

E. CONTROLLER with *I/O PORT module*

CONTROLLER module (Fig. 7) is the most important module. It is responsible for implementing previously designed control algorithm and for the communication with other end modules via CloudBus protocol. CONTROLLER module is synchronized by *clk* input and it gets data (*func*, *vars*, *digitalIO*, *analogIO*) from PARSER module. *Error* input delivers information about correctness of the incoming data. Furthermore CONTROLLER module controls the 16 digital inputs/output and 4 analog inputs/outputs (16-bit each of I/O). Outside data transmission is made by *dataOut*, *sendData* outputs. *DataOut* carries data to transmit by TXD module. *SendData* triggers when data are ready to send.

For the research comparison CONTROLLER module does not perform any control algorithm. It is important, because of the different FPGAs architectures.

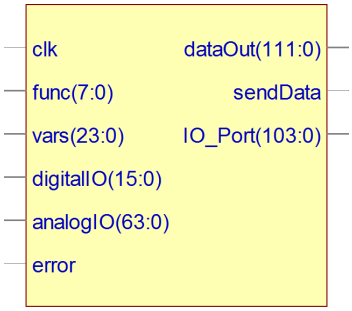

Fig. 7. Schematic diagram of the CONTROLLER module.

F. TRANSMITTER BUFFER module

TRANSMITTER BUFFER module (Fig. 8) preparing data and encoding entire frame for TXD module. Module also counts frame length and CRC checksum of CloudBus protocol frame. TRANSMITTER BUFFER is synchronized by *clk* clock, *data* input corresponds to data received from CONTROLLER module (data contains CloudBus protocol commands). *Reset* input clears TRANSMITTER BUFFER. Two outputs connected with TXD module transfers byte to send (*byteForTransmit*) and ready to send signal (*readyToSend*).

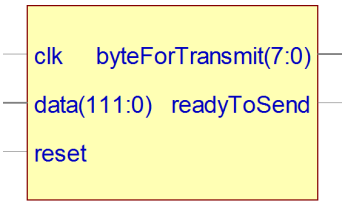

Fig. 8. Schematic diagram of the TRANSMITTER BUFFER module.

G. TXD TIMER module

TXD TIMER module (Fig. 9) generates (*clkLO*) sending clock for TXD module. *clkMain* input is external CLK input. *loRst* is timer reset input.

Fig. 9. Schematic diagram of the TXD TIMER module.

H. TXD module

TXD module (Fig. 10) is responsible for data transmission on *TxD* output line – outside of the end module. It takes 3 inputs: *clkLo* – sending clock, *send* – signal which starts byte transmission and *data* – byte to send. Outputs are: *TxD* which is connected with TXD OUTPUT as external transmission line and *loRST* for counter/timer reset.

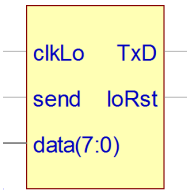

Fig. 10. Schematic diagram of the TXD module.

IV. RESEARCH RESULTS

The implementation and the synthesis was made using two different development tools: Xilinx ISE Design Suite 14.7 – synthesis to the following devices: Kintex-7 (XC7K70T), Spartan 3E (XC3S1600E), Virtex-6 (XC6VLX75T); Quartus II 13.1 – synthesis to the following devices: Arria II GX (EP2AGX45DF29C4), Cyclone IV E (EP4CE 115F23I8L), Cyclone V (5CGXFC7D7F27C8). Source code for both development platforms and for all devices was customized to be universal (all devices used same source code).

Results of the synthesis for Xilinx devices of each module is shown in Table II. Maximum percent of total register (*Reg.*) resource usage was noted for Spartan 3E (XC3S1600E), percent of the maximum occupied slices (*Slices*) and LUTs (*LUTs*) for the Kintex-7 (XC7K70T). Furthermore, minimum percent of resource usage was noted for Virtex-6 (XC6VLX75T) device.

TABLE II. RESOURCE USAGE FOR THE XILINX DEVICES

Implemented modules	Device								
	Kintex-7 (XC7K70T)			***Spartan 3E (XC3S1600E)***			***Virtex-6 (XC6VLX75T)***		
	Reg.	*Slices*	*LUTs*	*Reg.*	*Slices*	*LUTs*	*Reg.*	*Slices*	*LUTs*
RXD TIMER	21	14	41	21	17	29	21	14	47
RXD	31	14	28	32	33	41	31	13	28
RECEIVER BUFFER	266	93	289	266	199	264	265	127	289
PARSER	112	170	364	0	37	46	0	18	28
CONTROLLER	0	1	2	0	2	3	0	1	2
TRANSMITTER BUFFER	9	2	8	9	6	9	9	3	8
TXD TIMER	12	8	29	12	10	16	12	11	29
TXD	8	9	15	8	12	19	8	7	17
Total used	459	311	776	348	316	427	346	194	448
Total available	82000	10250	41000	29504	14752	29504	93120	11640	46560
Usage [%]	0,56	3,03	1,89	1,18	2,14	1,45	0,37	1,67	0,96

Results of the synthesis for Altera devices of each module is shown in Table III. Maximum percent of total register (*Reg.*) resource usage and percent of the maximum occupied LUTs (*LUTs*) was noted for Arria II GX (EP2AGX 45DF29C4). Minimum percent of total register (*Reg.*) resource usage was noted for Cyclone V (5CGXF C7D7F27C8). Minimum percent of LUTs (*LUTs*) for both Cyclone IV and Cyclone V devices.

TABLE III. RESOURCE USAGE FOR THE ALTERA DEVICES

Modules	Device					
	Arria II GX (EP2AGX 45DF29C4)		***Cyclone IV E (EP4CE 115F23I8L)***		***Cyclone V (5CGXF C7D7F27C8)***	
	Reg.	*LUTs*	*Reg.*	*LUTs*	*Reg.*	*ALMs*
RXD TIMER	21	28	21	28	21	18
RXD	36	35	32	47	36	23
RECEIVER BUFFER	262	152	262	275	263	140
PARSER	113	32	113	155	113	73
CONTROLLER	3	2	3	3	3	2
TRANSMITTER BUFFER	9	2	9	9	9	5
TXD TIMER	12	16	12	16	12	11
TXD	8	15	8	17	8	10
Total used	464	282	460	550	465	282
Total available	36100	36100	114480	114480	225920	56480
Usage [%]	1,29	0,78	0,40	0,48	0,21	0,50

All of the presented results, for all of the selected devices for comparison are oscillating around 1-2% of the total available resource usage. Xilinx and Altera devices, compared between each other or even within same manufacturer has got different hardware architecture, so it is impossible to make direct comparison of them. Cheap and small devices like Arria II GX (EP2AGX45DF29C4) and Spartan 3E (XC3S1600E) uses about 1% more of the available resources than other presented platforms.

Average register usage for the Xilinx devices is 0,7%, average for occupied slices is 2,28% and average for used LUTs is about 1,43%. Average register usage for Altera devices is 0,63% and average LUTs usage 0,59%. This comparison allows to make conclusion that in this specified implementation and synthesis, Altera device with Quartus II development tool, gives a little bit more efficient synthesis result that Xilinx ISE.

The most important research results is very low resource usage for all of the devices, after implementing CloudBus protocol. It allows to use approximately 98% of resource e.g. implementing control algorithm.

V. CONCLUSIONS

This paper presented the implementation and synthesis results of the CloudBus protocol for distributed embedded systems on different FPGA platforms. It considers the resource usage depending on the FPGA device and development platform.

Presented research results allows to make conclusion, that implementing CloudBus protocol on the FPGA platform gives negligibly small resource usage. CloudBus protocol implementation almost doesn't limit the implementation of the other control algorithms, on the same field-programmable gate array. This feature is especially important in large and complex embedded systems which needs lot of the resources to preform designed control algorithm.

REFERENCES

[1] M. Adamski, A. Karatkevich and M. Wegrzyn "Design of Embedded Control Systems", Springer, 2005

[2] K. Krzywicki and G. Andrzejewski, "Concurrent process synchronization in distributed systems", Proceedings of the XV International PHD Workshop – OWD 2013, 2013, pp.36-39

[3] J.P. Deschamps, G. Bioul and G. Sutter, "Synthesis of arithmetic circuits: FPGA, ASIC and embedded systems", Wiley, 2006

[4] H. Attiya and J. Welch, "Distributed Computing: Fundamentals, Simulations and Advanced Topics", J. Wiley Interscience, 2004

[5] V. K. Garg, "Elements of Distributed Computing", Wiley&Sons, 2002

[6] L. Shang and N.K. Jha, "Hardware-software co-synthesis of low power real-time distributed embedded systems with dynamically reconfigurable FPGAs", ASP-DAC '02 Proceedings of the 2002 Asia and South Pacific Design Automation Conference, 2002, pp. 345

[7] K. Krzywicki, G. Andrzejewski, "Interfejsy wymiany danych w systemach rozproszonych", in press

Practical use of FPGA Chips for Implementation of Linear Motor Control System

Matěj Bartík
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Email: bartimat@fit.cvut.cz

I. Abstract

This paper describes design and implementation of control system for a Linear Motor EZ Limo EZC6E030M–C and its control unit ESMC–C2. These parts were supplied with control application, that was insufficient for research team of Laboratory of Biomechanics, Department of Mechanics, Biomechanics and Mechatronics, Faculty of Mechanical Engineering, CTU in Prague.

The Linear Motor should be used for measuring mechanical parameters of various samples or should be used as a universal pulsator for simulating flow inside blood–vessels or heart beating.

The new control system has been designed and developed to satisfy all requirements of research team of Laboratory of Biomechanics. These requirements include execution of complex trajectories with high dynamic range of speed and distance, cooperation with other equipment in Laboratory (synchronization), make the system fail-safe and reliable.

This functionality is required for precise measuring of mechanical parameters of biological and synthetic samples. Control system implementation benefits from advantages of FPGA chips for reaching precise timing, deterministic behaviour and high reliability at high speed.

Control system is implemented as a systolic algorithm (Fig.1), where all parts are interconnected by queves. Main module is realized by complex FSM, that can handle 27 different instructions for performing complex movements and setting configuration for system. System can recognize and handle 22 events with 8 priority levels. Every unexpected state result in system stop to prevent destruction of system or measured samples.

The control system has been split into software part (control/manipulation application) and hardware part (new advanced control unit implemented on Xilinx's FPGA). The control system and all its parts were exhaustly tested under full operational conditions. The control system actually does not meet EMC directives. New printed circuit board is currently under development to meet the EMC directives.

The whole control system is part of larger project NT13302 (The Optimalization of Physical Characteristics of Vascular Substitutes for Low Flow).

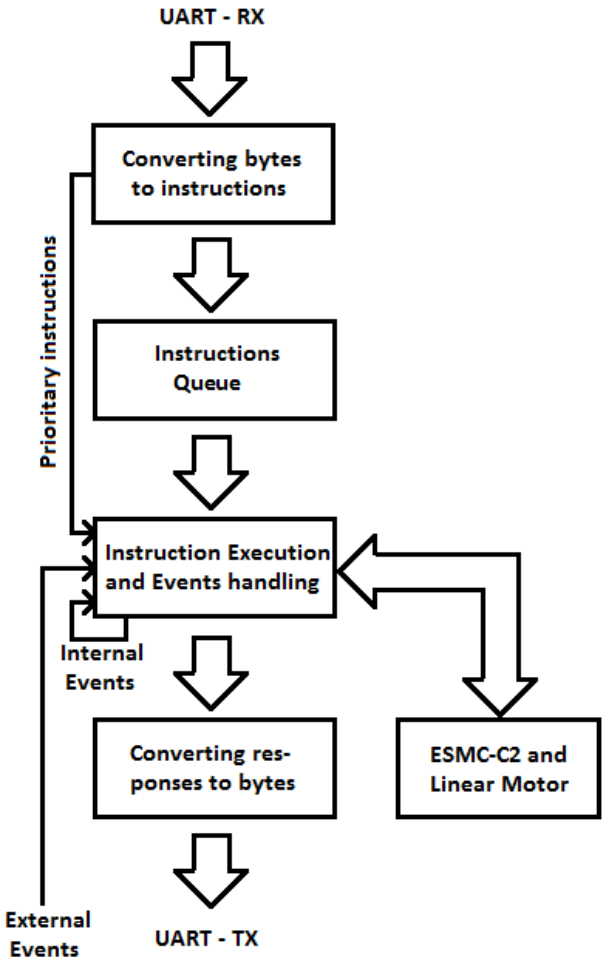

Fig. 1. Control System Architecture — Block Diagram

Towards Trusted Devices in FPGA by Modeling Radiation Induced Errors

Jan Pospíšil, Tomáš Vaňát, Jan Schmidt
Department of Digital Design
Czech Technical University in Prague, Faculty of Information Technology
Prague, Czech Republic
Email: jan.pospisil@fit.cvut.cz, tomas.vanat@fit.cvut.cz, jan.schmidt@fit.cvut.cz

***Abstract*—Field Programmable Gate Arrays (FPGAs) offer great opportunity for wide range of applications. FPGA chips are also used in secure applications, where they can represent critical parts of such systems when implementing some of key functions. On one hand, FPGAs offer universal usage through their programmability, but on the other hand their complex structure is prone to several types of disturbances, one of which can be a radiation, either natural, or induced by an attacker. The response of basic CMOS structures to radiation (e.g. laser or ionizing radiation) is well documented in terms of Single Event Effects (SEE) and Single Event Upsets (SEU), but the impact of these interferences on a particular design implemented in an actual FPGA can be predicted with difficulty. This weakness can be exploited to fault-attack the target design. Although radiation attack is less probable than other types, a security device properly eliminated from service by radiation can impose a significant risk to the whole system.**

There are two main possibilities of testing the impact of radiation on the design dependability parameters. The first one is the Accelerated Life Test (ALT), which is based on increasing the amount of the radiation in the environment, where the device is tested. Using this method (in simple terms), the device receives the same amount of radiation as it would receive during years of operation, in few minutes. If appropriate energy spectrum of particles is used, this method gives very accurate results. The problem is that it is very expensive and not easily accessible. The alternative is to use a simulation model. To create the model, it is necessary to have a detailed description of the physical structure of the tested hardware, which is not always available for commercial FPGAs, thus we can use only approximate models. Methods allowing to classify faults in nearly all models have been published.

In the presentation we will introduce our method of modeling the behavior of FPGA described by a custom architecture with emphasis to resistance to radiation induced soft-errors. Because of detailed descriptions of architectures used in commercial FPGAs are not available, we have to propose an artificial architecture based on all knowledge we have about the commercial ones. This proposed architecture is consecutively processed through modified open-source VTR toolchain and testing designs are placed and routed to this architecture by the same tool. Beside this, ALT fault injections on real FPGAs are conducted to obtain data characterizing a behavior of the same testing designs under radiation. Data obtained by these experiments helps to calibrate the results obtained by simulation of the proposed architecture. The result of this work will provide a calibrated model to this method to perform more realistic SEU testing and prediction of the impact of natural radiation and radiation-based attacks on a design implemented in FPGA. As a result, the obtained models will help us to design radiation and attack resilient FPGA architectures, and/or attack-resistant designs based on existing FPGAs.

Optimal algorithm for phase shift searching for the DPSBF

Viktor Cerny
Czech Technical University,
Faculty of Electrical Engineering,
Czech Republic, Prague
Email: viktor.cerny@fit.cvut.cz

Alex Moucha
Czech Technical University,
Faculty of Information Technology,
Czech Republic, Prague
Email: alex.moucha@fit.cvut.cz

Jan Kubr
Czech Technical University,
Faculty of Electrical Engineering,
Czech Republic, Prague
Email: kubr@fel.cvut.cz

***Abstract*—The distributed phase shift beamforming is the great technique for spatial signal filtering in wireless networks. We already showed that this technique can be exploited for the interference reduction in wireless networks. This paper deals with optimal phase shift search for this method.**

I. Introduction

The distributed phase shift beamforming method needs to be supported by the hardware and software working closely together. This paper optimizes the software part. The one of the main problems is the proper phase shift search. It can be done by brute force however this attitude is almost always time consumptive. If the distributed phase shift beamforming technique is used in the connected network, they can be exploited the existing connections for optimization of phase shift search. This paper propose the modified binary search algorithm which should significantly decrease the complexity of the phase shift searching.

In the world of wireless networks are many of techniques to create some type of beamforming. The basic principle of our type of beamforming is described in the second chapter of this paper. The third chapter propose the new optimal phase shift searching algorithm whose detailed analysis is in the following fourth chapter.

II. Problem definition

A. Network Model

Let us first consider the pure mathematical network model: the network topology can be considered as an undirected graph $G = V \times E$ where the set of vertices V represents the set of hosts and the set of edges E represents the connections between them.

Hosts in V are randomly distributed over a flat surface (thus we take into account only the horizontal plane of the antenna radiation patterns) and all of them are identical. Because in this paper we compare the improvement brought by interference cancellation towards interference in the same network without cancellation (in the same conditions of propagation and with identical transmitters) we can simplify the model by supposing that there are no obstacles involved and no ground reflection therefore the signal propagation is ideal. Every host has only one data transceiver connected to one ideal antenna (gain is 0 dBi in all directions).

The energy source in this model is great enough to be considered as unlimited thus we do not apply models for energy conservation, being oriented towards interference only.

Edges in E represent connections between hosts. Any edge between two hosts can be established only if each is in the transmission range of the other.

B. Distributed Phase Shift Beamforming

In recent years many varied techniques which use different types of beamforming were developed. The other name for such techniques is even more precise: spatial filtering. With beamforming it is now possible to precisely determine the spatial destination of the transmission. Beamforming can be established by using sector turning antennas or by using multiple sector antennas which can be switched on and off on a per packet basis. The same result can be obtained by using antenna arrays [2]. Our work is derived from the antenna array results.

The antenna array is a set of antennas (in many cases omnidirectional) which are connected to a shared transceiver and they transmit the same signal simultaneously. The positions of all antennas are known and, in most cases, the distances between them are comparable to the wavelength which is used for the transmission. The resulting radiation pattern is affected by the multiple signals addition. There are directions where the signal is amplified (constructive interference) and directions where is attenuated (destructive interference). The resulting amplitude change thus depends on the mutual phase difference.

C. Data transmission

The effect of the antenna array can be achieved in a distributed way by independent modules equipped with only one data transceiver and one antenna. In this case there is no central transmitter and the positions of all antennas are unknown. This distributed antenna array needs to be synchronized by using a second transceiver (not used for data transmissions) in order to transmit synchronously with certain signal phase combination. Explanation of the synchronising mechanism is beyond the scope of this paper and it is described in [3] and in the granted patent [4]. The mutual phase shift between transmitters affects the resulting shape of the radiation pattern. In figures 1 and 2 two radiation patterns which were created by two transmitters can be seen. The distance between the transmitters is 30λ and the only difference between them is the mutual phase shift.

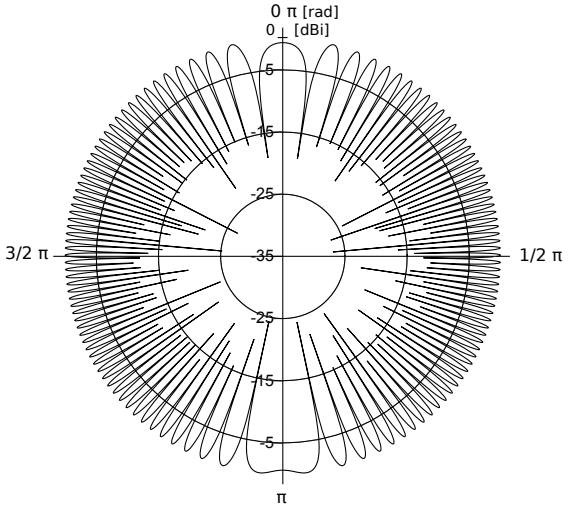

Fig. 1. Radiation pattern of two transmitters with phase shift $\pi/3$ and distance 30 wavelengths

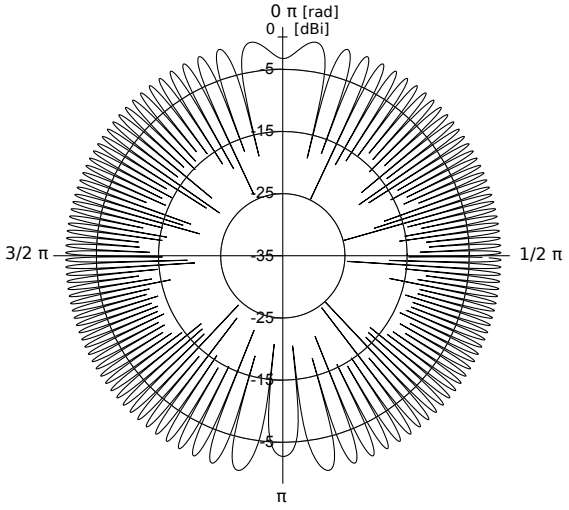

Fig. 2. Radiation pattern of two transmitters with phase shift $\pi + \pi/3$ and distance 30 wavelengths

In this work we will use only two transmitters to create the transmission using DPSBF. The proper timing of both synchronized transmitters is crucial for the successful signal delivery to the destination. The hardware synchronizing mechanism for DPSBF is described in [3], [4] and its explanation is beyond the scope of this paper. In this text we simply assume that synchronization is fully reliable. The transmission using DPSBF takes the same time as the classic one and it can be received be the classic receiver without any special hardware. There is only one condition - the data for the transmission has to be present on both transmitters.

D. Interference Cancellation

As it can be seen in the figures 1 and 2 the resulting radiation patterns using DPSBF do not cover the same area as the single transmitter (whose coverage area is modelled by a circle due to the ideal antenna). This spatial filtering allows the transmission to reach the destination host and, in the same time, can cover a smaller area than the single transmitter.

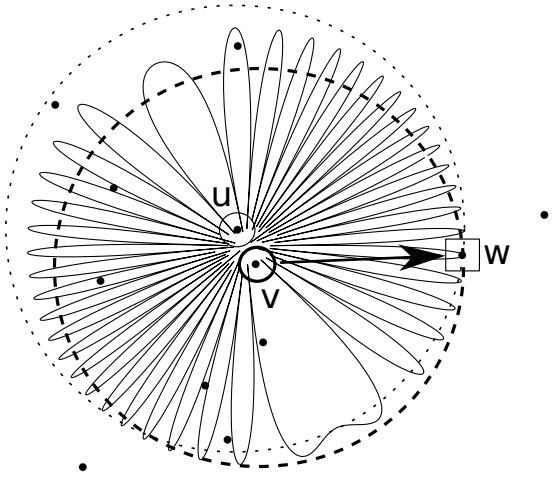

Fig. 3. Example of transmission without DPSBF and with DPSBF

Let us consider the situation in the figure 3 which contains one zone of a wireless network having several hosts. Two of them are denoted by circles (the transmitters u and v) and they are together transmitting by using DPSBF to the destination host (receiver w) which is denoted by the a square. If the DBSBF would not have been used, v would have been used for the classic direct transmission. The continuous line marks the range of both transmitters transmitting together by using DPSBF and the dashed circles mark the range of the individual transmitters without cooperation (u and v). In all cases the transmitting power is set to the minimal value which is sufficient for reaching the destination receiver.

From the figure is obvious that the transmission using DPSBF covers completely different surface and in this case is more convenient than the classic transmission because less other hosts are affected by the transmission. More information about this topic is in [1].

Figure 3 shows us that in certain areas in the network topology it is more convenient to use transmission with DPSBF instead of the classic one. The stronger dashed circle denotes the transmission range of v. We can see that six hosts are covered by this transmission. If we use both transmitters together in DPSBF, the resulting radiation pattern covers only four of other hosts. It means that two more hosts are able to receive other transmission in the same time (spatial filtering).

In figure 3 we can see that the transmission with DPSBF covers the uppermost hosts which were not covered by the classic transmission. In this case the overall result is better (lower interference) however there could be cases in which using DPSBF leads to worse results (higher interference). The solution to this problem is to use classic transmission or DPSBF depending on the results achieved by each method: as TC replaces long edges in the graph with shorter ones only when the shorter ones are better (the metric depends from TC algorithm to algorithm) we can replace edges by DPSBF transmission only when the interference of DPSBF is lower than the original edge caused interference.

By this technique is possible to improve the quality of some edges from the interference view. Not all edges can be improved because the DPSBF is not applicable in all situation.

Unfortunately we do not know in advance which edge can be selected for the DPSBF. Therefore it is necessary to check every edge whether it could be improved.

E. Phase Shift Searching

In order to be the DPSBF applicable, it needs to be properly configured however the finding of the proper phase shift is a time consumptive process. Unfortunately we need to find the best phase shift as possible. It is due to minimizing of the resulting radiation pattern surface. With this pattern is possible to reach the destination with minimal power and in the same time to cover minimal area by the signal. Let us consider that there can be active only one searching process in whole network. (It is possible to execute more phase searching in the same time in one network however one execution of the searching can affect another one if they are in the transmitting range. The solution of this problem is a topic for another paper.) According to this assumption is necessary to execute every searching individually for every edge in the network. Furthermore the transmitting hosts can have multiple neighbours and every of them can be used for DPSBF and every needs individual phase shift searching. Every host x is the neighbour of host u when:

$$\{\forall x \in V \setminus \{u\} : (ux) \in E\} \quad (1)$$

where (ux) means the edge between u and v. If exists the edge between hosts in E, they are neighbours. The searching can be done by brute force. At first is defined the minimal shift step (MS). After it can start searching. For every possible phase are sent data to destination (for example to the host w in the figure 3) and the number of this transmissions is $360/MS$. The host w collects the received signal quality during whole searching and at the end of searching is asked by the host v to send back the best result. After it the host v knows best phase shift between itself and the neighbour host which was used for the searching. The number of all transmissions over the whole network during network initiation can be roughly computed:

$$trNumber = 2 * |E| * (avgNeigh - 1) * (360/MS + 2) \quad (2)$$

where $trNumber$ means overall number of all transmissions to be executed, $avgNeigh$ is the average count of neighbours for all hosts in network. It can be computed: $avgNeigh = 2 * |E|/|V|$. Explanation of all three members of this equation:

1) **2 * |E|** - the doubled number of all edges in the network. The searching needs to be done for all edges in the network however the results are different for both sides of the edge.
2) **(avgNeigh - 1)** - every couple of hosts gives different results. The phase searching needs to be executed for all neighbours of the host which initiates the transmission with DPSBF.
3) **(360/MS + 2)** - number of all possible phase shifts. The two additional transmissions are request for the best result to the destination and answer for this request.

To minimize the count of all transmission during the network initiation there is very few possibilities. Lets discuss about all three parts:

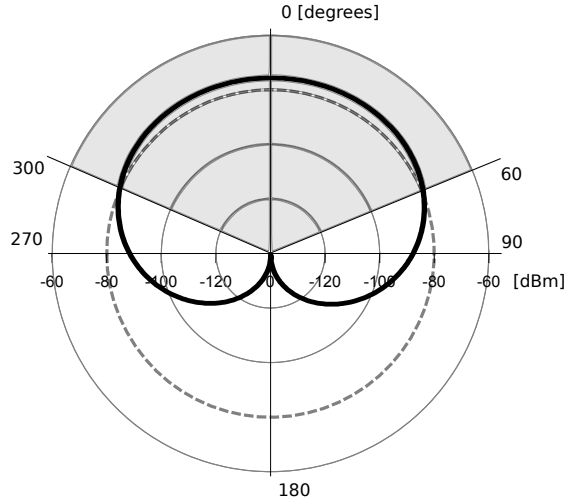

Fig. 4. The dependency of the received power [dBm] on the phase shift [degrees] with using two host for transmission with DPSBF

1) The number of all edges can be decreased for example by some topology control algorithm. This solution unfortunately changes the network topology and it can not be always possible.
2) This part could be improved by a smart neighbour selection. The neighbours which are closer to the original host are in most cases more convenient for the transmission with DPSBF. The smart selection of neighbours has no great effect to the overall complexity especially in the denser networks. Furthermore if we remove some neighbours from the searching, the resulting network performance could be negatively affected.
3) The value MS is crucial for the resulting number of transmissions for this part of equation. The increasing of MS decreases the transmission number however in the same time it is significantly decreased the chance for finding the good phase shifts. In fact we want to have the MS smallest as possible.

From the discussion is obvious that most interesting is the third member of the equation. The MS value affects rapidly the quality of result and overall number of transmissions. The MS value is inversely proportional to the overall number of transmissions.

III. Proposed Solution

In the previous chapter we showed that searching for convenient phase shift for all edges is a complex problem. In the figure 4 can be seen the dependency of the received power on the phase shift used for transmission with DPSBF. The chart is made from the view of the receiving host (for example the w in the figure II-D). The dashed circle denotes the sensitivity of the receiver and the signal power above this line can be received. This sector is grayed. In the situation in the figure 4 is the maximal power for the phase shift 0 degrees. In order to minimize the number of trials by the searching process, we can add immediate feedback from the host w to host u. The host u is able to change the phase shift according the information from the host w. It knows immediately if the

last trial transmission improved or decreased the signal power on the host w. With this knowledge it can tune the phase shift towards the maximum received power. In the next few chapters is described the algorithm based on the previous facts. This algorithm is composed of two stages.

A. Stage 1

At the beginning the both transmitting hosts set the power to the maximum. In the Stage 1 the algorithm needs to find such phase shift for which is signal received on the destination host. It does not matter if the signal is strong or weak. Most important is that the signal is received. Look at the figure 4. We only need to find phase shift from the area where is the signal above threshold, so it could be any value from 300 degrees to 60 degrees.

Lets define the searching step (SS). This step begins at maximum possible value - 360 degrees and it is decreased to the half at every iteration of the algorithm. The Stage 1 logic can be seen in algorithm 1. There is defined the set $used$ which should contain all phase shift values which were tried by the algorithm. The main loop continues as long as the signal is not received on the destination host. The Stage 1 finishes when the signal is received on the destination host or if the SS is less than MS. If the signal was not received until the $SS > MS$, the desired phase shift was not found. This situation can possibly occur when the MS is too great and the desired area is too small.

The Stage 1 has two loops. The outer loop ensures termination of whole algorithm. The inner loop sends sets of messages to the destination in such way that phase shifts uniformly cover whole range of 360 degrees and their density decreases at every iteration. The function *send()* has two parameters. First is the phase shift and second is the identification of the destination host. This function distributes the data to the second transmitter and executes the transmission using DPSBF. Into data, which are sent to the destination, is encapsulated information about current phase shift so the destination host can identify the best incoming signal and it can send this information back to the source of transmission. Every message sent by DPSBF contains the information about current phase shift. The destination host keeps this information until is asked to send it as the answer to the source host.

B. Stage 2

In the Stage 1 we searched the phase shift for which is signal received on the destination host. The Stage 2 is looking for the best phase shift which gives a greatest signal power on the destination. Look at the figure 4. The Stage 1 found the phase shift which is somewhere in the grayed area. In this stage we want move the phase towards 0 degrees because there is in our example maximum of power.

The Stage 2 keeps the SS value from the previous stage and sets the current phase to the best value which was found during the first stage. The SS is decreased by half at every iteration and this step is used for exploring the smaller and smaller neighbourhood of the current phase shift. In every iteration are sent two trial transmissions and after that is destination host asked to send the result. In the answer is information whether were the new phase shift better or worse. See the algorithm 2.

Algorithm 1 Stage 1

```
1:  SS ← 360                          ▷ Current searching step size
2:  used ← empty set                  ▷ Set of all checked values for SS
3:  while SS > MS do
4:      p ← 0
5:      for p < (360/SS) do
6:          if s ∉ used then
7:              send(s, destination)
8:              used ← s
9:          end if
10:         p ← p + 1
11:     end for
12:     SS ← SS/2
13:     ask the destination host for the results
14:     if destination received signal then
15:         phase ← best received phase
16:         break
17:     end if
18: end while
```

Algorithm 2 Stage 2

```
1:  phase ← best confirmed phase from the Stage 1
2:  SS remains unchanged from the Stage 1
3:  while SS > MS do
4:      send(phase + SS, destination)
5:      send(phase − SS, destination)
6:      ask the destination for the result
7:      according the confirmation do:
8:      phase ← [phase + SS|phase − SS|SS]
9:      SS ← SS/2
10: end while
```

C. Optimal Algorithm for the Searching of Phase Shift for the DPSBF

The final algorithm is very simple and it is composed of both stages. See the algorithm 3.

Algorithm 3 Composition of Both Stages

```
1:  phase ← 0
2:  execute the Stage 1
3:  if the Stage 1 was successful then
4:      execute the Stage 2
5:      return phase
6:  else
7:          ▷ The Stage 1 did not find the usable phase shift
8:      return ERROR
9:  end if
```

IV. Algorithm analysis

The main purpose of the proposed algorithm is decreasing of communication and time complexity. In this chapter we will analyse both stages of the algorithm. In the following expressions the n equals to $360/MS$ and it is the number of transmissions which are sent by the DPSBF. Before we analyse our algorithm, we should know the complexity of the brute force solution. If we use it, the maximal possible number of transmissions is executed so the complexity is:

$$O(n) \tag{3}$$

A. Stage 1 Complexity

At first lets begin with best case. The main purpose of the Stage 1 is to find some phase shift which allows the signal delivery to the destination. It is possible that the first trial transmission is successful. In the first iteration is sent only one message to the destination. After every iteration is sent request for the results and the answer. It gives together three transmissions. In the best case is the complexity:

$$\Omega(1) \tag{4}$$

The worst case is the situation when the success comes in the last iteration (or if there is no success). In the first iteration is sent one message (the same like in the best case). In the second iteration is sent one message too. Every next iteration is sent the doubled number of messages of the previous iteration. After every iteration are sent two messages - request and answer. We can express the overall number of all transmission as:

$$3 + \sum_{i=0}^{\log_2(n)} (2^i + 2) \tag{5}$$

The complexity in the worst case is:

$$O(n + log_2(n)) \tag{6}$$

In the worst case needs to be executed n transmissions using DPSBF and $2log_2(n)$ classic transmissions for request and answer messages after every iteration. After comparison with complexity of the brute force solution (equation 3), we can see that the complexity of the Stage 1 is worse. In the following sections we will show that worst case never occures.

B. Stage 2 Complexity

The analysis of the Stage 2 is more easier. In the best case are executed four transmissions (only one iteration is executed). The number of iteration is in the worst case $\log_2(n)$ and every iteration are executed exactly four transmissions. The complexity equals to:

$$\Theta(log_2(n)) \tag{7}$$

C. Algorithm Complexity

In the figure 5 we can see the first four iterations of the Stage 1 part of the algorithm. The finite number of algorithm steps depends on the size of the mutual phase shift between both compounding signals.

Theorem 1. *Stage 1 of the proposed algorithm finishes the most in two steps in all possible cases.*

Proof: As we can see in the figure 5 that the first part of the algorithm finishes in two steps only if the range of the phase shift between signal is equal to or bigger than 180 degrees. Let us set one of the signal as the reference. If we add any other signal to the reference and the mutual phase shift between signals will be between 0 and 90 degrees, the resulting signal will be always greater than the original reference signal (see the figure 6b, 6c). So now we can see that the we always have 180 degrees range for the successful signal transmission to the destination therefore the we can do that only in two trials. ■

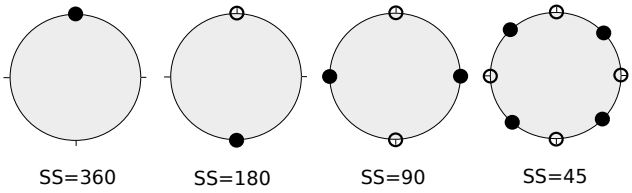

Fig. 5. Execution of Stage 1

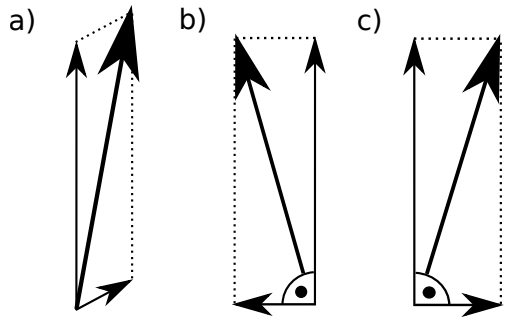

Fig. 6. The signal addition

Now we can put together complexity of both stages. According the previous complexity equations and the theorem, we know that Stage 1 complexity is $O(1)$ so the overall complexity equals to the complexity of the Stage 2:

$$\Theta(log_2(n)) \tag{8}$$

V. Conclusion

We showed that the complexity searching of the phase shift for the method DPSBF can be decreased from the linear to the logarithmic complexity. Using modified binary search algorithm can really speed up the initial process of the network using interference cancelling by usage of DPSBF. In case of periodic network reconfiguration could the proposed method conserve the energy of hosts.

References

[1] Viktor Cerny, Alex Moucha and Jan Kubr, *Interference Cancellation by the Usage of Distributed Phase Shift Beamforming*, 2014.

[2] Constantine A. Balanis, *Antenna Theory: Analysis and design*, John Wiley & Sons, Inc., ISBN-10: 047166782X, ISBN-13: 978-0471667827, 2005.

[3] Alex Moucha and Viktor Cerny, *Anisotropic Antenna Collaborative Beamforming in AdHoc Networks - Beyond Horizon Communication*, Proceedings of the International Wireless Communications and Mobile Computing IWCMC conference, ACM, ISBN 978-1-4503-0062-9, 2010.

[4] Alex Moucha, Jan Kubr and Viktor Cerny, *Distributed System for Beamforming*, Patent - Urad prumysloveho vlastnictvi, 2011-785, CZ303761.

[5] Jan Kubr, Viktor Cerny and Alex Moucha, *Advanced Methods for Phase Search in Beamformed Ad-Hoc Wireless Networks*, Proceedings of the International Conference on Telecommunication Systems Management ICTSM, IEEE, ISBN 978-0-9820958-8-1.

[6] K. R. Gabriel and R. R. Sokal, *A new statistical approach to geographic variation analysis*, Systematic Zoology (Society of Systematic Biologists) 18 (3): 259270, 1969.

[7] G. T. Toussaint, *The relative neighborhood graph of a finite planar set*, Pattern Recognition 12 (4): 261268, 1980.

Fault Recovery Method of Modular Systems based on Reconfigurations

Jaroslav Borecký, Pavel Vít and Hana Kubátová
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Email: {borecjar; pavel.vit; kubatova}@fit.cvut.cz

***Abstract*—This paper presents the method of dependability parameters improvement for systems based on unreliable components such as Field Programmable Gate Arrays (FPGAs). It combines Concurrent Error Detection (CED) techniques [4], FPGA dynamic reconfigurations and our previously designed Modified Duplex System (MDS) architecture. The methodology is developed with respect to the minimal area overhead. It is aimed for practical applications of modular systems. Therefore it is applied and tested on the safety railway station system. This Fault-Tolerant (FT) design is tested to fulfill strict Czech standards [7]. The proposed method is based on static and partial dynamic reconfiguration [5] of totally self-checking blocks which allows a full recovery from a Single Even Upset (SEU).**

I. Introduction and Motivation

Systems realized by programmable hardware like FPGAs are widely used in all of applications due to their capability to implement complex circuitry within a very short development time, together with the potential for an easy change of a design by reconfiguration.

Thanks to the Partial Dynamic Reconfiguration (PDR) the FPGAs will be more applied because a part of the circuit can be changed without disturbing of a rest of the functional FPGA. But PDR can be applied in a different way and it can help us to increase dependability parameters.

Most of modern FPGAs are based on SRAM memories. These logic arrays can not be used in mission critical applications without any additional protection due to their high sensitivity to the radiation effects. For example a Single Event Upset (SEU) changes one bit of configuration memory, that causes a radical change of an implemented circuit. Applications used in space missions or public transport need to satisfy strict safety standards to avoid tragic consequences. We propose on-line testing method, because these critical applications must not be interrupted by any tests.

The method described below can be used in highly reliable modular systems which are based on these unreliable components.

II. Theoretical Background

Whole device is composed of different modular system blocks. Each part of a block has to be secured, because a SEU can occur. A change of one bit leads to a modification of the circuit function, often drastically. That causes unpredictable behavior in practical applications, for example the control device can change signals to green in all traffic lights of a crossroad.

Therefore we must guarantee continuous function without interruption, and it is possible only by on-line tests. Our method is based on methods which follows.

A. Totally Self-checking Circuit

Every Totally Self-checking Circuit (TSC) is composed of three small parts, where each block corresponds the TSC property. The universal structure of the compound design satisfying the TSC property is shown in Fig. 1.

You can see six places where an error can occur in the TSC block diagram shown below. The idea is, that if an error is in the check bits generator, it will be observable on the check bits wire (the wire number 1). When an error is in the original combinational circuit, it will be observable on the primary output (the wire number 5). This implies that the checker in block N will detect an error on the wire number 1, 2, 4 or 5. Or if an error occurs on the wire number 3 or 6, it will be detected in the next block (N+1) by its checker. The method used to satisfy the TSC property for the compound design is described in detail [2].

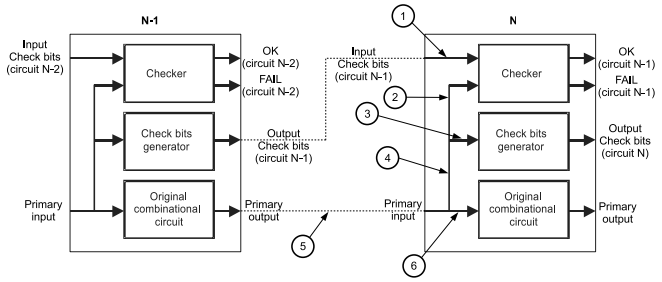

Fig. 1. The structure of compound system corresponding the TSC property

Not every TSC block (in the compound design) satisfies the fault coverage to 100%. The TSC structure, which uses only one copy of the TSC circuit is not sufficient to increase dependability parameters. Thus, we assume to use our Modified Duplex System (MDS) architecture [1], [2], which has a parity generator in all TSC.

B. Modified Duplex System

Modified Duplex System (MDS) architecture uses two instances (instead of mostly used TMR architecture like e.g.

[8]) of design that may be not fault tolerant. The purpose of MDS architecture is to achieve the whole circuit including all checkers and comparators to be fault tolerant. The MDS block diagram is shown in Fig. 2.

If an error (caused by SEU) is not detected inside the system by some TSC block, it is detected by comparators. The error detected by comparators triggers initiate the reconfiguration of both blocks (outputs from blocks are different, but the source of the error cannot be determined). But this full reconfiguration is a time demanding process and can cause synchronization problems and therefore leads to decrease of the whole system availability. Due to it the Partial Dynamic Reconfiguration (PDR) is used in our improved architecture.

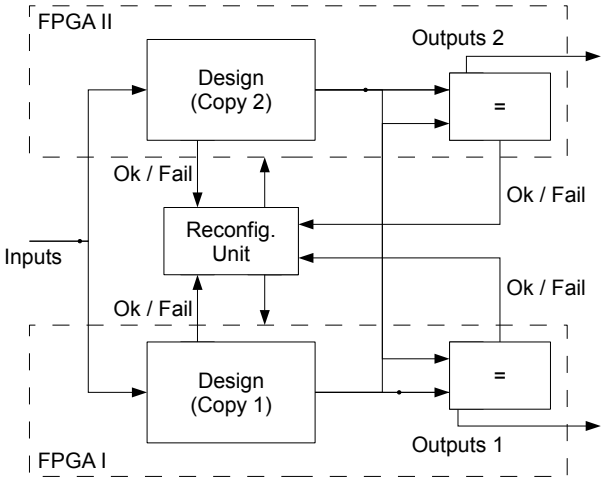

Fig. 2. The block scheme of Modified Duplex System

III. Upgraded Modified Duplex System

Mission critical systems have to run with high availability. It is necessary to develop a methodology how to repair soft errors immediately during their normal operational process. We propose to use an architecture composed of blocks which is derived from practical applications. These blocks will be utilized by partial reconfiguration to repair their transient faults. One big block or few small blocks will be placed in one Reconfiguration Module (RM).

Our method is capable to secure any modular circuit. It was evolved during the evolution of the railway station safety system in our department [3]. This system is modular and based on five types of blocks. This method reduces recovery time, because it uses partial reconfiguration often and whole FPGA reconfiguration only in critical situations. Availability of the whole system increases thanks to a short time of partial reconfiguration. System designed in this way uses less area overhead compared to other methods like TMR or NMR.

A. Basic Scheme

In Fig. 3, you can see our proposed system. It uses two boards with one FPGA, where the same design is loaded.

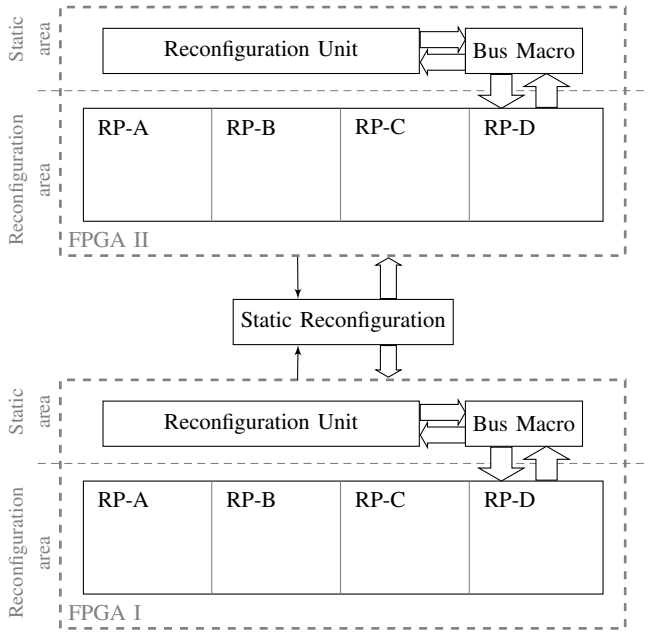

Fig. 3. Upgraded MDS Architecture

It simplifies the systems design and reduces time of developement. There are two main parts (Reconfiguration area and Static area) in each FPGA.

You can see in Fig. 2 that each FPGA in MDS is composed of a design and a comparator. These parts are divided into blocks and placed into Reconfiguration Area in UMDS (bottom part of each FPGA shown in Fig. 3). UMDS uses simpler static reconfiguration unit than the MDS, which is placed between FPGA boards.

The top part of the design is innovated and it improves reliability by performing partial reconfiguration of faulty part when it is needed.

1) Reconfiguration area: is a part of FPGAs which we divided into several Reconfiguration Partitions (RP). The number of RP depends on used application and their size depends on the specific architecture of an FPGA. In one RP, there is also a comparator derived from MDS. One set of RMs is prepared for both FPGAs, where each RM belongs to pertinent RP.

2) Static area: is composed of two parts. The Reconfiguration Unit is constructed by FSM, which controls the status of each TSC block in the reconfiguration area. The Bus Macro is a bridge between reconfiguration and static areas and is here present for compatibility with older FPGAs.

3) Static reconfiguration: is the control logic which performs reconfiguration of the whole FPGA (one or both in the same time). The reconfiguration is initiated by checkers from Reconfiguration Units and Comparators.

B. Fault Recovery Flow

An error can occur in every part of an UMDS and change the functionality some block. This method achieves 100% of fault cover as described below.

When an error is in the static area, the Static Reconfiguration unit performs reconfiguration of the whole FPGA, where the error was detected. When an error is in Reconfiguration area, it could be in the secured design or in the comparator. Errors in secured design are detected by checkers. An error in the comparator is detected by Static reconfiguration unit or checkers. Static reconfiguration unit reconfigures both FPGAs.

When an error is detected by some checker, then Reconfiguration unit reconfigures only this RM. For example RP-A part detects the error and RM-A is loaded into RP-A, where the broken block is placed. Other blocks in different parts (RP-B, RP-C, etc.) are able to work at this time.

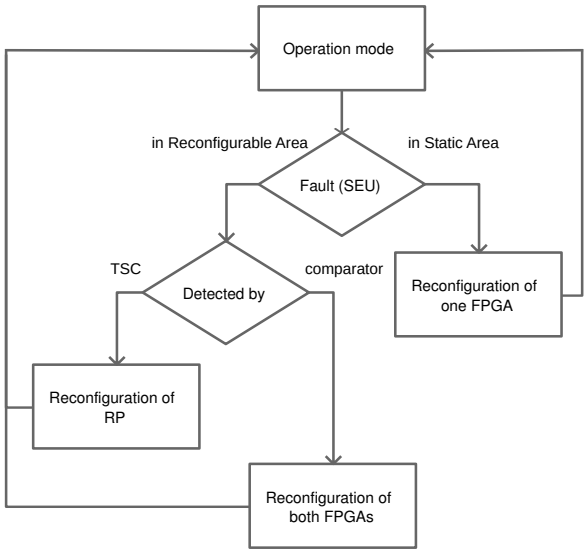

Fig. 4. The block diagram of the Fault Recovery Flow

IV. Conclusions

Our new method of fault recovery of safety systems was presented in this paper. The method is based on two independent FPGA boards with the same design. The FPGA is divided into two main parts. Whole system is placed in the reconfiguration area and static area checks failure signals and immediately repairs soft errors in RPs. Our method is aimed for modular systems which are composed from blocks. Every block is designed as TSC, also the static area satisfies TSC property.

Whole system is derived from MDS and is innovated. The main improvement is in usage of the partial reconfiguration and a block structure of the design. This allows faster detection and correction of faults. Reconfiguration of only one RP is faster than load a whole FPGA. It leads to increase availability and security within minimal area overhead. Most of dependable systems are based on TMR which uses more than three times more area of an FPGA than the original circuit.

Our method was simulated by using Markov model and failure distribution function was calculated. Comparison with original MSD is in Fig. 5.

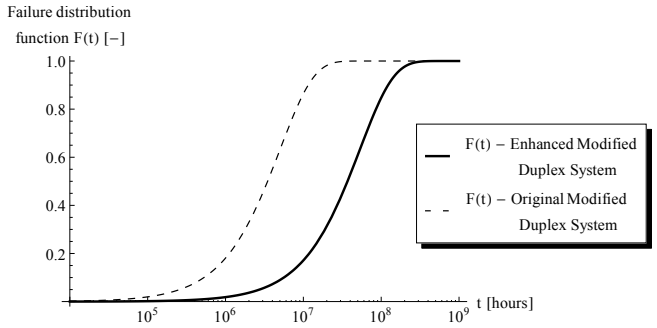

Fig. 5. The block diagram of the Fault Recovery Flow

A. Future work

An implementation of proposed method on the railway station safety device is in progress. We use two XUPV505-LX110T boards with Virtex 5. First tests will check the functionality of whole system and correct function of each block. Finally the verification of our technique will be performed by implementing errors directly into bitstream. One random bit will be changed in the bitstream of one FPGA by a testing device (simulation of SEU). Behavior of the whole system will be monitored and checked.

Another tests and simulations of faults are subscribed in [6]. An insertion of the error will be performed by changing of configuration bits in the FPGA by a neutron beam. A probability of fault will be recalculated in natural environment and compare it with [7].

Acknowledgment

This research has been partially supported by the project SGS14/039/OHK3/1T/18.

References

[1] Kubalik, P., Dobias, R., Kubatova, H.: *"Dependable Design for FPGA based on Duplex System and Reconfiguration"*, In Proc. of 9th Euromicro Conference on Digital System Design. Los Alamitos: IEEE Computer Society, 2006, pp. 139–145

[2] Kubalik, P., Kubatova, H.: *"Dependable design technique for system-on-chip"*, Journal of Systems Architecture, no. 54, 2008, pp. 452–464. ISSN 1383-7621

[3] Borecky, J., Kubalik, P., Kubatova, H.: *"Reliable Railway Station System based on Regular Structure implemented in FPGA"*, Proc. of 12th EUROMICRO Conference on Digital System Design, Los Alamitos, IEEE Computer Society, 2009, pp. 348–354.

[4] D. K. Pradhan. *"Fault-Tolerant Computer System Design"*, Prentice-Hall, Inc., 1996

[5] XILINX *"Xapp864: Seu strategies for virtex-5 devices."*

[6] Vanat, T. and Kubatova, H. *"Experiments with Physical Error Injection into FPGA Circuits"*, Work in Progress, DSD 2011, Oulu, 30.8. - 2.9.2011

[7] SN EN 50126, Czech Technical Norm *"http://nahledy.normy.biz/nahled.php?i=59709"*, 2011

[8] M. Lanuzza, P. Zicari, F. Frustaci, S. Perri, and P. Corsonello. 2010. *Exploiting Self-Reconfiguration Capability to Improve SRAM-based FPGA Robustness in Space and Avionics Applications* ACM Trans. Reconfigurable Technol. Syst. 4, 1, Article 8 (December 2010), 22 pages. DOI=10.1145/1857927.1857935 http://doi.acm.org/10.1145/1857927.1857935

Properties of Boolean functions in Cognitive Complexity Measure

Gabi Shafat
Afeka Tel Aviv Academic College of Engineering
AFEKA
218 Bney Efraim rd. Tel Aviv 69107, Israel
gabis@afeka.ac.il

Ilya Levin
School of Education, Tel Aviv University
TAU
Ramat Aviv, Tel Aviv, 69978, Israel
ilia1@post.tau.ac.il

***Abstract*— The study focused on three different measures of cognitive complexity: Minimal Description (MD), Structural Complexity (SC) and Mental Model (MM). With respect to these complexity measures, the relationship between symmetry (S), linearity (L) and monotony (M) of Boolean concepts and the different complexity measures presented. Effect of properties of Boolean functions on three different measures of cognitive complexity is studied on solving problems of Boolean recognition and Boolean reconstruction.**

Keywords—Boolean Concepts, Recognition, Reconstruction, Faults, Digital systems

An important issue of the theory of concept learning is the ability to predict the difficulty in learning different types of concepts. Difficulties in learning Boolean concepts have been studied extensively by Shepard, Hovland, and Jenkins (SHJ) [1]. This study focused on Boolean concepts with three binary variables, where the concept receives value "1" for 4 out of 8 possible combinations and value "0" for the remaining 4 combinations. Some of the 70 possible Boolean concepts are congruent (NPN-equivalent). They can be divided into six subcategories. The six SHJ subcategories can be represented graphically as follows (Fig 1).

Results of the SHJ study are significant since SHJ formulated two informal hypotheses. The first hypothesis states that the number of literals in the minimal expression corresponds to the level of concept's cognitive complexity.

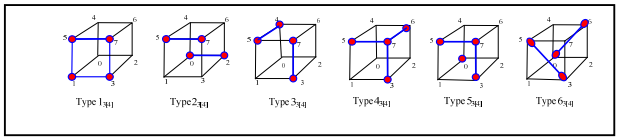

Fig. 1. SHJ category types.

The second hypothesis states that ranking the cognitive complexity among the concepts in each type depends on the number of binary variables in the concept. Feldman [2], based on the conclusions from the SHJ study, defined a quantitative relationship between the level of the cognitive complexity of Boolean concepts. According to [2], the complexity measure of a Boolean concept is the number of literals in the minimal SOP expression that represents the concept. Since there are several minimization techniques, Vigo [3] proposed using the specific Quine-McCluskey (QM) technique to obtain the minimal description (MD). The definition of the Boolean concept's complexity as a minimal number of literals in a minimal expression has two following drawbacks.

1. Since the complexity is defined as the number of literals in the minimal expression and the expressions can be minimized using different techniques, a single complexity cannot be obtained.

2. Studies show that the Boolean concept "xor" is learned and acquired as a concept by humans to not harder than the Boolean concept "or".

Aware of the above issues, Vigo [4], developed an alternative approach for calculating the complexity of a Boolean concept by defining a so-called structural complexity (SC). The approach is based on a Boolean derivative. Vigo's account of the invariance of concepts, as he acknowledges, does not specify how individuals learn concepts. He assumes that cognitive processes could detect invariances by comparing a set of instances to the set yielded by the partial derivative of each variable. Calculations at the foundation of the approach are complex. Mental processes are not taken into account at the foundation of the calculations. SC approach do not comprise a mental representation of concepts or processes.

As an alternative to the complexity theories presented above that predict the difficulty in learning Boolean concepts, a Mental Model (MM) complexity theory [5] is proposed. The MM theory presumes that the mind is not logical and also not a probability system but rather, in essence, it conducts simulations. The theory applies to inclusion thinking and it presumes that when people think, they are attempting to imagine the possibilities of the presumptions that they must address and they draw conclusions. Each of the combinations from all the possibilities that receive a "1" in the result is a MM. When people learn the concepts they can minimize the number of mental models by cancelling irrelevant variables relative to other variables with a known logical value. The number of models of the concept that obtained after minimizing the irrelevant variables predict the difficulty of learning the concept and define the complexity measure of the concept's degree of difficulty.

The recognition problem is can be modeled by using a visual representation of various objects of a common pattern. Solving the recognition problem may thus be considered as recognizing a visually represented Boolean concept, with further formulation of the concept.

The process of finding and reconstructing operating mechanisms in a given functional system of a digital electronic unit is defined as reconstructing (RE) [6]. RE problem means reconstructing a Boolean function implemented within a given "black box".

The experiment was conducted in two stages for 13 concepts, where each concept was described by means of a Boolean expression in Table 1. On the first stage, RE problems was examined using a black box that could be used to control the lighting of a bulb using independent switches. During the second stage, recognition problems (Fig 2) were examined using a questionnaire with 13 patterns, where each pattern represents one of the 13 concepts examined, respectively.

Our paper deals with the question: What is the relationship between property of a Boolean concept and the cognitive complexity of the Boolean concepts? This question refers to the research hypothesis that properties of Boolean functions affect the complexity beyond the complexity measures that were presented.

With regards to property of a Boolean concept, all the complexity measures that we relied on failed to predict the difficulty in solving reconstruction problems. Among the symmetric functions that were tested, the "xor" operator was more complex to solve in the two types of problems compared to other symmetric concepts that were examined. Monotonic and symmetrical concepts are the easiest solution. The structural complexity (SC) measures better predictor compared to the minimal description (MD) and Mental Model (MM), except concepts with properties of symmetry, linearity and monotonicity.

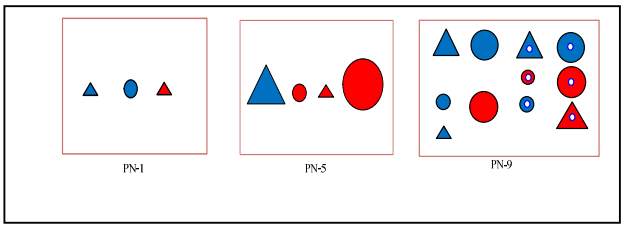

Fig. 2. patterns for CN 1, 5 and 12

Boolean Concept (CN)	MD	SC	M M	Property of Concept
$\bar{b}(a+c)$	3	1.54	2	-
$\bar{a}c+b\bar{c}$	4	2.14	3	-
$\bar{a}\,\bar{b}+ab=\overline{a\oplus b}$	4 (3)	2.14	2	S+L
$a(b+c)+bc$	5	2.14	3	S+M
$(\bar{a}+\bar{b})\bar{c}+abc=\overline{c\oplus(ab)}$	6 (3)	2.34	3	-
$a+bc+\bar{b}\,\bar{c}=\overline{a+b\oplus c}$	5 (3)	2.79	3	-
$a\,b\,\bar{c}+\bar{a}\,b\,c+a\,\bar{b}\,c=b(a\oplus c)+a\,\bar{b}\,c$	9 (6)	3	3	S
$\bar{a}(b+c)+bc$	5	2.14	3	-
$\bar{a}\,b\,d+b\,\bar{c}\,\bar{d}+a\,\bar{b}\,\bar{c}\,d$	10	2.95	3	S+L
$a(\bar{b}\,c+\bar{c}\,b)+\bar{a}(\bar{b}\,\bar{c}+b\,c)=\overline{a\oplus b\oplus c}$	10 (3)	4.00	4	S+L
$a(b\,\bar{c}+\bar{c}\,\bar{b})+\bar{a}(\bar{b}\,\bar{c}+b\,c)=a\oplus b\oplus c$	10 (3)	4.00	4	S+L
$a(b+c+d)+b(d+c)+cd$	9	4.48	6	S+M
$\bar{a}(\bar{b}+\bar{c}+\bar{d})+\bar{b}(\bar{d}+\bar{c})+\bar{c}\,\bar{d}$	9	4.48	6	S+M

Table 1. The 13 concepts were tested during the experiment and their descriptions according to MD minimal descriptions using "xor", SC, MM and Property of Concept (S), (L) and (M).

REFERENCES

[1] Shepard, R. N., Hovland, C. I., & Jenkins, H. M. (1961). Learning and memorization of classifications. Psychological Monographs: General and Applied 75(13), 1-42.

[2] Feldman, J., (2000). Minimization of Boolean complexity in human concept learning. Nature, 407, 630–633.

[3] Vigo, R., (2006). A note on the complexity of Boolean concepts. Journal of Mathematical Psychology, 50(5), 501_510.

[4] Vigo, R., (2009). Categorical invariance and structural complexity in human concept learning. Journal of Mathematical Psychology, 53, 203–221.

[5] Johnson-Laird, P. N., (2006). How we reason. Oxford University Press.

[6] Chikofsky, E.J., & J.H. Cross II, (1990). Reverse Engineering and Design Recovery, A Taxonomy in IEEE Software. IEEE Computer Society, January, 13.17.