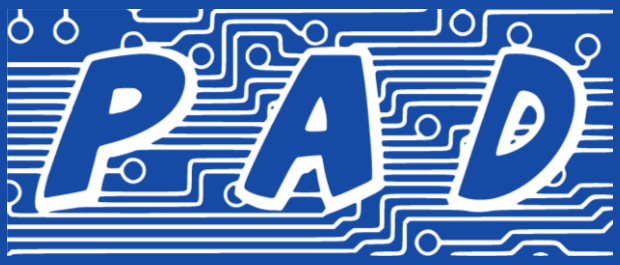
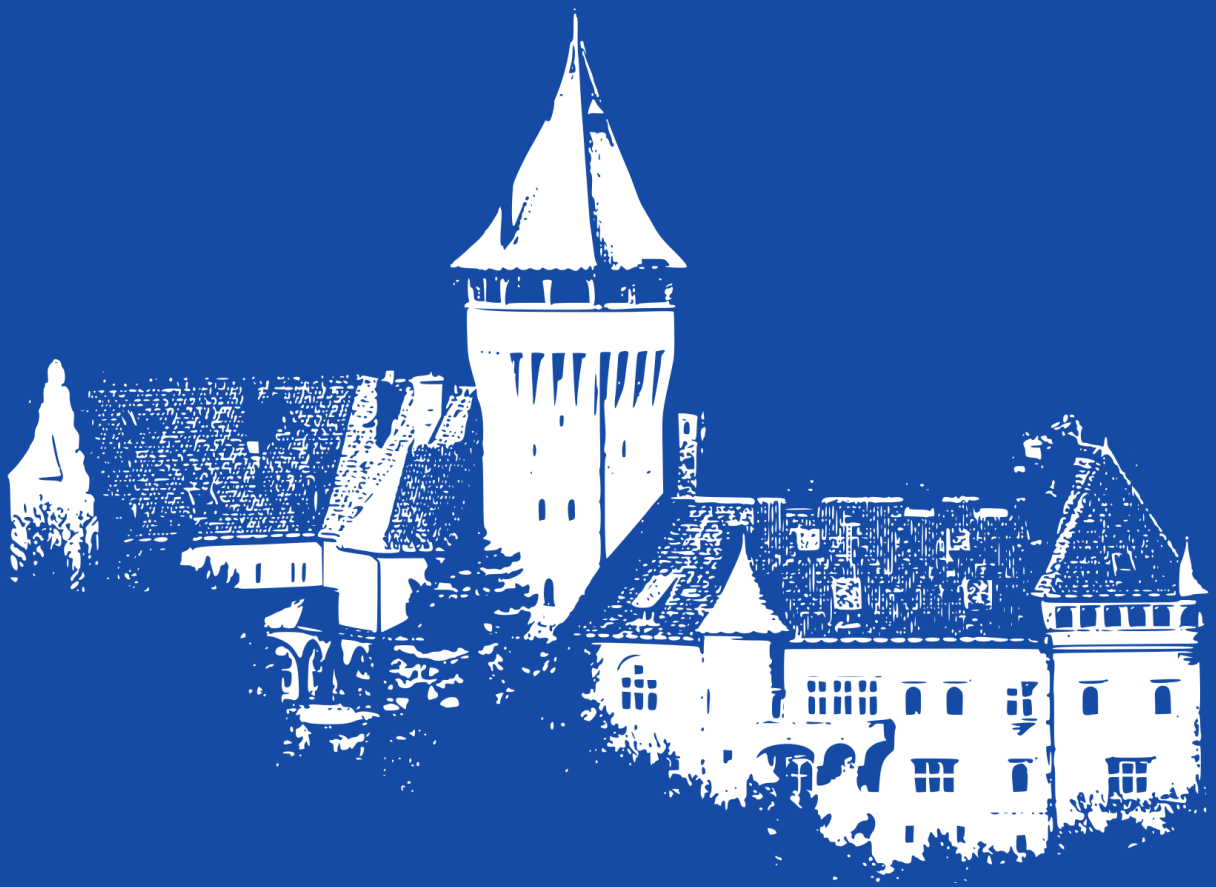


Česko-slovenský seminár
pre študentov
doktorandského štúdia

Počítačové architektúry & diagnostika

PAD 2017

Smolenice, 6. – 8. septembra 2017



Počítačové architektúry & diagnostika PAD 2017

Česko-slovenský seminár pre študentov doktorandského štúdia
Smolenice, 6.– 8. septembra 2017

Zborník príspevkov

Slovenská technická univerzita v Bratislave

Ústav informatiky, Slovenská akadémia vied, Bratislava

Editori zborníka

Juraj Brenkuš
juraj.brenkus@stuba.sk

Viera Stopjaková
viera.stopjakova@stuba.sk

Fakulta elektrotechniky a informatiky
Slovenská technická univerzita v Bratislave

ISBN 978-80-972784-0-3

Všetky príspevky boli vytlačené podľa podkladov dodaných autormi príspevkov.

Návrh obálky: Michal Šovčík

Vydalo:
STU Scientific, s.r.o.
Pionierska 15, 831 02 Bratislava

Programový výbor PAD 2017

Baláž Marcel	ÚI SAV Bratislava
Drábek Vladimír	FIT VUT Brno
Dudáček Karel	FAV ZČU Plzeň
Fišer Petr	FIT ČVUT Praha
Jaroš Jiří	FIT VUT Brno
Jelemenská Katarína	FIIT STU Bratislava
Koutný Tomáš	FAV ZČU Plzeň
Krištofík Štefan	FIIT STU Bratislava
Kubátová Hana	FIT ČVUT Praha
Lórencz Robert	FIT ČVUT Praha
Macko Dominik	FIIT STU Bratislava
Plíva Zdeněk	FMIMS TU Liberec
Rozkovec Martin	FMIMS TU Liberec
Růžička Richard	FIT VUT Brno
Schmidt Jan	FIT ČVUT Praha
Smotlacha Vladimír	FIT ČVUT Praha
Stopjaková Viera	FEI STU Bratislava
Strnadel Josef	FIT VUT Brno
Vlček Karel	UTB Zlín
Zahradnický Tomáš	FIT ČVUT Praha
Zachariášová Marcela	FIT VUT Brno

Organizačný výbor PAD 2017

Jelemenská Katarína	FIIT STU Bratislava
Stopjaková Viera	FEI STU Bratislava
Čičák Pavel	FIIT STU Bratislava
Baláž Marcel	ÚI SAV Bratislava
Brenkuš Juraj	FEI STU Bratislava
Arbet Daniel	FEI STU Bratislava
Šovčík Michal	FEI STU Bratislava
Macko Dominik	FIIT STU Bratislava

Riadiaci výbor PAD

Drábek Vladimír	FIT VUT Brno
Jelemenská Katarína	FIIT STU Bratislava
Růžička Richard	FIT VUT Brno
Kvaček Robert	ASICentrum Praha
Lórencz Robert	FIT ČVUT Praha
Novák Ondřej	TU Liberec
Koutný Tomáš	FAV ZČU Plzeň
Stopjaková Viera	FEI STU Bratislava (predsedníčka RV)

Poďakovanie

Organizátori seminára PAD 2017 ďakujú Fakulte elektrotechniky a informatiky, Fakulte informatiky a informačných technológií STU v Bratislave a Ústavu informatiky SAV, za podporu a pomoc pri organizovaní seminára.

Poďakovanie taktiež patrí sponzorom seminára, spoločnostiam ASICentrum s.r.o, Hewlett-Packard Enterprise Slovakia, s.r.o, Microsoft a Soitron, a.s. za poskytnutú finančnú podporu, ktorou podporili zorganizovanie i priebeh seminára.

Seminár bol podporený aj Agentúrou na podporu výskumu a vývoja prostredníctvom projektu APVV-15-0254.

Organizátori semininára PAD 2017 ďakujú spoločnosti STU Scientific, s.r.o. za administratívne zabezpečenie finančných operácií v rámci organizácie seminára.

Predhovor

V mene organizátorov 15. ročníka česko-slovenského seminára doktorandov Počítačové architektúry a diagnostika – PAD 2017, vítame všetkých jeho účastníkov na malebnom zámku Smolenice, ktorý sme vybrali pre zorganizovanie tohto ročníka. Veríme, že sa nám úspešne podarilo predĺžiť históriu tohto priateľsky príjemného a vedecky podnetného podujatia aj vsadením seminára do lokality pohoria Malých Karpát a krásnej prírody.

História konania seminára PAD sa začala v roku 2003, necelý rok po úmrtí prof. Ing. Jana Hlavičky, DrSc., po ktorom je pomenovaná aj cena udeľovaná doktorandom za vynikajúce výsledky v doktorandskom štúdiu. Cena nesúca meno vedca, ktorý zasvätil svoj profesijný život počítačom a diagnostike, má nepochybne svoju hodnotu a v histórii jej udeľovania sa stala pre doktorandov skutočnou motiváciou. V minulom ročníku (PAD 2016) bola „Cena prof. Ing. Hlavičky, DrSc. za vynikajúce výsledku v doktorandskom štúdiu“ udelená týmto študentom:

1. ročník (12 študentov) – Lukáš Kohútka (FEI STU)
2. ročník (9 študentov) – Ondrej Kachman (UI SAV)
3. ročník 5 študentov)– Adam Crha (FIT VUT)

Mimoriadna cena za excelentný štart do doktorandského štúdia: Filip Kodýtek (FIT ČVUT)

Oceneným doktorandom blahoželáme a veríme, že programový výbor udelí ocenenia aj tento rok.

Seminár PAD je neformálnym a priateľským fórom, na ktorom majú doktorandi možnosť prezentovať vedecké témy a otvorene o nich diskutovať. PAD je pre doktorandov prínosný hlavne možnosťou získania cennej spätnej väzby ohľadne zámeru dizertačnej práce, plnenia stanovených cieľov, vhodnosti zvolených riešení, ako aj možnosti využitia dosiahnutých výsledkov. A práve v tomto tkvie nezameniteľná úloha školiteľov, ktorí nielen zodpovedne zrecenzovali Váš príspevok, ale počas prezentácie príspevku na seminári povedú plodnú a efektívnu debatu. Táto môže na jednej strane poskytnúť zhodnotenie kvality dosiahnutých výsledkov, a na strane druhej aj poukázať na prípadné nedostatky práce a tým odkryť ďalšie možné smerovania. Touto cestou ďakujem všetkým recenzentom za prípravu posudkov a objektívne hodnotenia.

Pre mnohých z nás sa PAD stal každoročnou udalosťou a neoddeliteľnou súčasťou nášho profesijného života, o čom svedčí aj 22 pôvodne prihlásených príspevkov v tomto ročníku. Žiaľ dvaja doktorandi sa napokon PAD seminára nemohli zúčastniť. PAD je pre doktorandov aj zaujímavá forma vedeckej súťaže, v ktorej sa môžu porovnať so študentmi z ostatných univerzít v rámci príslušného ročníka.

Záverom by sme chceli poďakovať sponzorom za ich finančnú podporu, bez ktorej by nebolo možné toto podujatie úspešne zorganizovať. Ďakujeme aj spoločnosti STU Scientific, s.r.o. za administratívnu pomoc a ekonomickú asistenciu. V neposlednom rade by sme chceli poďakovať celému organizačnému výboru a inštitúciám, ktoré PAD 2017 spoločne zorganizovali.

Prajeme Vám všetkým príjemný pobyt v Kongresovom centre SAV na zámku Smolenice a želáme Vám, aby ste načerpali nové podnety pre svoju prácu a pookriali v pokojnom vidieckom prostredí.

V Bratislave, 28.8. 2017

Viera Stopjaková a Katarína Jelemenská
Organizačný výbor PAD 2017

OBSAH

Sekcia 1

Filip Kodýtek:
A COMMON DESIGN FOR PUF AND TRNG BASED ON RING OSCILLATORS 1

Tomáš Apeltauer:
AUTOMATICKÉ TESTOVÁNÍ MODELŮ KYBER-FYZIKÁLNÍCH SYSTÉMŮ 5

Stanislav Jeřábek:
DYNAMICKÁ REKONFIGURACE JAKO OPATŘENÍ PROTI DPA 9

Michal Šovčík:
DIGITÁLNE METÓDY KALIBRÁCIE ANALÓGOVÝCH INTEGROVANÝCH OBVODOV 12

Tomáš Jakubík:
OFDM AND FHSS HYBRID NETWORK 16

Sekcia 2

Marta Cudova:
FRAMEWORK FOR PLANNING, EXECUTING AND MONITORING OF COOPERATING COMPUTATIONS 20

Richard Pánek:
SYSTÉMY ODOLNÉ PROTI PORUCHÁM - METODIKA NÁVRHU ŘADIČE REKONFIGURACE 24

Miroslav Potočný:
USMERŇOVAČE PRE VYSOFREKVENČNÝ ZBERAČ ENERGIE INTEGROVANÝ NA ČIPE 28

Michal Valiček:
ZABEZPEČENIE VNORENÝCH SYSTÉMOV PROTI PORUCHÁM 32

Sekcia 3

Robert Hülle:
GENEROVÁNÍ TESTU S NULOVÝM MASKOVÁNÍM PORUCH 35

Šimon Danko:
KOMUNIKAČNÝ MODUL PRE IMPLANTOVATEĽNÉ SENZORICKÉ SYSTÉMY 39

Vojtěch Miškovský:
ČÍSLICOVÝ NÁVRH SPOJUJÍCÍ ODOLNOST PROTI PORUCHÁM A ODOLNOST PROTI ÚTOKŮM 43

Sekcia 4

Ondrej Kachman:
CONFIGURABLE REPROGRAMMING METHODOLOGY FOR EMBEDDED LOW-POWER DEVICES 47

Karel Szurman:
STATE SYNCHRONIZATION OF FAULTY SOFT CORE PROCESSORS IN RECONFIGURABLE TMR ARCHITECTURE 51

Sekcia 5

Lukáš Kohútka:
HARDVÉROVÝ KERNEL PRE SYSTÉMY REÁLNEHO ČASU 55

Jakub Lojda:
AUTOMATIZACE NÁVRHU SYSTÉMŮ ODOLNÝCH PROTI PORUCHÁM POMOCÍ VYSOKOÚROVŇOVÉ SYNTÉZY 59

Ivo Háleček:
LOGICKÁ SYNTÉZA ZALOŽENÁ NA OBECNÝCH OPERÁTORECH 63

Matej Rakús:
ROZVOJ TECHNÍK NÁVRHU NÍZKO-NAPÄŤOVÝCH INTEGROVANÝCH SYSTÉMOV 67

Vladimír Kunštár:
ZABEZPEČENIE VNORENÝCH SYSTÉMOV S KRITICKOU DOBOU ODOZVY PROTI PORUCHÁM 71

A common design for PUF and TRNG based on ring oscillators

Filip Kodýtek
1st class, full-time study
Supervisor: Róbert Lórencz

Faculty of Information technology
Czech Technical University in Prague
Thákurova 9, 16000, Prague, Czech Republic
kodytfil@fit.cvut.cz

***Abstract*—This contribution deals with a hardware design of a circuit to be used for both Physical Unclonable Function (PUF) and True Random Number Generator (TRNG). The originally designed circuit is based on ring oscillators and was intended to be utilized as PUF. However, as it is shown in this paper, it turned out that the same circuit may also be used for generating true random numbers. The motivation behind using the same circuit for both applications is utilization of resources and designing a universal cryptosystem that can be used for various cryptographic applications. All of our experiments were performed on Digilent Basys 2 FPGA boards (Xilinx Spartan3E-100 CP 132) and the evaluation of the generated random sequences was performed using NIST statistical test suite.**

***Keywords*—Hardware security, physical unclonable function, true random number generator, field-programmable gate array, ring oscillator**

I. Introduction

Digital circuits implemented in Field-programmable gate arrays (FPGAs) often implement security features such as authentication or encryption. Depending on the application, Physical Unclonable Functions (PUFs) can be used for secure authentication or key storage, because numerous security protocols require some secret key that needs to be stored. A complex and expensive secure storages of keys need to be designed in order to ensure a safe and secure storage of cryptographic keys. However, the nonvolatile memory, in which the keys can be stored, can be vulnerable to invasive attacks, since the key is stored in a digital form.

PUFs offer an easy and highly secure solution to the issue of secure storage of cryptographic keys. PUF is a function which provides a response for a given challenge and a physical state of the electronic device it is implemented on. PUFs are based on physical properties that depend on manufacturing variations that make each chip unique. This can be used to generate unique digital fingerprints of devices and distinguish various devices from each other. The main advantage of PUFs is the fact that we can generate the cryptographic key on the fly when it is needed instead of storing it in a memory.

Some of the basic properties which need to be achieved by PUF's outputs are stability (the same or similar responses for the same challenge on one device), uniqueness (different responses for the same challenge among different devices) and randomness (unpredictability of its responses for new challenges or new devices). Due to its properties, a natural basic applications of PUFs are for device identification, authentication and cryptographic key storage.

The PUF designs suitable for FPGAs typically exploit two different sources of randomness, namely delay variations and memory initialization variations. Many devices have embedded SRAM, which is used by the memory-based PUF as a source of randomness that is derived from the power-up SRAM content [2]. However some FPGAs clear their memory after power-up, thereby losing all randomness. Other memory-based PUF variants such as Butterfly PUF [6] or Flip-flop PUF [8] were proposed to avoid this.

Delay-based PUFs exploit the manufacturing variations that influence delays of logic gates and interconnects. Arbiter PUF [7] was one of the first delay-based PUFs, while others include e.g. the Ring Oscillator PUF (ROPUF) [10] and others.

Beside secure storage of cryptographic keys, we also need random numbers for numerous cryptographic protocols which require generation of keys (e.g. key for symmetric cipher), nonces, initialization vectors, salts etc. This implies that a true random number generator (TRNG) producing unpredictable sequences of bits with good statistical properties is necessary, if a high level of security is to be achieved.

This work describes how to utilize a ring oscillator (RO) based circuit originally designed to be used as a PUF for TRNG. The PUF circuit was proposed and published in [4]. The proposed circuit showed good results in terms of good statistical properties, simplicity of design and efficiency. We extend our current work to show that the same design could also be used as TRNG.

This paper is organized as follows. Section II provides a brief description of the ROPUF, that was proposed in [4]. The evaluation method for TRNGs is described in Section III. Section IV presents the results of experiments. The last Section V concludes the paper.

II. The proposed circuit

In this section we provide a brief description of the proposed circuit that was originally intended to be used as PUF. The

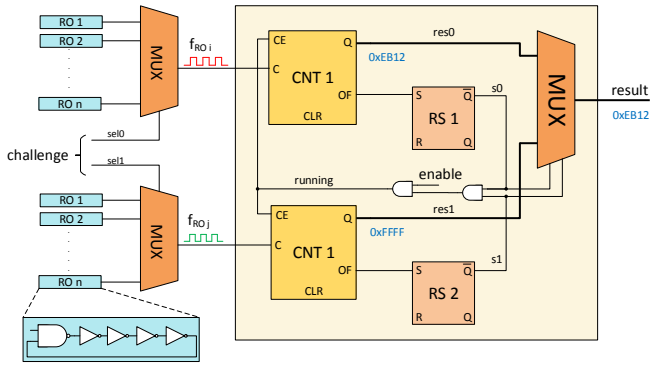


Fig. 1: The proposed ROPUF circuit used for TRNG.

proposed design is based on ROs which serve as the source of the entropy. In case of PUF, the entropy is given by the process variations which determine the differences between the delays of logic gates and interconnects among various ROs and devices. On the other hand, the entropy of TRNG using ROs is determined by jitter of these oscillators.

When proposing the ROPUF design, our main goal was an easy to implement and efficient design. Moreover, the proposed ROPUF provides more output bits from each pair of ROs than the classical approach [10]. This is possible because we use a different approach to extract the PUF output bits. Instead of comparing the frequencies of two mutually symmetric ROs in a pair, we use their frequency ratio. While the classical approach offers only one PUF output bit per RO pair, our approach extracts multiple bits from each RO pair based on the properties of the PUF outputs we want to achieve. The proposed design does not require the ROs to be mutually symmetric; however, the statisitcal properties of the PUF outputs are better with symmetric ROs when exposed to variable physical conditions [5]. The original ROPUF architecture is explained in better details in our previous work [4].

A. The main principle of the proposed ROPUF design

As mentioned before, in the classical approach [10], the frequencies of ROs are compared and the result of this comparison produces only one output bit for PUF. In order to achieve unpredictability of the PUF outputs, this approach requires all ROs to be mutually symmetric so that the differences in frequencies of ROs are influenced only by the random variations in delays of logic gates and their interconnects. As also mentioned in [10], the number of RO pairs for this comparison is limited, so that the bits in the PUF outputs are independent.

In our proposal, the PUF outputs are still obtained based on the selected RO pairs, but the problem of selecting particular RO pairs is no longer present. The ROs in our design consist of 1 NAND and 4 inverters, forming a combinational loop. As shown on Fig. 1, there are 2 sets of ROs and using 2 multiplexers, one RO from each set is selected and its output is connected to a counter. When the ROs are selected, the measurement is started and when one of the counters overflows, the measurement is stopped. The other counter (the one that did not overflow) contains a resulting value which is used for further processing.

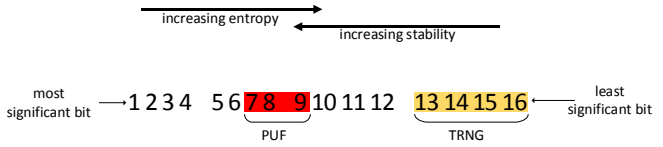


Fig. 2: The example behaviour of the bits in counter value of a 16-bit value.

The proposed method implies that if we knew the exact frequencies of the ROs during measurement, we could determine the resulting counter value (in case of 16-bit counters) that is later processed as follows:

$$Counter\ value = \frac{f_2}{f_1} \times 2^{16}, \tag{1}$$

where f_1 is the frequency of the faster RO and f_2 is the frequency of the slower RO.

B. Utilization of the proposed design for PUF and TRNG

The counter values are represented in binary code and therefore, we can select an appropriate part of the counter values for the PUF output based on the statistical properties of the selected bit positions. It can be expected that the positions close to the most significant bit (MSB) will be very stable, while the positions close to the least significant bit (LSB) will be highly unstable and would vary with almost each measurement due to noise effects - a possible source of entropy for TRNG. The described behaviour is depicted in Fig 2.

Taking into account the desired properties of PUFs, the suitable positions for PUF are located somewhere in the middle of the counter value, where both entropy and stability are high. It is important to realize that the sources of entropy for PUF and TRNG differ significantly. The entropy for PUF is given to a particular circuit only once during the manufacturing process. Therefore, the entropy is determined for given positions among various RO pairs or the same RO pair but on various devices. On the other hand, the entropy of TRNG is given by the jitter of ROs, so the entropy is determined for each RO pair using multiple measurements. This implies that the suitable positions for TRNG will be the ones close to the LSB.

III. TRNG EVALUATION

There are various statistical properties of TRNG designs that we can evaluate, such as bit rate, area efficiency or sensitivity to physical disturbances. However, the most important property of any TRNG is the unpredictability of its output. Therefore, we should carefully evaluate the TRNG in the perspective of randomness it offers.

There are some recommendations or guidelines how to evaluate TRNGs, e.g. the German document AIS 31 [3]. In order to evaluate TRNG properly, it is not sufficient to test only the generated sequences of bits by the TRNG, because even a deterministic random number generator can pass test suites such as NIST or DieHard. These tests may be necessary,

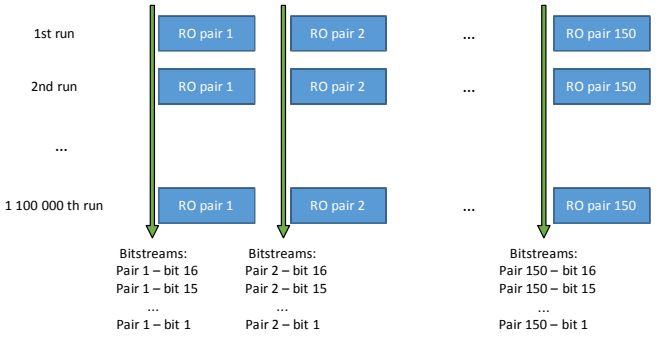


Fig. 3: An example of forming a random sequence from single bits from individual RO pairs.

but even if the TRNG pass these tests, it doesn't mean that it really is a TRNG.

The problem is that the generated sequences are already digitized and we evaluate them after some algorithmic post-processing which enhances its statistical properties. But what we need to do is to make a stochastic model of the noise and compute a lower bound of the entropy per bit of the source of the entropy [1].

First, we need to identify the source of randomness [1]. TRNG rely on a random physical phenomenon known as analog physical noise. Therefore, analog physical noise is the source of randomness we need to identify. There might also be some other unidentified phenomena which would contribute to the randomness of TRNG, but it shouldn't be taken into account in entropy estimation. After identifying the source of randomness, we need to make a statistical model for the physical noise used.

Having the statistical model for physical noise, one must be able to evaluate experimentally the parameters of the model and evaluate the measurement errors of these parameters. Also, the stability of parameters of the statistical model must be evaluated for physical noise with regard to physical environmental conditions of the TRNG (temperature, supply voltage...) and technological environmental operating conditions of the TRNG (installed alone on a circuit or with other circuits).

The next requirement in order to evaluate the TRNG properly is to have a statistical model for the TRNG (i.e. the bits it generates). It is assumed, that all of the conditions mentioned above are fulfilled, because the statistical model for the physical noise is needed. To ensure that the TRNG is working properly during its life, parametric tests must run at startup and continuously.

IV. Experimental results

In this section we present the results of testing the proposed design as a TRNG. All of the measurements were performed on Digilent Basys 2 FPGA board containing Xilinx Spartan3E-100 CP132. There were two separate sets of measurements. The first one set of measurements was performed on a circuit containing 150 RO pairs, where all of the ROs were running during the measurement. An on-board switching regulator was used as the power supply. In the second set of measurements, we used a circuit with 130 RO pairs where the ring oscillators

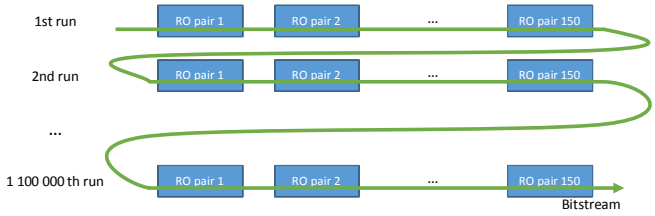


Fig. 4: Concatenating the outputs from all RO pairs to form one long random sequence of bits.

run and are measured separately. In this case, we modified the Digilent Basys 2 FPGA board so that the original power supply is disconnected and replaced with a new power supply consisting of a battery and linear regulators.

The measured sequences of bits were evaluated using NIST statistical test suite [9] that was proposed specifically to test random number generator for cryptographic purposes. The version of the NIST software we used is STS 2.1.2. This test suite consists of tests such as frequency test, runs test, cumulative sums test, entropy test, etc.

A. Individual RO pairs tests

In this experiment, we examined single bits from each RO pair and we considered each RO pair as a unique source of entropy. Each RO pair's counter value was measured 1 100 000 times. Therefore, we obtained 150×16 bit streams as shown in Fig. 3. Some of the tests in NIST STS 2.1.2 require longer bit streams than we could provide, which led us to exclude such tests.

The results of the tests are as follows: Bit 16 (LSB) failed in some tests, but the positions 15 and 14 show passed all of the tests indicating that these bits may be suitable to be used for TRNG output. All of the other positions failed all of the statistical tests except for bit 13 which failed only in some tests.

B. Concatenated RO pairs outputs tests

The previous experiment indicated that each RO pair could be used as a stand alone source of entropy. However, it would be more natural for this particular design to use multiple bits from each measured counter value for the TRNG output. Moreover, since there are 150 RO pairs, we can concatenate their outputs to form a single long bit stream as shown in Fig. 4.

The concatenated bit streams were then tested. After this concatenation, we had enough data to run all of the tests from NIST STS. We tested both concatenated single bits from all RO pairs and concatenated multiple bits from all RO pairs. Moreover, since some of the selections of positions failed for frequency test, indicating bias in the TRNG output (this can happen when dealing with TRNG), we used 2 post-processing methods: XOR corrector and Von Neumann corrector.

Von Neumann corrector works as follows: If the input is "00" or "11", the input is discarded , if the input is "10", the output is "1" and finally if the input is "01", the output is "0". The disadvantage of this post-processing method is the

	Concatenated bits			
	Von Neumann corrector		XOR corrector	
Position	0–3	1–3	0–3	1–3
Frequency	69/100	99/100	82/100	98/100
Block Frequency	100/100	99/100	99/100	100/100
Cumulative Sums I	69/100	100/100	85/100	99/100
Cumulative Sums II	71/100	99/100	83/100	97/100
Runs	96/100	99/100	96/100	100/100
Longest Run	98/100	99/100	99/100	99/100
Rank	99/100	98/100	99/100	99/100
FFT	99/100	99/100	100/100	98/100
Non Overlapping Template	98–100/100	97–100/100	98–100/100	97–100/100
Overlapping Template	99/100	100/100	99/100	100/100
Universal	98/100	100/100	99/100	100/100
Approximate Entropy	100/100	100/100	98/100	100/100
Random Excursions	30–31/30	62–63/63	43–44/100	55/55
Random Excursions Variants	30–31/30	62–63/63	43–44/100	55/55
Serial I	99/100	99/100	100/100	100/100
Serial II	100/100	100/100	98/100	98/100
Linear Complexity	100/100	100/100	98/100	97/100

TABLE I: Results of NIST STS tests of concatenated bit stream after applying Von Neumann and XOR correctors. Minimum allowed pass rate is 96/100. The red cells indicate that the test failed for the distribution of p-values.

fact that it shortens the generated sequence by approximately 75%.

On the other hand, XOR corrector shortens the generated sequence only by 50%. It takes two subsequent bits from the input and puts the result of XOR operation on these two bits into the generated sequence.

The results of the NIST STS after applying Von Neumann and XOR correctors are shown in Table I. This table shows the pass rates for each of the tests and the red cells show that the test failed for the distribution of p-values. As can be seen in Table I, the bits 15–13 show excellent behaviour after applying these post-processing methods.

C. Ruling out crosstalk and parasitic frequencies

To eliminate any potential crosstalks between individual RO pairs and parasitic frequencies from switching regulators influencing randomness of generated bitstream and therefore to verify that each RO pair can be considered as a unique source of entropy, we tested individual RO pairs separately using a set of linear regulators as power supply. In this experiment, all of the ROs were not running simultaneously as before, but only one selected pair of ROs is running during the measurement. Otherwise, the setup of the experiment remains the same. Each RO pair was measured 1 100 000 times and there were 130 RO pairs on the examined circuit. The results of this experiment are very similar to those, where all ROs were running simultaneously and switching regulator was used. Therefore, we can assume that each individual RO pair is a unique source of entropy.

V. Conclusion

PUFs and TRNGs are two different cryptographic primitives that have one thing in common. They both exploit some random physical phenomenon. TRNG needs a continuous random phenomenon, e.g. noise. On the contrary, PUFs exploit a random variation in the manufacturing process where the randomness appears only once and defines the physical properties of the device.

In this work, we dealt with a circuit originally designated to be used as ROPUF and showed that it has a potential to be also utilized as a TRNG. For the evaluation of the TRNG, we used NIST statistical test suite. We tested two different experimental setups. One with RO pairs running simultaneously and switching regulator as a power supply, while the second one with only one RO pair running at a time and linear regulator as a power supply. The results have shown that up to three bits can be extracted from each RO pair from one measurement, but further post-processing is required, which causes the generated random sequence to be shorter by 50% in case of XOR corrector and by approximately 75% in case of Von Neumann corrector.

In our future work, we would like to evaluate the proposed TRNG in accordance with the methodology described in Section III. It means that at first, we need to build a statistical model of the physical noise that serves as the source of the entropy. Also, the behaviour of the proposed TRNG needs to be evaluated under varying physical conditions. Moreover, some online tests need to be used in order to detect the failure of the TRNG.

Acknowledgement

This research has been partially supported by the grant GA16-05179S of the Czech Grant Agency, Fault-Tolerant and Attack-Resistant Architectures Based on Programmable Devices: Research of Interplay and Common Features (2016-2018) and SGS17/214/OHK3/3T/18, “Studentská grantová soutěž ČVUT v Praze (2017)”.

References

[1] Fischer, V. Design and evaluation of a physical random number generator. In *Cryptographic architectures embedded in logic devices*, Smolenice, SK, June 2017.

[2] Holcomb, D. E., Burleson, W. P., Fu, K. Power-Up SRAM State as an Identifying Fingerprint and Source of True Random Numbers. *IEEE Transactions on Computers 58, 9*, pages 1198-1210, 2009.

[3] Killmann, W., Schindler, W. A Proposal for: Functionality Classes and Evaluation Methodology for True (Physical) Random Number Generators, Version 3.1, English translation, 25.09.2001

[4] Kodýtek, F., Lórencz, R. A design of ring oscillator based PUF on FPGA. In *IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems - DDECS 2015*. Belgrade, RS, April 2015.

[5] Kodýtek, F., Lórencz, R., Buček, J. Improved ring oscillator PUF on FPGA and its properties. In *Microprocessors and Microsystems*. 2016.

[6] Kumar, S., Guajardo, J., Maes, R., Schrijen, G.-J., Tuyls, P. Extended abstract: The Butterfly PUF Protecting IP on Every FPGA. In *IEEE International Symposium on Hardware-Oriented Security and Trust - HOST 2008*, pages 67–70. IEEE, Washington, DC, USA, 2008.

[7] Lee, J. W., Lim, D., Gassend, B., Suh, G.E., van Dijk, M., Devadas, S. A technique to build a secret key in integrated circuits for identification and authentication applications. In *Symposium on VLSI Circuits - VLSIC 2004*, pages 176–179, June 2004.

[8] Maes, R., Tuyls, P., Verbauwhede, I. Intrinsic PUFs from Flip-flops on Reconfigurable Devices. In *Benelux Workshop on Information and System Security - WISSec 2008*. Eindhoven, NL, 2008.

[9] Ruhkin, A. et al. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. *NIST Special Publication 800-22 Revision 1a*. 2010.

[10] Suh, G. E., Devadas, S. Physical Unclonable Functions for Device Authentication and Secret Key Generation. In *Design Automation Conference - DAC 2007*, pages 9–14. ACM, New York, NY, USA, 2007.

Automatické testování modelů kyber-fyzikálních systémů

Tomáš Apeltauer
1. ročník prezenčního studia
Školitel: Stefan Ratschan

České vysoké učení technické v Praze, Fakulta informačních technologií
Thákurova 9, Praha, 16000
apelttom@fit.cvut.cz

***Abstrakt*—Článek pojednává o modelech kyber-fyzikálních systémů a problematice jejich testování. Sleduje současný trend průmyslu, čím dál častějšího prolínání fyzikálního světa se světem výpočtů a reaguje na vzrůstající potřebu tvořit nové, složitější systémy, které již dokáží monitorovat a ovlivňovat skutečný svět kolem nás. Samotná tvorba kyber-fyzikálních systémů představuje netriviální problém, ale díky spojení s metodikou Model-Based design a automatickým generováním testů se z ní stává atraktivní oblast aktivního výzkumu. Současné analytické nástroje bohužel nezvládají plně pokrýt komplexitu kyber-fyzikálních systémů, nebo adekvátně predikovat jejich chování. Řešením je tvorba takových algoritmů pro automatické testování modelů kyber-fyzikálních systémů, které by uměly využívat vnitřní strukturu modelu, ale zároveň by fungovaly v případě, že rozšíříme model o prvky, které se běžně vyskytují v průmyslové praxi.**

***Klíčová slova*—kyber-fyzikální systémy, testování, Model-Based design, hybridní systém, hybridní dynamický model, Zenónův běh, validace, verifikace, Simulink**

I. Úvod

Pro systémy, které kombinují fyzikální svět se světem výpočtů, používáme pojem kyber-fyzikální systém [1]. Tato těsná interakce fyzikálního světa a světa výpočtů má za následek vyšší složitost kyber-fyzikálních systémů, protože spojením obou oblastní dostáváme mnohonásobně větší množinu stavů a situací, do kterých se může kyber-fyzikální systém dostat a obsáhnout je všechny není možné.

A. Problematika testování kyber-fyzikálních systémů

Navzdory významnému pokroku v rámci technologie kyber-fyzikálních systémů, stále postrádáme dostatečně vyspělý výzkum, který by zaštítil oblast vysoce spolehlivých kyber-fyzikální systémů. Důsledkem toho nezvládají současné analytické nástroje plně pokrýt komplexitu kyber-fyzikálních systémů, nebo adekvátně predikovat jejich chování. Příkladem je Internet věcí, který se neustále rozmáhá, a jenž má potenciál škálovat do úrovní biliónů propojených zařízení, která dokáží monitorovat, kontrolovat i jinak interagovat s fyzickým prostředím okolo nás. Přirozeně jsou zde kladeny vysoké nároky na spolehlivost, bezpečnost a robustnost takových systémů [2].

B. Význam oblasti testování kyber-fyzikálních systémů

V průmyslové sféře je hojně využíván přístup tvorby a aplikace abstraktních modelů v procesu návrhu (Model-Based design [3]). Model-Based design nám umožňuje simulovat, testovat a verifikovat výsledný systém už v raných fázích procesu návrhu. Příkladem nástrojů z praxe muže být software *MATLAB/Simulink*, *Statemate*, nebo software *Modelica*, příkladem nástroje z akademického prostředí je *Ptolemy* (UC Berkeley).

Softwarové komponenty proto již nejsou výhradně psány pouze v C, nebo Assembleru, ale stále častěji modelovány pomocí výše zmíněných nástrojů a tak nabývá na významu i oblast testování těchto modelů [4]. Současně existuje velmi silná motivace proces testování automatizovat a snížit tak náklady na vývoj modelů kyber-fyzikálních systémů. Navíc bychom tím dokázali zvýšit použitelnost již vytvořených testovacích scénářů.

C. Obtížnost problematiky testování

Hlavním zdrojem obtížnosti v oblasti testování modelů kyber-fyzikálních systémů je složitost nástrojů, velký počet různých toolboxů a absence jasně definované, standardizované formální sémantiky v programech jako je například Simulink. Modely v tomto nástroji se sestávají z funkčních bloků a každý z nich má jasně definované vstupní i výstupní kanály. Tyto bloky nefungují izolovaně ale mohou si předávat data pomocí námi určených komunikačních rozhranní, navíc je možné modely hierarchicky strukturovat, protože jeden funkční blok lze reprezentovat i jako množinu podbloků a jejich rozhranní. Takto lze v programu Simulink vytvářet složité komplexní modely, které reálně reprezentují v praxi využívané systémy.

II. Vymezení oblasti výzkumu

Pro modely vytvořené softwarem Simulink zatím existují pouze black box algoritmy, nebo toolboxy s omezenou funkcionalitou, například *T-Vec Tester* a *Reactive Systems Reactis Tester* [5], [6]. Tyto toolboxy pracují s funkčními bloky Simulinku bez nutné znalosti vnitřní hierarchické struktury, staví zejména na definovaném komunikačním rozhranní a dále

pak na jasně formulované specifikaci systému. Navíc pro svou optimalizaci využívají algoritmy black-box optimalizace [7].

A. Omezení

Tyto nástroje mají své praktické využití při verifikaci konzistence modelů z pohledu validní manipulace s daty (dělení nulou, přetečení), nebo při kontrole metrik jako jsou *state coverage*, *branch coverage* a hlavně *MD/DC coverage*, ale pro otestování modelů pod intenzivní zátěží, v situacích simulujících pokud možno co nejvěrohodněji reálné případy užití, jsou tyto nástroje nedostačující. Dokáží generovat takové testy, aby dosáhli vysoké míry pokrytí kódu a jsou schopné pracovat i se stavovými diagramy. Bohužel neuvažují vnitřní strukturu modelů, použité materiály, fyzikální veličiny a zákony, což se v praxi může projevit selháním systému za určitých specifických okolností. Navíc jsou omezeny velikostí generovaných testů (potažmo délkou generovaných signálů) a zvládnou zpracovat modely jen do určité míry složitosti.

B. White-box testing

Algoritmy, které využívají vnitřní strukturu modelů a jsou určeny pro testování a verifikaci, zatím existují pouze pro hybridní dynamické systémy [8]–[11]. Model hybridního dynamického systému vznikl v akademické obci právě pro účely automatického testování modelů [12]. Modely jsou jednodušší než ty vytvořené v Simulinku, ale mají jasnou sémantiku.

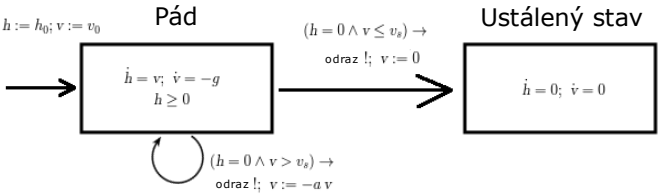
III. Cíle práce

Cílem mé disertační práce je vývoj algoritmů pro automatické testování kyber-fyzikálních systémů nad modely softwarových nástrojů, běžně používaných v praxi. Algoritmy by měly využívat vnitřní strukturu modelu, ale zároveň by měly fungovat v případě, že rozšíříme modely o prvky, které se běžně vyskytují v průmyslové praxi. Věnuji se snaze o aplikování obecných algoritmů pro testování a verifikaci hybridních dynamických systémů na modely, jenž nemají jasně definovanou formální sémantiku.

IV. Hybridní systémy

Hybridní dynamické modely používáme pro modelování hybridních systémů, které obsahují jak spojitou část, jejíž vývoj závisí na čase, tak diskrétní část. Hybridní dynamické modely nám umožňují lépe pracovat se spojitým světem a změnami závislými na čase, pomocí diferenciálních a algebraických rovnic. Tato vlastnost naneštěstí komplikuje automatizaci testování a verifikaci abstraktních modelů.

Hybridní dynamický model můžeme reprezentovat pomocí hybridního stavového automatu, který vychází z klasického stavového automatu. Pro zobrazení používáme stavový diagram, jak je ukázáno na obrázku 1. Diskrétní část systému je zobrazena pomocí stavů a přechodů mezi nimi. Je definován počáteční stav. Přechody jsou definovány pomocí přechodových podmínek, ve kterých mohou figurovat předem definované konstanty. Dále se tu objevují proměnné typu `cont`, které nabývají hodnot z množiny reálných čísel (nebo intervalu reálných čísel) a jsou aktualizovány spojitě spolu s ubíhajícím časem, zatímco proces čeká v určitém stavu.



Obrázek 1: Hybridní model odrážejícího se míče

A. Hybridní proces

Hybridní dynamické modely se úzce pojí s pojmem hybridního procesu, a během hybridního procesu, který nám poskytuje nástroj pro vyjádření takového modelu v čase. Hybridní proces se skládá z: [1]

1) Asynchronního procesu *P*, kde jsou některé vstupní, výstupní a stavové proměnné typu `cont`
2) Časově-spojitého invariantu *CI*, reprezentovaného booleovským výrazem nad stavovou proměnnou *S*
3) Pro každou vstupní proměnnou y typu `cont`, výrazem ohodnocení h_y nad stavovými a vstupními proměnnými typu `cont`
4) Pro každou stavovou proměnnou x typu `cont`, výrazem ohodnocení f_x nad stavovými a vstupními proměnnými typu `cont`

Dále platí, že vstupy, výstupy, stavy, počáteční stavy, vnitřní děje, vstupní děje a výstupní děje hybridního procesu *HP* jsou stejné, jako u asynchronního procesu *P*. Pro daný stav s, časový úsek $\delta > 0$ a vstupní signál $\bar{u}$ pro každou vstupní proměnnou u typu `cont` na intervalu $[0, \delta]$ je odpovídajícím časovým dějem procesu *HP* diferenciovatelný stavový signál $\bar{S}$ nad stavovými proměnnými a signál $\bar{y}$ pro každou vstupní proměnnou y typu `cont` nad intervalem $[0, \delta]$ takový, že: [1]

1) Pro každou stavovou proměnnou $x, \bar{x}(0) = s(x)$
2) Pro každou diskrétní stavovou proměnnou x a čas $0 \leq t \leq \delta, \bar{x}(t) = s(x)$
3) Pro každou výstupní proměnnou y typu `cont` a čas $0 \leq t \leq \delta$ se $\bar{y}(t)$ rovná h_y vyhodnoceného pomocí hodnot $\bar{u}(t)$ a $\bar{S}(t)$
4) Pro každou každou stavovou proměnnou x typu `cont` a čas $0 \leq t \leq \delta$ se derivát času $(d/dt)\bar{x}(t)$ rovná f_x vyhodnocené na základě hodnot $\bar{u}(t)$ a $\bar{S}(t)$
5) Pro všechna $0 \leq t \leq \delta$ splňují časově-spojitý invariant CI hodnoty $\bar{S}(t)$ nad stavovými proměnnými v čase t

B. Zenónův běh hybridního procesu

Nekonečný běh hybridního procesu *HP* se nazývá *Zenónovým během*, pokud je suma časových úseků všech měřených dějů v daném běhu ohraničena konstantou. Stav s, náležící hybridnímu procesu *HP* se nazývá *Zenónovým stavem*, pokud každý konečný běh, který obsahuje stav s je

Zenónovým během. Hybridní proces *HP* se nazývá *Zenónovým procesem*, pokud obsahuje stav s, který je dosažitelný a zároveň je Zenónovým stavem.

Pokud vyvozujeme závěry za pomoci Zenónových běhů hybridních procesů, nedospějeme ke korektním závěrům. Přítomnost jediného Zenónova procesu může mít nepředvídatelný vliv na analýzu celého systému, proto bychom se měli Zenónovým komponentám během formálního modelování vyhnout. Zenónův proces lze převést do tvaru, který nevyhnutelně nevyžaduje přepínání stavů po stále kratší a kratší době, čímž lze Zenónovu vlastnost odstranit. [1]

C. Stabilita hybridních systémů

Další z důležitých vlastností hybridních procesů je jejich stabilita. Vzhledem k faktu, že hybridní procesy obsahují přepínání stavů, není možné použít v tomto případě matematickou analýzu, užívanou pro charakteristiku stability lineárních systémů, ani přidružené techniky pro návrh stabilizačních kontrol. Analýza stability hybridních systémů je velmi náročný problém. Využívání analyzačních technik z teorie spojitých systémů na hybridní systémy zůstává aktivní oblastí vědeckého výzkumu. [1]

V. Simulink

Modelování kyber-fyzikálních systémů není otázka čistě akademická, ale je často využíváno i v praxi. Nejrozšířenějším nástrojem pro Model-based design v průmyslu je software Simulink. Softwarový nástroj nabízí obsáhlou knihovnu komponent s jejíž pomocí jsme schopni systém popsat, většinou skrze matematické rovnice a algebraické operace.

A. Metodika modelování

Proces modelování je rozdělen do několika fází:

1) Stanovení cílů a požadavků na model (jaké otázky nám zodpoví, požadavky na přesnost, definice problému)
2) Vymezení systémových komponent (identifikování fyzikální a kybernetické části modelu, vztahy mezi komponentami)
3) Definice rovnic popisujících systém (v případě kyber-fyzikálních systémů se často jedná o diferenciální rovnice)
4) Tvorba sady parametrů (seznam konstant, koeficientů a jejich hodnot - získané např. měřením)
5) Proces tvorby modelu (v software Simulink pomocí grafické reprezentace)
   - Vytvoření bloku pro danou komponentu
   - Validace komponenty pomocí simulace chování komponenty
6) Integrace komponent mezi sebou a validace jejich vzájemné spolupráce (využití simulace)

Proces validace komponenty pomocí simulace, případně pomocí kontroly formálních požadavků již z velké části pokrývá balíček *Simulink Verification & Validation Toolbox*. Ten dokáže automaticky kontrolovat požadavky kladené na komponenty, validovat oproti průmyslovým standardům (*ISO 26262*, *DO-1788*) a kontrolovat shodu oproti formálnímu popisu. Nepracuje však se všemi prvky, které jsou v praxi nezbytné, což otevírá dveře dalšímu výzkumu.

B. Integrační testování

Otázka integračního testování v nástroji Simulink je velmi komplexní a zahrnuje užívání formálních metod, jako jsou MC/DC pokrytí, nebo automatické generování testů. I přes velkou snahu nedosahují současné techniky požadované kvality a míry pokrytí proměnných.

VI. Závěr

Článek nastínil problematiku složitosti automatického testování modelů kyber-fyzikálních systémů a představil oblasti, které z akademického hlediska nabízí zajímavé a jen částečně probádané problémy. Zmíněna byla i motivace, která stojí za úsilím objevit a zformulovat praktiky pro automatické testování a případný dopad do praxe.

V druhé části článku je představen pojem hybridního dynamického modelu, s ním spojený pojem hybridního procesu, vlastnosti Zenónova běhu procesu a nakonec i netriviální otázka, jenž se týká stability hybridních systémů. Dále je popsán konkrétní softwarový nástroj Simulink, hojně využívaný zejména v průmyslové sféře. Byly identifikovány možnosti testování, které nástroj sám nabízí. Zmíněny byly také omezení nástroje Verification & Validation Toolbox a otevřené otázky, jenž by si zasloužily hlubší analýzu.

A. Další směřování výzkumu

Výzkum bude nadále mapovat oblast nástrojů používaných pro Model-Based developement, jakými jsou např. TALIRO-TOOLS, Statemate, MATRIXX, LabVIEW, JModelica.org, nebo Ptolemy a zároveň bude hledat nové způsoby využití již existujících algoritmů pro testování hybridních dynamických systémů v problematice testování modelů bez jasné formální sémantiky. Součástí výzkumu budou otázky detekce Zenónových běhů, jejich transformace a využití analytických technik z teorie spojitých systémů pro stabilitu hybridních systémů.

Poděkování

Tento výzkum byl částečně podporován z projektu ČVUT `SGS17/213/OHK3/3T/18`.

Literatura

[1] R. Alur, *Principles of cyber-physical systems*. Cambridge, Massachusetts: The MIT Press, 2015.

[2] "Cyber-physical systems nsf - national science foundation," https://www.nsf.gov/funding/pgm_summ.jsp?pims_id=503286, accessed: 2017-04-20.

[3] J. C. Jensen, D. H. Chang, and E. A. Lee, "A model-based design methodology for cyber-physical systems," in *2011 7th International Wireless Communications and Mobile Computing Conference*, July 2011, pp. 1666–1671.

[4] E. Bringmann and A. Krämer, "Model-based testing of automotive systems," in *2008 1st International Conference on Software Testing, Verification, and Validation*, April 2008, pp. 485–493.

[5] M. R. Blackburn and R. D. Busser, "T-vec: a tool for developing critical systems," in *Computer Assurance, 1996. COMPASS '96, Systems Integrity. Software Safety. Process Security. Proceedings of the Eleventh Annual Conference on*, Jun 1996, pp. 237–249.

[6] S. Sims and D. C. DuVarney, "Experience report: The reactis validation tool," *SIGPLAN Not.*, vol. 42, no. 9, pp. 137–140, Oct. 2007. [Online]. Available: http://doi.acm.org/10.1145/1291220.1291172
[7] M. Gendreau and J.-Y. Potvin, *Handbook of metaheuristics*. New York: Springer, 2010, vol. 2.
[8] T. Dang and T. Nahhal, "Coverage-guided test generation for continuous and hybrid systems," *Formal Methods in System Design*, vol. 34, no. 2, pp. 183–213, 2009.
[9] E. Plaku, L. E. Kavraki, and M. Y. Vardi, "Falsification of ltl safety properties in hybrid systems," *International Journal on Software Tools for Technology Transfer*, vol. 15, no. 4, pp. 305–320, 2013.
[10] A. Zutshi, S. Sankaranarayanan, J. V. Deshmukh, and J. Kapinski, "A trajectory splicing approach to concretizing counterexamples for hybrid systems," in *52nd IEEE Conference on Decision and Control*, Dec 2013, pp. 3918–3925.
[11] J. Kuřátko and S. Ratschan, *Combined Global and Local Search for the Falsification of Hybrid Systems*. Cham: Springer International Publishing, 2014, pp. 146–160.
[12] A. van der Schaft and J. Schumacher, *An Introduction to Hybrid Dynamical Systems*. New York: Springer, 2000.

Dynamická rekonfigurace jako opatření proti DPA

Stanislav Jeřábek
První ročník, prezenční studium
Jan Schmidt, Martin Novotný

Czech Technical University in Prague, Faculty of Information Technology
Thákurova 9, 160 00 Praha 6
jerabst1@fit.cvut.cz

***Abstrakt*—Tato práce pojednává o směřování výzkumu v rámci tématu dizertační práce věnující se mu bezpečným a spolehlivým architekturám pro programovatelný hardware, především FPGA. Konkrétně práce pojednává o již existující implementaci šifry PRESENT na FPGA, kde je použita dynamická rekonfigurace jako jedno z opatření proti útoku pomocí rozdílové odběrové analýzy. Dílčími cíli výzkumu jsou v první fázi reimplementace výše zmíněné práce, prozkoumání jejích vlastností a vliv úprav parametrů na bezpečnost a spolehlivost návrhu. Dalšími kroky jsou prošetření vlivu navrhovaných nových způsobů užití dynamické rekonfigurace pro zvýšení bezpečnosti návrhu, konkrétněji odolnosti vůči rozdílové odběrové analýze. Nakonec bychom se pak rádi věnovali také vyšetření těchto modifikací z pohledu spolehlivosti obvodu a možnostem editace návrhu na nižší úrovni, tedy úpravy mapování obvodu přímo v bitstreamu.**

***Klíčová slova*—dynamická rekonfigurace, FPGA, DPA, bezpečnost a spolehlivost, XDL.**

I. Motivace

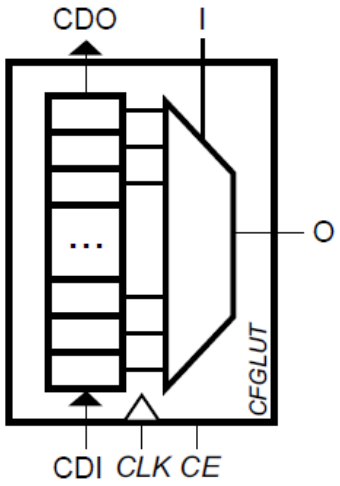
Ideální návrh obvodu je návrh spolehlivý a současně bezpečný. Bohužel v mnoha případech se tyto vlastnosti v jisté míře navzájem potlačují. Spousta metod, jak docílit vyšší spolehlivosti, ve výsledku přináší jistou režii, ať už časovou či prostorovou, a s tím také více možností pro potencionálního útočníka obvod napadnout. Metody zajišťující bezpečnost (security) také přináší obdobné režijní náklady a výsledný složitější obvod je tak náchylnější k výskytům poruch. Vydáme se cestou bezpečnosti a pokusíme se pak doladit spolehlivost.

Spolehlivost a bezpečnost jsou kritické vlastnosti všech hardwarových návrhů. Nicméně snaha o vylepšení jedné z těchto vlastností velmi často způsobuje zhoršení té druhé [1], [2]. Náš cíl je najít novou metodu návrhu pro programovatelný hardware, který by vylepšoval jak spolehlivost, tak bezpečnost, nebo alespoň jednu z těchto vlastností aniž by ta druhá byla zhoršená.

V této práci se vydáme cestou zlepšení bezpečnosti a budeme zkoumat, jaký mají vliv tyto úpravy na spolehlivost výsledného návrhu. Následně se pokusíme pomocí editace návrhu pro FPGA na nízké úrovni (jazyk XDL [3]). Chystáme se použít dynamickou rekonfiguraci při návrhu šifry nenáročné na výpočetní prostředky, konkrétně PRESENT [4], pro dosažení vyšší odolnosti vůči útokům rozdílovou odběrovou analýzou. Budeme znovu implementovat metodu popsanou v již publikovaném článku [5]. Poté prozkoumáme vliv některých úprav této metody, naimplementujeme náš nový způsob použití dynamické rekonfigurace v kombinaci s metodou ukrývání v čase [6], a také prošetříme chování obvodu při použití různých kombinací nově navrhovaných i dříve publikovaných metod. Poznatky získané při zkoumání vlivů všech výše popsaných úprav na bezpečnost a spolehlivost implementace nenáročné šifry PRESENT pak doufejme budou moci být alespoň částečně zobecněny pro návrh číslicových obvodů.

A. Platforma pro implementaci a měření

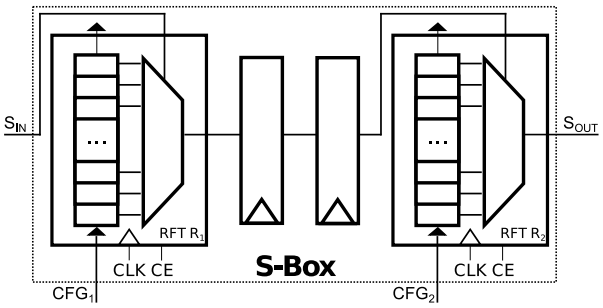
Celý výzkum bude implementován a také rovnou měřen na experimentální desce s FPGA čipem určené pro kryptografické aplikace [7]. Tato deska je vybavena čipem Xilinx Artix–7 FPGA [8] obsahující LUTy se šesti vstupy, které mohou být použity pro umístění primitiv s názvem CFGLUT5 primitive [9]. Tato primitiva vnitřně obsahují bloky distribuované paměti, které jsou schopny se chovat jako posuvný registr, jak je ukázáno na Obrázku 1. Tyto bloky pak slouží právě pro implementaci dynamické rekonfigurace, kdy je pomocí posunu hodnot do řídící logiky pro výstupní funkci měněna funkce, kterou celý blok vykonává, za běhu zařízení bez jakéhokoli vnějšího zásahu. Tato deska vznikla na naší katedře právě pro potřeby měření útoků rozdílovou odběrovou analýzou na návrhy šifer určené pro FPGA.



Obrázek 1. Konfigurovatelný Look–Up Table (CFGLUT5), převzato z [5].

II. Možné metody pro zvýšení odolnosti vůči DPA

DPA (Differential power analysis – Rozdílová odběrová analýza) je jedna z metod útoků postranními kanály používaných pro lámání číslicových obvodů. Spočívá v měření spotřeby celého obvodu v čase a následné analýze závislosti spotřeby na datech vstupujících do algoritmu. Implementace, kterou používáme jako výchozí, využívá dynamickou rekonfigurace pro funkční rozdělení S–boxů. Konkrétněji je S–box funkčně rozdělen do dvou po sobě navazujících entit, přičemž ty jsou společně funkčně ekvivalentnímu standardnímu S–boxu pro PRESENT [5]. Klopné obvody pro ukládáni dat mezi jednotlivými rundami šifrovacího algoritmu jsou umístěny mezi tyto dvě entity, a tak jsou data ukládaná do klopných obvodů jiná, než data na výstupu S–boxu, resp. druhé části S–boxu ve variantě s dvěma entitami. Po každé zašifrování jsou pak náhodně vybrány dva signály v první části S–boxu (R1), které jsou překříženy, a následně je přepočítána druhá entita (R2) takovým způsobem, aby opět obě entity společně byly funkčně ekvivalentní standardnímu S–boxu pro šifru PRESENT. Tímto způsobem je tedy z vnějšího pohledu zachována vstupně/výstupní funkce celého návrhu, přičemž se vnitřní struktura mění a především jsou do klopných obvodů ukládána data přímo nesouvisející s hodnotou na výstupu S–boxu. Tímto je tedy potenciální útok na zařízení ztížen. Jak se píše ve výchozím článku, tato metoda je při kombinaci s dalšími metodami uvedenými v článku dostatečnou ochranou proti útokům rozdílovou odběrovou analýzou prvního řádu. Struktura rozděleného S–boxu je dobře viditelná na Obrázku 2.



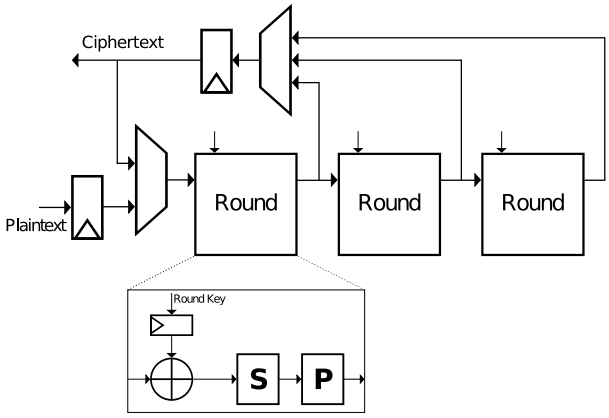
Obrázek 2. Rozdělení S–boxu, převzato z [5].

A. Nově navrhovaná protiopatření

Prvním způsobem, jak použít dynamickou rekonfiguraci trochu jinak, než je tomu v původním článku [5], je jiná frekvence rekonfigurace samotné. V původním článku se rekonfigurace provádí vždy po dokončení jednoho celého šifrování či dešifrování. Například rekonfigurace S–boxu po každé rundě šifrovacího algoritmu by sice pravděpodobně mělo vysokou režii, ale také by mohlo mít pozitivní vliv na odolnost vůči útokům pomocí rozdílové odběrové analýzy.

Rozdělení ostatních modulů v implementaci šifry v klasické rundovní architektuře jako opatření proti rozdílové odběrové analýze je zbytečné a nebude předmětem dalšího zkoumání. Důvodem je jednoduchá úvaha, že v modulu, který pouze počítá výsledek funkce XOR pro aktuální data podklíč pro danou rundu, není funkčně co dělit a rozdělit permutační vrstvu na dvě různé permutace nijak nezmění Hammingovu váhu dotčených dat.

Zajímavou metodou se zdá být použití dynamické rekonfigurace pro náhodný výběr skutečně použitých a vypočítaných, avšak nepoužitých, rund šifrovacího algoritmu, tedy jeden ze způsobů ukrývání výpočtu v čase [6]. Toho je možné dosáhnout implementací více než jedné rundy, které na sebe budou vzájemně navazovat, a použitím výpočtu náhodného počtu z nich v každém hodinovém cyklu obvodu. Například pokud vezmeme v potaz šifru PRESENT s jejími 31 rundami a jedním rundovním podklíčem navíc (nadále zjednodušeně uvažuji o 32 rundách), mohli bychom implementovat 3 na sebe navazující rundy. S výše uvedenou hardwarovou architekturou potřebujeme 16 hodinových cyklů pro celé šifrování za předpokladu, že průměrně jsou vypočítány právě 2 rundy algoritmu. Můžeme zahájit výpočet s náhodně vybraným počtem použitých rund od jedné do tří. Nepoužité rundy taktéž způsobí přepnutí logických hodnot na vnitřních signálech a tedy zvýšený odběr elektrické energie obvodem, ale nijak se data na jejich výstupech nepromítnou do hodnot uložených v klopných obvodech. Abychom dosáhli vždy stejného času celého výpočtu, můžeme počítat použité rundovní výpočty a s blížícím se koncem algoritmu pak použít vždy buď jednu nebo tři rundy z výpočtu tak, abychom ve výsledku dosáhli výpočtu jednoho šifrování či dešifrování trvajícího právě 16 hodinových taktů. Schéma návrhu s touto modifikací je na Obrázku 3.



Obrázek 3. Struktura návrhu s nevyužitými rundami.

Při použití modifikace s nepoužitými spočítanými rundami by bylo jistější nejprve předem spočítat všech 32 potřebných rundovních podklíčů hned na začátku šifrování/dešifrování. Výpočet nově potřebných rundovních podklíčů by totiž mohl případnému útočníkovi poskytnout tolik potřebný únik informací.

Zajímavou možností k prozkoumání je pak kombinace jednotlivých přístupů, tedy například implementace více rund s náhodně určeným množstvím skutečně použitého výpočtu

a rozdělení S–boxu. Tato metoda by se mírně zkomplikovala u komplikovanějších šifer, kde se různí rozložení S–boxů pro jednotlivé rundy, jelikož bychom museli řešit situaci, kdy musíme správně přiřazovat jednotlivým rundám S–boxy ačkoli deterministicky neznáme postup výpočtu.

B. Generátor náhodných čísel

Pro všechny uvedené metody použití dynamické rekonfigurace je nezbytné, aby byl použit generátor náhodných čísel. Toto je nutné vzhledem k používání náhodných čísel pro řízení jednotlivých prvků rekonfigurace v použitých protiopatřeních, jako je například výběr signálů pro překřížení jejich hodnot při rozdělení S–boxu. V našem výzkumu se generátory náhodných čísel nebudeme zabývat a použijeme některý z běžných způsobů implementace generátorů pseudonáhodných, například lineární zpětnovazebný registr. Na použitém řešení generátoru (pseudo)náhodných čísel pak závisí také další vlastnosti obvodu, jako například zda dva stejné a stejně naprogramované obvody generují identické posloupnosti čísel. Z tohoto důvodu je důležité během měření použít více různých generátorů, aby nedošlo ke zkreslení výsledků.

III. Zachování spolehlivosti

Velmi důležitým aspektem všech metod pro zvýšení odolnosti vůči útokům pomocí rozdílové odběrové analýzy je jejich vliv na celkovou spolehlivost návrhu/obvodu. Pro potlačení těchto negativních vlivů můžeme vylepšit spolehlivost obvodu některou ze známých metod jako je například TMR (Triple Module Redundancy). Stále je však velmi důležité těmito opatřeními pro zvýšení spolehlivosti opět nezhoršit odolnost vůči útokům, tedy nezpůsobit žádný únik tajné informace. K dosažení výše uvedeného je třeba důkladně prozkoumat chování obvodu. Lze předpokládat, že je možné dosáhnout lepších výsledků pomocí editace výsledného bitového proudu (bitstreamu) pro FPGA na nízké úrovni. Úpravou namapování návrhu na FPGA čip v XDL formátu [3] s použitím softwarového nástroje TORC [10] můžeme potenciálně dosáhnout více synchronního nebo spolehlivějšího návrhu. Pomocí nástroje TORC je možné číst vygenerovaný bitstream, prozkoumat fyzické namapování návrhu, provést potřebné změny a následně vytvořit bitstream nový se zohledněním provedených změn. Struktura programu TORC je ukázána na Obrázku 4.

IV. Závěr

Cílem výzkumu je nalézt způsob, jak navrhovat číslicové obvody na FPGA spolehlivě a bezpečně zároveň. Výzkum je prakticky zaměřen na algoritmy výpočetně nenáročných šifer, přičemž poznatky získané při zkoumání těchto algoritmů by mohly být alespoň z části použitelné všeobecně pro návrh číslicových obvodů. Začneme tedy nejprve zopakováním implementace šifry PRESENT s použitím dynamické rekonfigurace podle již dříve publikovaného příspěvku. Poté prozkoumáme vliv námi navrhovaných metod pro zvýšení odolnosti proti útokům pomocí rozdílové odběrové analýzy založených na dynamické rekonfiguraci. Nakonec taktéž prozkoumáme tyto metody z pohledu spolehlivosti návrhu a možnosti editace fyzického mapování na FPGA čip za účelem dosažení přijatelné spolehlivosti i bezpečnosti.

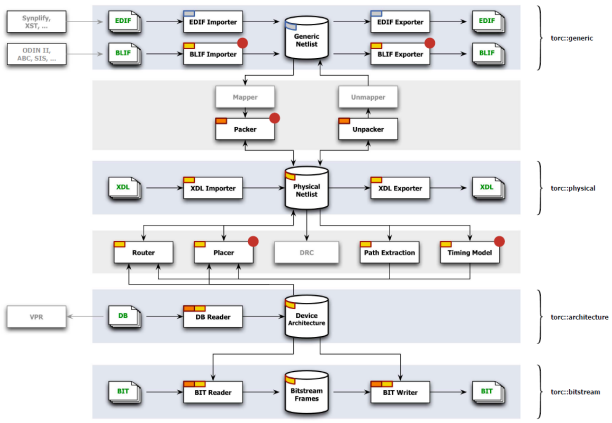


Figure 1: Torc block diagram. *Red dots indicate components still under development.*

Obrázek 4. Struktura programu TORC se zobrazením podporovaných formátů a vybraných komerčních programů pracujících s těmito formáty. Převzato z [10].

Poděkování

Tento výzkum byl částečně financován z grantu GA16-05179S České grantové agentury "Fault Tolerant and Attack-Resistant Architectures Based on Programmable Devices: Research of Interplay and Common Features" (2016-2018).

Tato práce byla podpořena z grantu ČVUT SGS17/213/OHK3/3T/18.

Reference

[1] D. K. Pradhan, *Fault-tolerant computer system design*, 1995.

[2] V. Klíma and T. Rosa, "Attack on private signature keys of the OpenPGP format, PGP (TM) programs and other applications compatible with OpenPGP," *IACR ePrint archive 2002/76*, 2002.

[3] C. Beckhoff, D. Koch, and J. Torresen, "The xilinx design language (XDL): Tutorial and use cases," in *Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC), 2011 6th International Workshop on*, 2011, pp. 1–8.

[4] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: an ultra-lightweight block cipher," in *Cryptographic Hardware and Embedded Systems - CHES 2007*, 2007, pp. 450–466.

[5] P. Sasdrich, A. Moradi, O. Mischke, and T. Güneysu, "Achieving side-channel protection with dynamic logic reconfiguration on modern fpgas," *Journal of Cryptographic Engineering*, no. 2, pp. 107–121, June 2014.

[6] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks*, 2007.

[7] M. Bartík and J. Buček, "A low-cost multi-purpose experimental fpga board for cryptography applications," in *2016 IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, 2016, pp. 1–4.

[8] *7 Series FPGAs Data Sheet: Overview*, [online], Xilinx, [rev. 28. 3. 2017], [cit. 23. 6. 2017].

[9] *Xilinx 7 Series FPGA and Zynq-7000 All Programmable SoC Libraries Guide for HDL Designs*, [online], Xilinx, [rev. 2. 10. 2013], [cit. 23. 6. 2017].

[10] N. Steiner *et al.*, "Torc: towards an open-source tool flow," in *Proceedings of the 19th ACM/SIGDA international symposium on Field programmable gate arrays*, 2011, pp. 41–44.

Digitálne metódy kalibrácie analógových integrovaných obvodov

Michal Šovčík
1. ročník, denná prezenčná forma štúdia
Školiteľ: Viera Stopjaková

Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave
Ilkovičova 3, 812 19 Bratislava, SR
michal.sovcik@stuba.sk

***Abstrakt*—Tento príspevok sa zaoberá vplyvom rastúcej miery integrácie elektronických obvodov (tzn. zmenšovanie minimálneho rozmeru technológie) na parametre obvodov. Náš záujem je sústredený na metódy kalibrácie integrovaných obvodov (IO), ktorými je možné nežiaduce vplyvy integrácie kompenzovať. Práca opisuje všeobecný princíp kalibrácie na príklade kompenzácie vstupného napäťového ofsetu operačného zosilňovača (OZ), realizovaného v nanometrovej technológii využitím nízko-príkonového návrhu. Taktiež sú analyzované náležitosti návrhu samotného kalibračného obvodu. V závere príspevku sú formulované očakávané prínosy kalibračnej metodiky v rámci použitia v OZ a taktiež sú tu stanovené ciele dizertačnej práce.**

***Kľúčové slová*—kalibračný systém, fluktuácia technológie, vstupný ofset, nízko-príkonové obvody, bulk-driven**

I. Úvod

Rýchly rozvoj technológie výroby polovodičových štruktúr umožňuje realizáciu IO s rapídne zmenšujúcou sa veľkosťou. Minimálny rozmer obvodových elementov v súčasnosti dosahuje 7 nm [1]. Medzi hlavné prínosy integrácie patrí hlavne menšia plocha čipov a nižšia spotreba energie. Na druhej strane zmenšovanie rozmerov tranzistorov a vodivých prepojov je sprevádzané významnou náhodnou fluktuáciou parametrov výrobného procesu (napr. koncentrácia dopácie polovodiča či hrúbka izolačných vrstiev). Tieto významné odchýlky sa prejavujú už v rámci substrátu jedného čipu.

II. Motivácia

Jedným z najdôležitejších parametrov IO, ktoré trend fluktuácie technológie negatívne ovplyvňuje je prahové napätie tranzistora (V_{TH}). Štandardná odchýlka V_{TH} tranzistorov vyrobených v 45 nm CMOS technológii dosahuje 16% typickej hodnoty v tejto technológii [2]. Keďže vlastnosti precíznych analógových IO závisia od miery prispôsobenia (z angl. *matching*) obvodových elementov a diferenciálnych signálových ciest, nezhodnosť obvodových elementov znižuje celkovú výťažnosť a spoľahlivosť IO.

Medzi elektrické parametre, ktoré sú degradované rozptylom parametrov výrobného procesu patria tiež parazitné vlastnosti vodivých prepojov, t.j. parazitný odpor a parazitná kapacita [1]. Náhodná zmena týchto parametrov môže následne degradovať hlavne frekvenčné vlastnosti elektronického obvodu. Stabilita parametrov IO sa tiež znižuje po výrobe vplyvom starnutia materiálov a štruktúr. Toto má za následok konkrétny jav tzv. tepelnú nestabilitu hodnoty prahového napätia (NBTI z angl. *negative-bias temperature instability*).

Zvýšený rozptyl hodnoty prahového napätia V_{TH} MOS tranzistorov v rôznej miere degraduje väčšinu elektrických parametrov IO, či už jednosmerných alebo striedavých. Osobitná pozornosť je však venovaná vstupnému napäťovému ofsetu operačného zosilňovača V_{in_OFF}. Od tohto totiž závisia mnohé ďalšie vlastnosti obvodov a komplexnejších systémov využívajúcich OZ.

Ďalší trend v oblasti návrhu IO vyplýva z požiadaviek mobilných aplikácií, ktoré sa premietajú do znižovania hodnoty napájacieho napätia obvodov. Nižšie napájacie napätie prináša na jednej strane výhodu nižšej spotreby energie. Na druhej strane sa v obmedzených energetických podmienkach zhoršujú niektoré vlastnosti IO. Medzi tieto vlastnosti patria najmä dynamický rozsah, potlačenie rušivého signálu z napájania (PSRR z angl. Power Supply Rejection Ratio) a odstup signál-šum [3]. Aby bolo možné udržať tieto parametre na úrovni, ktorú vyžaduje implementácia IO v mnohých aplikáciach, pri návrhu obvodov je nevyhnutné využívať špecifické topológie a techniky návrhu. Jednou z veľmi sľubných metód návrhu sa javí technika riadenia tranzistorov substrátovou elektródou. Ako signálový vstup sa v tomto prípade používa práve substrátová elektróda tranzistora. Týmto spôsobom je odstránená potreba prekonávať V_{TH} v signálovej ceste [4].

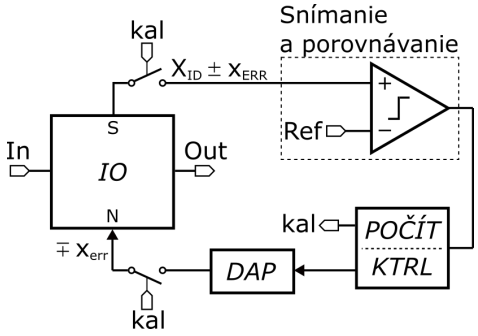
Z uvedených nežiaducich vplyvov v nanometrových technológiách vyplýva, že pre správnu funkciu obvodov je nevyhnutná kompenzácia parazitných a degradovaných parametrov prostredníctvom kalibrácie. Tento príspevok pojednáva o vlastnostiach zvolenej kalibračnej techniky v spojení s očakávaným prínosom. Z hľadiska konkrétnej realizácie, kapitola IV uvedie náhľad do problematiky kompenzácie ofsetu plne diferenčného operačného zosilňovača. V kapitole V sú sformulované rámcové ciele dizertačnej práce.

III. Princíp kalibrácie

Kalibračný podobvod pozostáva zo systému prídavných obvodových blokov a elementov, ktoré sú pripájané ku kalibrovanému IO pomocou spínačov s určitou frekvenciou podľa

potreby a typu kalibrácie. Táto teda závisí od faktorov, ktoré ovplyvňujú rozptyl parametrov systému. Pokiaľ sa vlastnosti obvodov menia s ich vekom, postačovať bude jednorazová kalibrácia pri spustení systému. Ak sa niektoré parametre menia aj v závislosti od teploty, kalibrácia týchto môže byť vykonávaná pravidelne s nízkou frekvenciou (rádovo v jednotkách Hz). V prípade prudkých variácii vlastností, ktoré môžu byť spôsobené napríklad blikavým šumom, kompenzácia musí prebiehať s vyššou frekvenciou [5].

Obr. 1 znázorňuje blokovú schému jednoduchého systému IO s kalibračný podobvodom, ktorý je možné rozčleniť podľa funkcie jednotlivých blokov.



Obrázok 1. Bloková schéma IO s kalibračným obvodom.

V prvom rade je potrebné zvoliť vhodný elektrický parameter daného IO, ktorý reflektuje nežiaduci vplyv variácie výrobného procesu na funkciu alebo vlastnosti obvodu. Takto zvolený parameter je následne snímaný a porovnávaný pomocou komparátora s referenčnou hodnotou. Snímaný signál vo všeobecnosti opisuje nasledovný vzťah:

$$X_{SNIM} = X_{ID} \pm x_{ERR}, \tag{1}$$

kde X_{ID} je ideálna hodnota zvolenej veličiny v súlade s návrhom a x_{ERR} je aktuálna odchýlka X_{SNIM} od ideálnej hodnoty spôsobená rozptylom výrobného procesu. Podľa výstupnej hodnoty komparátora potom kontrolný blok spolu s počítadlom (POČÍT/KTRL na obr. 1) generuje príslušný kód pre D/A prevodník (DAP na obr. 1). Prevodník následne pripája na nulovací port kalibrovaného IO opačnú hodnotu aktuálnej odchýlky kompenzovanej veličiny.

Uzly IO, ktoré sú na obr. 1 reprezentované portami S a N je nutné zvoliť jednak podľa povahy kalibrovaného parametra a tiež podľa konkrétnej topológie IO. Všeobecnými požiadavkami pri voľbe týchto uzlov sa bude hlbšie zaoberať kapitola IV. Hlavnou úlohou kontrolného bloku je riadenie intervalov použitia kalibrácie IO. Pri jeho návrhu je preto potrebné zvoliť frekvenciu kalibrácie IO podľa vyššie opísaných kritérií a tiež je potrebné zvoliť dĺžku trvania jednotlivých kalibračných cyklov.

Plne kalibrovanú hodnotu zvoleného parametra je možné vyjadriť nasledovne:

$$X_{KALIB} = X_{ID} \pm x_{MIN}, \tag{2}$$

kde x_{MIN} je minimálna možná odchýlka, ktorú je možné dosiahnuť so stanovenou presnosťou. Minimálna odchýlka parametra závisí od nepresnosti porovnávacieho procesu a hlavne od nedokonalosti D/A prevodníka. Trvanie kalibračného cyklu závisí od rozdielu aktuálnej odchýlky x_{ERR} od požadovanej konečnej odchýlky x_{MIN}. Počítadlo počas jedného cyklu generuje postupne rastúce digitálne kódy. Podľa nich D/A prevodník privádza na nulovací vstup IO zodpovedajúcu kompenzačnú hodnotu parametra $\mp x_{ERR}$. Po dosiahnutí rovnosti $x_{ERR} = x_{MIN}$ kontrolný blok ukončí kalibračný cyklus a odpojí kalibračný obvod od IO.

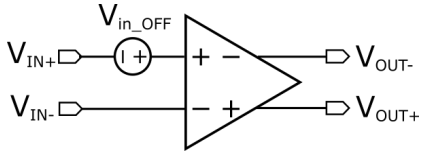
Z hľadiska presnosti kalibrácie je taktiež dôležité uvažovať vplyv anomálií výrobného procesu na bloky samotného kalibračného podobvodu. Značnú pozornosť je potrebné venovať aj vlastnostiam D/A prevodníka. Optimálnu funkciu tohto bloku je možné zabezpečiť použitím topológie tzv. sub-binárneho $M/2^{+}M$ prevodníka [6]. Tento je pritom súčasne kompenzovaný algoritmom pre konverziu základu prevodníka [5].

IV. Kalibrácia diferenčného operačného zosilňovača

Predmetom výskumu v rámci dizertačnej práce z hľadiska konkrétneho použitia zvolenej kalibračnej techniky bude v najbližšej dobe kompenzácia vstupného ofsetu V_{in_OFF} plne diferenčného operačného zosilňovača, ktorý bol realizovaný v $130\ nm$ CMOS technológii.

Kalibráciu je vo všeobecnosti možné zamerať na rôzne parametre. Ich výber závisí od druhu IO, keďže rôzne obvody sa líšia v kritických parametroch, ktoré markantne ovplyvňujú ich činnosť. V prípade diferenčného OZ to môže byť taktiež paramter PSRR, či schopnosť zosilňovača potlačiť zosilnenie súhlasnej zložky (parameter CMRR).

Z pohľadu definície, ofset V_{in_OFF} predstavuje rozdiel napätí na vstupných termináloch operačného zosilňovača, pri ktorom je výstupný napäťový rozdiel $0\ V$ (medzi výstupnými terminálmi navzájom v prípade zosilňovača so symetrickým výstupom alebo voči zemi v prípade nesymetrického výstupu). To znamená, že ofset V_{in_OFF} môže byť modelovaný napäťovým zdrojom, ktorý je pripojený k jednému terminálu OZ, ako znázorňuje obr. 2.



Obrázok 2. Modelovanie vstupného napäťového ofsetu diferenčného OZ.

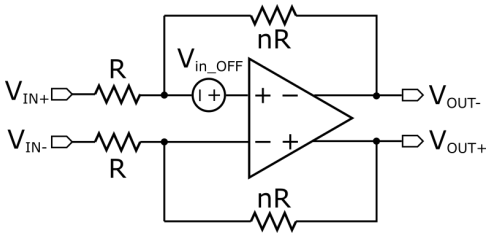
Pre konkrétnu realizáciu vybranej kalibračnej techniky je potrebné v obvode určiť dva uzly: detekčný uzol a kompenzačný uzol [5]. Kritéria pre túto voľbu sú nasledovné: a) snímaný signál je funkciou kalibrovaných nedostatkov a nezávisí od ostatných parametrov IO, b) úroveň snímaného signálu je omnoho vyššia ako je úroveň šumu a rušivých signálov, spôsobených nedostatkami samotného kalibračného podobvodu [5].

Zvolené uzly jednotlivo prislúchajú čiastkovým fázam kalibrácie. Následne je potrebné navrhnúť jednotlivé konfigurácie pre detekciu a kompenzáciu degradovaného parametra.

A. Detekčná konfigurácia

Detekčná konfigurácia celého systému je taká, v ktorej je možné pozorovať degradovaný parameter v detekčnom uzle. V závislosti od povahy funkcie kalibrovaného IO je možné detekciu realizovať paralelne s normálnou činnosťou IO tak, aby nedochádzalo k prerušovaniu. Pokiaľ to však činnosť IO dovoľuje, je možné do topológie obvodu zasahovať so zvolenou frekvenciou a sériovo meniť činnosť IO medzi pracovným režimom a jeho kalibráciou.

Na obr. 3 je znázornená detekcia ofsetu V_{in_OFF} plne diferenčného OZ v konfigurácii s uzavretou slučkou spätnej väzby. V tomto type detekčnej konfigurácie nie je potrebné vnútorné zapojenie OZ modifikovať, a je preto vhodná pre systémy, kde je potrebná ich spojitá funkcia v čase.



Obrázok 3. Detekčná konfigurácia pracovného OZ.

Napäťový zdroj pripojený ku kladnému vstupu OZ v zobrazenom zapojení modeluje vstupný napäťový ofset. Z nasledujúceho vzťahu je možné získať hodnotu napätia V_{in_OFF}:

$$V_{in\_OFF} = \frac{V_{OUT} - A_{CL}.V_{IN}}{A_{CL}}, \tag{3}$$

kde V_{IN} a V_{OUT} je diferenciálne vstupné respektíve výstupné napätie OZ a A_{CL} je zisk uzavretej slučky zosilňovača.

B. Kompenzačná konfigurácia

Kompenzáciu/korekciu vychýleného parametra je možné uskutočniť injekciou vhodného kompenzačného prúdu prostredníctvom kompenzačného uzlu IO. Kompenzačný uzol musí byť preto zvolený tak, aby korelácia medzi injektovaným prúdom a korigovaným parametrom bola čo najväčšia. Na druhej strane je rovnako dôležité zachovať minimálnu koreláciu medzi kompenzačným prúdom a ostatnými parametrami IO [5]. Uvedený spôsob kompenzácie je vhodný pre obvody založené na činnosti MOS tranzistorov, keďže tieto transformujú napätie na svojom vstupe na prúd tečúci obvodom.

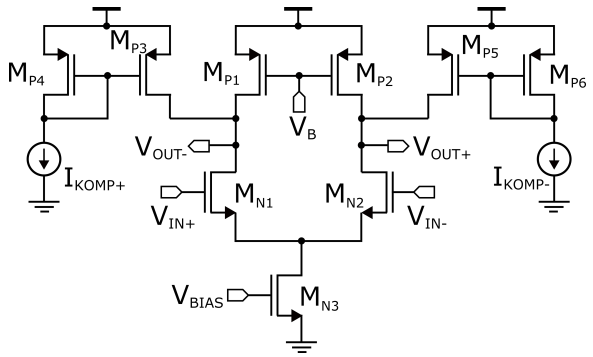
Keďže signál spracovávaný OZ je primárne vedený prúdom a následne konvertovaný na napätie prostredníctvom výstupnej impedancie, najefektívnejšie je korigovať ofset na výstupe [5]. Obr. 4 znázorňuje optimálnu konfiguráciu kompenzácie na všeobecnom príklade diferenčného OZ.

Keďže V_{in_OFF} priamo ovplyvňuje prúd jednotlivými vetvami OZ, úpravou týchto prúdov je možné eliminovať vplyv ofsetu na vlastnosti OZ. K obom výstupom OZ sú jednotlivo pripojené prúdové zrkadlá, pozostávajúce z PMOS tranzistorov. Tieto vhodne distribuujú kompenzačný prúd I_{KOMP+} alebo I_{KOMP-} do príslušnej vetvy OZ. Veľkosť a voľbu potrebného kompenzačného prúdu riadi kontrolný blok (obr. 1). Ideálne prúdové zdroje v schéme na obr. 4 predstavujú výstupy D/A prevodníka (v tomto prípade diferenciálneho).

Z hľadiska návrhu kompenzačného podobvodu je dôležité dodržať niekoľko kritérií, aby spätný vplyv na kalibrovaný IO bol minimálny. Rozsah možných hodnôt dĺžky kanála tranzistorov M_{P3} a M_{P5} je na ohraničený vodivosťou kanála (g_{DS}). Táto musí byť dostatočne nízka, aby neovplyvnila impedanciu kompenzačného uzla. Na druhej strane, dĺžka kanála uvedených tranzistorov musí byť dostatočná na to, aby sa zamedzilo vplyvu výkyvu napätia kompenzačného uzla na injektovaný kompenzačný prúd. Tento efekt je spôsobený moduláciou dĺžky kanála MOS tranzistora v saturačnom režime. Táto je charakterizovaná koeficientom modulácie dĺžky kanála nasledovne [7]:

$$\lambda = \frac{\Delta L}{L.V_{DS}}, \tag{4}$$

kde L je celková dĺžka kanála MOS tranzistora, ΔL je úbytok efektívnej dĺžky kanála tranzistora vplyvom modulácie a V_{DS} je napätie medzi kolektorom a emitorom.

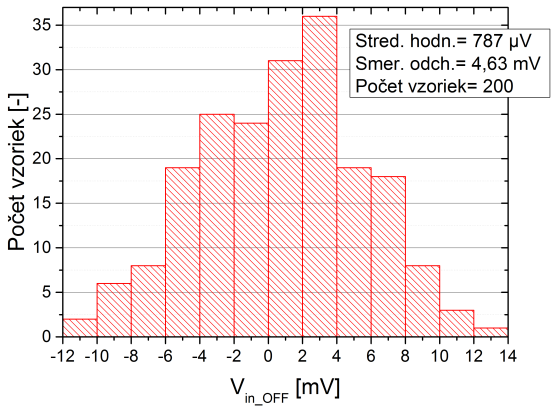


Obrázok 4. Kompenzačná konfigurácia plne diferenčného OZ.

C. Ciele kalibrácie

Obr. 5 znázorňuje simulačný výsledok Monte Carlo analýzy vstupného napäťového ofsetu plne diferenčného zosilňovača s variabilným ziskom (VGA). Tento je napájaný napätím $0,6\ V$ a je realizovaný v 130 nm CMOS technológii. Pri použití 200 vzoriek analýzy, štandardná odchýlka dosahuje hodnotu $4,63\ mV$ a stredná hodnota ofsetu je $787\ \mu V$.

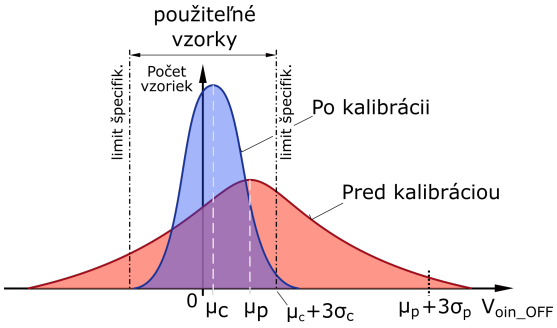
Obr. 6 principiálne znázorňuje štatistické rozloženie hodnôt V_{in_OFF} pre zvolené množstvo vzoriek IO bez použitia kalibrácie (ako napr. výsledky na obr. 5) a cieľové rozloženie štatistiky s použitím kalibrácie. Cieľom použitia kalibračnej techniky bude pokiaľ možno znížiť strednú hodnotu ofsetu V_{in_OFF}, no dôležitejšie bude znížiť hodnotu štandardnej odchýlky na σ_c ako je znázornené v obr. 6. Vzorky IO, ktoré prekračujú limity špecifikácie musia byť vyradené alebo opätovne testované. Je nutné poznamenať, že v dôsledku



Obrázok 5. Výsledok MC analýzy vstupného ofsetu napätia OZ.

rastúcej integrácie ako aj väčšieho rozptylu parametrov IO v nanotechnológiách, náklady na testovanie dosahujú pri niektorých integrovaných systémoch až približne 40-50% celkových výrobných nákladov [2]. V našom prípade teda znížením štandardnej odchýlky V_{in_OFF} pomocou kalibrácie, vzrastie počet vzoriek vyhovujúcich špecifikácii (tzv. výťažnosť).

Hranica intervalu vzoriek $\mu + 3\sigma$ v rámci simulovaných výsledkov na obr. 5 dosahuje $V_{in\_OFF} = 14,68\ mV$. Táto hodnota rozdielového vstupného napätia je potom zosilnená celkovým ziskom zosilňovača. Uvedený zosilňovač dosahuje zisk približne $33\ dB$. Výstupné ofsetové napätie dosiahne potom zhruba $650\ mV$, čo je viac ako hodnota napájacieho napätia. Výstupný dynamický rozsah bude z tohto dôvodu výrazne redukovaný [8].



Obrázok 6. Očakávaný rozptyl hodnôt vstupného ofsetu pred a po kalibrácii.

V. Zámer a rámcové ciele dizertačnej práce

Hlavným zámerom dizertačnej práce je analýza a rozvoj kalibračných techník z pohľadu účinnosti redukcie vplyvu fluktuácie technológie, teploty a starnutia na činnosť integrovaných obvodov v nanotechnológiách. Taktiež bude dôležité aj hľadisko možnej interakcie prídavných blokov s kalibrovaným IO. Cieľom bude nájsť techniku s najlepším kompromisom maximálnej účinnosti a minimálneho nežiaduceho vplyvu na samotný IO. Pre tento cieľ bude teda dôležité navrhnúť a optimalizovať funkcie jednotlivých blokov kalibračného obvodu a implementovať ho do celkového obvodu OZ (podľa obr. 1).

Ďalším cieľom dizertačnej práce je preskúmať spojenie opísanej kalibračnej techniky s ďalšou, ktorá bude vhodná pre kompenzáciu šumu v IO. Týmto spôsobom bude možné ťažiť z výhod oboch metód pri obmedzenom vplyve nedostatkov jednej z nich samostatne.

Výskum bude tiež zameraný na návrh a analýzu vybraných techník s použitím MOS tranzistorov riadených substrátovou elektródou. Očakávame, že tento prístup dovolí použiť kalibračnú metódu pri nižšom napájacom napätí.

Ďalším krokom v našom výskume bude testovanie a verifikácia zrealizovaných IO vzhľadom na návrh. Podľa získaných výsledkov bude neskôr možné presnejšie prispôsobiť parametre kalibračnej metódy.

VI. Záver

V tomto príspevku bol analyzovaný všeobecný princíp kalibrácie analógových IO za účelom zredukovať nežiaduci vplyv starnutia a fluktuácie parametrov technologického procesu a teploty na vlastnosti IO vyrobených v nanotechnológiách, a napájaných ultra-nízkym napätím. Podľa uvedených štatistických výsledkov Monte Carlo analýzy vstupného ofsetu OZ realizovaného v $130\ nm$ CMOS technológii, je výstupný dynamický rozsah vplyvom nestability výrobných a pracovných podmienok výrazne redukovaný.

Preto je potrebné napäťový ofset na vstupe OZ kompenzovať pomocou vhodne zvolenej kalibračnej techniky. V tejto práci boli uvedené princípy návrhu vybraných blokov kalibračného systému, ktoré budú neskôr použité pri celkovej implementácii kalibračného obvodu pre OZ.

V rámci mojej doterajšej práce a výskumu vznikli 4 publikácie, ktorých som prvoautorom alebo spoluautorom (3 príspevky na medzinárodnom sympóziu DDECS a 1 príspevok na medzinárodnej konferencii MIPRO).

Poďakovanie

Táto práca bola podporená projektami APVV-15-0254, VEGA 1/0762/16 a VEGA 1/0905/17.

Literatúra

[1] J. Chang, Y. Chen, and W. C. et. al., "A 7nm 256mb sram in high-k metal-gate finfet technology with write-assist circuitry for low-vmin applications," in *IEEE International Solid-State Circuits Conference*, Feb 2017, pp. 206–208.

[2] M. Onabajo and J. Silva-Martinez, *Analog circuit design for process variation-resilient systems-on-a-chip*. Springer Science & Business Media, 2012.

[3] D. Arbet, M. Kováč, L. Nagy, V. Stopjaková, and J. Brenkuš, "Low-voltage bulk-driven variable gain amplifier in 130 nm cmos technology," in *Design and Diagnostics of Electronic Circuits & Systems (DDECS), 2016 IEEE 19th International Symposium on*. IEEE, 2016, pp. 1–6.

[4] M. Rakús, "Analýza prúdových zrkadiel riadených substrátovou elektródou," in *Počítačové architektúry a diagnostika*, 2016, pp. 25–28.

[5] M. Pastre and M. Kayal, *Methodology for the digital calibration of analog circuits and systems*. Springer, 2006.

[6] D. Arbet, G. Nagy, V. Stopjakova, and G. Gyepes, "A self-calibrated binary weighted dac in 90nm cmos technology," in *Microelectronics Proceedings-MIEL 2014, 2014 29th International Conference on*. IEEE, 2014, pp. 383–386.

[7] B. a. Razavi, *Design of analog CMOS integrated circuits*. McGraw-Hill, 2001.

[8] R. Palmer, "Dc parameters: Input offset voltage," Texas Instruments, Tech. Rep. SLOA059, March 2001.

OFDM and FHSS Hybrid Network

Tomáš Jakubík
First year, full-time study
Jiří Jeníček

Faculty of Mechatronics, Informatics and Interdisciplinary Studies
Technical University of Liberec
Studentská 1402/2, 461 17 Liberec 1, Czech Republic
tomas.jakubik@tul.cz

***Abstract*—This paper describes thoughts and starting points on future studies in the area of wireless networks for home automation and security systems or sensor networks. The main concept is to combine Frequency Hopping Spread Spectrum (FHSS) and Orthogonal Frequency Division Multiplex (OFDM) types of communication into one network. The former type is used for low-power devices with small data-rate but fast response time. The latter type is used for high data-rate devices. The SDR needed for OFDM communication is the key in combining both technologies. Three different ideas are examined. First is a low-cost solution with DVB-T tuner and sub-GHz GFSK transceivers. Second is multiple FHSS hopping schemes fitting together to form an OFDM. Third is an OFDM link with holes for the FHSS communication.**

***Keywords*—FHSS, OFDM, Low-power, Sensor Network, Home Automation and Security**

I. Introduction

Wireless communication technologies are, as most engineering tasks, mostly developing in areas with the largest concentration of customers. Any complicated and expensive idea can be put into a mobile phone and made cheap by mass production. This unfortunately left certain segments of small electronics free of new wireless technologies. Sometimes it is easy to put the mass-produced technology in a specific situation, but there are cases where the only way is to start from the ground up. One of these cases is home automation and security and it is discussed here.

A. Home Automation and Security

Let's imagine what devices would need to communicate in a case of home area network for automation and security.

1) Fast Digital Input: It is any device which waits for a certain condition and transmits a simple digital information. This device may be a doorbell, TV remote, light-switch, but it can be a smoke detector or motion detector as well. These devices need to survive many years on a small battery. For a door or window open sensor it may be years on a coin 220 mA h CR2032 battery.

2) High Data-rate Devices: These are for example security cameras. Seldom they need high data-rates, but most of the time nobody is watching. Cameras which continuously send the video off-site need a cable connection anyway, so let's leave them off this list. Similar to camera might be a user interface device. The only difference is that the video feed ends here. These devices need large battery or be powered from the grid and have the battery only as a backup.

3) Automation Device: The last class might be a switchable AC socket. It must be powered from the grid, but it doesn't need much data except for firmware updates.

B. Spread Spectrum

The easiest way to connect a cheap fast digital input device is to use a single frequency system built on sub-GHz digital transceivers. The input device sleeps, waiting for an activation. When needed, it wakes up and starts transmitting. The other end is receiving all the time and immediately knows that something has happened. But using a single frequency communication can be sometimes quite problematic.

There are undesirable effects, which must be taken into consideration, during the wireless communication. As an example, multi-path propagation depends strongly on antennae positions and on used frequency. It can be simply solved by slightly changing the frequency. Multiple-Input Multiple-Output (MIMO) technology can even use that to its advantage and transmit multiple data at the same time. Nowadays, there is a need to use a spread spectrum technology.

There are many ways to spread the spectrum. One of these is FHSS. If the frequency is hopped in between regular packets (similarly to Bluetooth), then the sub-GHz digital transceivers can be used with all the advantages. Disadvantage is that the input device cannot just start transmitting, because the other end will be most likely listening on a different frequency. The problem can be solved by synchronisation, but that increases power consumption. The synchronisation requires the devices to precisely count time and periodically re-synchronise. The key to low consumption is sleeping for a long time, so in this paper I presume that devices don't know anything about current state of the network when they wake up. Another way might be to send periodic beacons. I have already tried that idea [1] and although the power consumption and response time were acceptable, there was too much radio pollution created by the beacons.

C. Existing Solutions

The easiest solution might be to combine several existing technologies. Each of them, however, has its disadvantages and

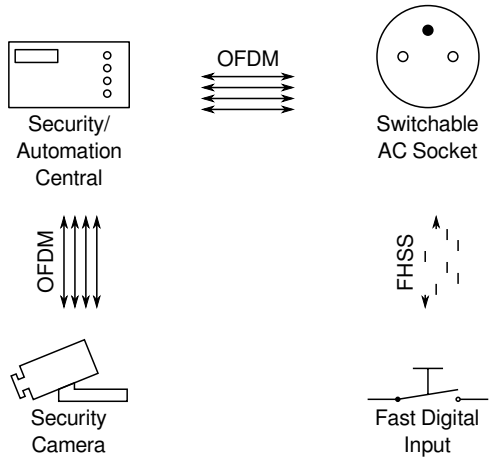


Fig. 1. OFDM and FHSS combination.

often they interfere with each other. Plus there is the price of several systems instead of one.

1) Wi-Fi: It is too powerful for fast digital input and consumes to much power. It is used by vulnerable devices (mobile phones, cheap routers) and usually badly secured. By irony, the least secure devices nowadays are security cameras.

2) Bluetooth LE: Bluetooth LE has barely enough data-rate for camera. Its range in a building is not great. Signal on 868 MHz travels much better. For fast digital input, it is either too slow in cyclic sleep [2] or it consumes too much power when connected firmly. This is perhaps improved in ver. 5 by extended advertising options.

3) ZigBee: It has barely enough data-rate for Camera at 2.4 GHz, not enough at 868 MHz. Only one channel at 868 MHz means no Adaptive Frequency Agility (AFA) and duty cycle limited to 1 %. However, for fast digital input this might be a good solution.

4) IOT networks: Some of them are ultra narrow band, some of them use chirps. Either way, there is not enough data-rate for anything except fast digital input, and for that it is too slow and insecure. Let's imagine that smoke detector detects fire. The signal is perhaps sent to the nearest mobile operator which maybe puts it to the Internet where it may find its way to the right cloud. The way back is even worse before the siren wakes you up to run.

II. The general idea to combine OFDM and FHSS

OFDM is a spread spectrum technique where the transceivers use many narrow band sub-channels at once. The sub-channel transmissions are orthogonal to each other. This means that the neighboring channel is put exactly to the frequency where the sinc spectrum of the first sub-channel is zero. The OFDM hardware is usually a kind of software radio. The receiver can receive high data-rates when all sub-channels are used, but it can listen for a single channel transmission on any of the channels. When the full OFDM link is not used, the OFDM hardware of grid powered devices can listen on all frequencies. Fast digital input devices equipped only with a single frequency hardware may sleep and start transmitting at any time on any random frequency. That results in negligible delay on input event and the lowest possible power consumption while keeping the spread spectrum properties. The OFDM hardware could also re-transmit messages on different channels and save time of routing. This will be possible only if the transmission on one channel doesn't disrupt receiving on different channels.

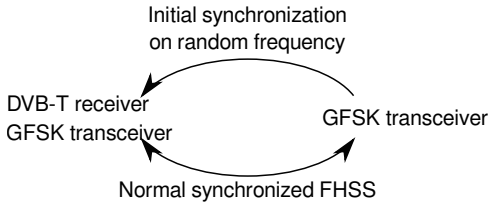


Fig. 2. FHSS with DVB tuner.

III. Simplified version with FHSS and DVB tuner

A chip for software radio costs lots of money unless it is mass produced. One of the less expensive chips is DVB-T tuner, which works approximately at the right frequency range of 868 MHz. The DVB-T tuner cannot transmit and the communication would be handled mostly by regular digital transceivers with Gaussian Frequency Shift Keying (GFSK) modulation. The network hubs or masters would have an extra DVB-T tuner reprogrammed to receive GFSK on every channel used by the network.

The fast digital input would act almost as in single frequency network. It would normally sleep. Upon an event, it would pick a random channel and start transmitting. The network hub would hear that and responded by the GFSK transceiver on the right frequency. In the response there could be enough information for the input device to synchronise with other devices. Simple scheme is on Fig. 2. The time between an event on the input and knowing the information in hub would be as small as possible, in length of one packet. The only delay could be if the input selected bad channel and had to wait for a clear channel or re-tune to another.

An important question is whether it is possible to reprogram the DVB-T tuner to receive GFSK. There are open-source projects GNU Radio and supplementing project RTL-SDR. The aim of the former is to provide a universal platform for Software Defined Radio (SDR). The later offers a very cheap solution for receive-only SDR connected to PC via a USB. Some suggest [3] that FSK demodulation is easy when all the computations are done in PC. In case of an automation central, it would be the DVB-T tuner in combination with small computer, such as Raspberry-Pi, running the GNU Radio.

The FHSS communication in sub-GHz range in European Union is handled by EN 300 220 [4]. The FHSS system must use at least 47 non-overlapping channels at most 100 kHz wide. The SDR needs to calculate at least two frequency bins for one channel to distinguish between logical 0 and logical 1 in GFSK modulation. The smallest FFT size is

$2^7 = 128$ to decode 47 channels. The sampling rate of the RTL-SDR is only 2.4 MSample s^{-1} [3]. That means the total bandwidth can be at most 2.4 MHz and bandwidth of one channel 37.5 kHz. Channel of this size can carry only a very slow communication. The DVB-T tuner itself must be able to process data much faster, but the DVB-T chip's documentation isn't public. Without special access to vendor's documentation, it is unknown whether the DVB-T chip could do custom FFT and send only processed data. The only publicly available option is to get raw In-phase and Quadrature (IQ) samples to the PC, and then the USB is the limiting factor.

In smaller devices, the Raspberry-Pi would be too large. It might be an option to use only the IQ demodulator, sample the signals by a custom AD converter and process in an MCU. The AD converter might be even faster than 2.4 MHz, but now the limit will be in FFT calculation. In the previous example, there are enough samples each 53 µs, but two FFTs would have to be calculated to recognize valid symbols. There are measurements of FFT speed in [5], and the result 26 µs is a realistic value for Cortex-M7. MCU of that size is still an overkill for switchable AC socket.

The last idea might be to use the DVB-T only to detect an incoming transmission. The fast digital input would reconfigure its GFSK transceiver and send few short beeps on different frequencies, without modulation. The hub would recognise position of the beeps by the DVB-T tuner and would have to re-tune its GFSK transceiver to catch a regular packet which follows. This would simplify the FFT, but there would be a possibility of collisions.

The simplest version of the simplest method is to just use the DVB-T tuner to detect preambles. Higher power on one channel would suggest there is a preamble being transmitted on that channel. The hub would tune the GFSK transceiver and start receiving on that channel to receive the rest of the preamble and the packet. Detection of power alone will be very faulty. Any transmission from any alien system or noise in the wide frequency range will trigger the hub to re-tune the GFSK transceiver and loose any ongoing transmission. It is quite probable that this method won't work at all.

IV. OFDM as multiple FHSS schemes fitting together

If we imagine an FHSS communication as a time-frequency graph (called waterfall if the time dimension points up), the transmissions seem to fill up the space equally, but only one frequency is occupied at one time. If we take N orthogonal pseudo-random FHSS sequences on N channels, they fill the time-frequency space completely as on Fig. 3. The communication would be indistinguishable from OFDM on conditions that the channels use phase shift modulation with square pulses and are close together to be orthogonal.

Communication between all devices could start as an FHSS connection with one time-frequency pattern. The network should mostly transmit small amounts of data. When there is a need to transmit more, either to retransmit a routed message

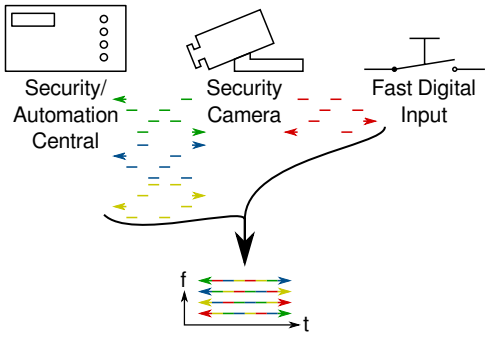


Fig. 3. OFDM as multiple FHSS schemes.

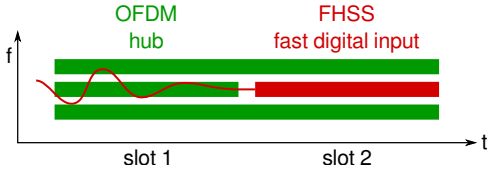


Fig. 4. FHSS device calibrating its frequency.

or to increase the data-rate, the devices could add another time-frequency pattern which fits into the available space.

Transmitting this way from one OFDM transceiver to another would probably be easy. Much more difficult problem is to synchronise multiple transmitters. There already is a promising technology called Orthogonal Frequency Division Multiple Access (OFDMA). It is similar to OFDM, but different sub-channels are used by different transceivers.

The first problem is the synchronisation of frequencies. It may be solved by pilot carriers which are sent by one of the devices [6]. Other devices then modify their frequencies to fit these pilot sub-carriers. This is not usable with single-frequency FHSS transceivers in low data-rate devices. The FHSS device would have to receive a signal from OFDM device in one slot, keep the frequency calibration and start transmitting the next slot. The situation is depicted on Fig. 4. The slot before would need an extra preamble for the FHSS device to successfully synchronise. This also solves the second problem, which is time synchronisation. Question is, whether the single-frequency device will be able to synchronise to one channel if there are two more neighbours filling the spectrum almost completely. All the above is under the assumption that the physical channel is relatively constant, the devices don't move and the frequency imperfection is only caused by the devices. In home area network, the devices are expected to move slowly or not to move at all.

When the air is full of transmission, this would be reasonably fast. In an idle situation, the synchronisation wouldn't be needed, because the orthogonal channels would be empty. The FHSS device doesn't know about the neighbouring orthogonal channels and cannot wait forever on the randomly selected channel, so there is a high possibility of collisions when the system will be at half the capacity. The FHSS sequence and algorithm to determine when to use the synchronisation and

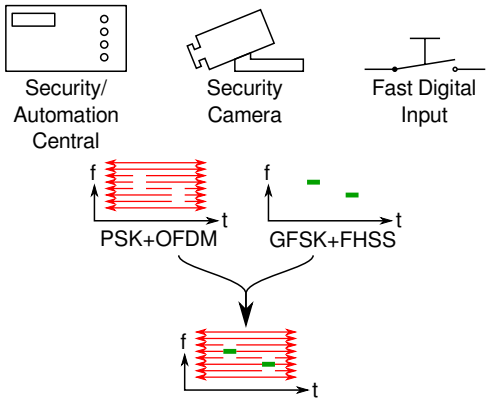


Fig. 5. OFDM with free spaces.

when to put an extra preamble would be quite complicated.

The last problem is with Phase Shift Keying (PSK) single-frequency transceivers. Most sub-GHz transceivers are GFSK, and if they support PSK, it is not the main purpose. GFSK spectrum is not sinc shaped, so the individual channels cannot be put as close together as for OFDM.

V. OFDM with free spaces for FHSS devices

The OFDM link has good properties because there is a lot of slow sub-carriers next to each other. If the physical channel is frequency selective, which almost certainly is, then each sub-carrier is affected independently. On the frequency width of the small sub-carrier, the attenuation is almost constant. That is good, because it doesn't change the pulse shapes and there is no additional Inter Symbol Interference (ISI). A single-frequency transmission on the same bandwidth and in the same physical channel would distort the transmitted symbols beyond recognition.

On the other way, the FHSS link must use comparatively larger sub-channels, because it uses only one at a time. To transmit decent data-rate, the FHSS sub-channels must be wider and there must be less of them to fit in the same bandwidth. For example Bluetooth LE uses 40 sub-channels, while DVB-T2 uses up to 32768 sub-carriers.

This results in an idea to split the OFDM and FHSS communications completely. The OFDM link can use the regular PSK or Quadrature Amplitude Modulation (QAM) with guard intervals and so on. The FHSS link can use the cheap GFSK transceivers. All that remains is to eliminate collisions and use single hardware in hubs for both networks. Collisions between FHSS and OFDM networks are recognized problem for example between Wi-Fi and Bluetooth [7].

The OFDM link needs to have 'holes' to contain the FHSS links. The holes need to be several channels wide to encompass the wide FHSS transmission. Fig. 5 depicts the situation. Normal communication could start as an GFSK modulated FHSS scheme. Each hub could set up its own orthogonal FHSS scheme, so that no collisions appear in one system, but routing may happen immediately. When high data-rate is needed, the software radios would switch to PSK modulation put into OFDM link. The link would have periodically repeated holes for each FHSS sub-channel. When the FHSS fast digital input device wakes up and selects a random sub-channel, the sub-channel is filled by the OFDM signals and the FHSS device must wait. Sooner or later, multiple OFDM sub-carriers silence and the FHSS device may start its transmission.

The hub would have to respond using the GFSK modulation while still transmitting the OFDM signal. That gives the FHSS device enough information where the FHSS holes continue.

This idea will be least difficult for the FHSS devices, but most difficult on the OFDM devices. The OFDM devices would need to calculate two FFTs for a received signal and two more for the outputted signal simultaneously. Plus it needs several synchronisation and other algorithms. For now the obvious choice is to use Zynq on PicoZed SDR platform, but it is really expensive to use in a custom embedded electronics.

VI. Conclusion and Future Goals

For the commercial use, the DVB idea might be the most interesting, because it is the cheapest as it relies on mass produced components. The other two ideas are too expensive so far, but Moore's law might make SDR cheap enough in near future. The second idea might be solved as a part of 4G or 5G OFDMA research, but so far mobile devices are still several orders of magnitude above the required power consumption.

The main goal of the dissertation thesis is to explore methods described above. That will seamlessly link to theoretical design of new wireless communication systems, their simulation and physical implementation. Last step will be to measure interesting parameters of the physical implementation in a real world situation. Hopefully, the results of this dissertation thesis will help engineers in selecting and implementing a wireless technology for home automation, and bring all low-power, quick response and spread spectrum to an area where only compromises have been so far.

Acknowledgement

This work was partially supported by the Student Grant Scheme 2017 of the Technical University of Liberec.

References

[1] T. Jakubík and J. Jeníček, "Asymmetric Low-power FHSS Algorithm," in *Proceedings of the IEEE 13th International Workshop On Electronics, Control, Measurement, Signals and their Application in Mechatronics*, 2017.

[2] S. T. Artem Dementyev, Steve Hodges and J. Smith, "Power Consumption Analysis of Bluetooth Low Energy, ZigBee and ANT Sensor Nodes in a Cyclic Sleep Scenario," in *2013 IEEE International Wireless Symposium*, 2013.

[3] M. A. Wickert and M. R. Lovejoy, "Hands-on Software Defined Radio Experiments with the Low-cost RTL-SDR Dongle," in *2015 IEEE Signal Processing and Signal Processing Education Workshop*, 2015.

[4] *EN 300 220-2*, ETSI Std., Rev. 3.1.1, 2017.

[5] STMicroelectronics, *Digital signal processing for STM32 microcontrollers using CMSIS*, 2016.

[6] J. E. K. Robert J. Baxley and G. T. Zhou, "Pilot Design For IEEE 802.16 OFDM And OFDMA," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2007.

[7] J. So and Y. Kim, "Interference-aware frequency hopping for Bluetooth in crowded Wi-Fi networks," *Electronics Letters*, 2016.

Framework for Planning, Executing and Monitoring Cooperating Computations

Marta Cudova
1st year, full-time study
Supervisor: Jiri Jaros

Centre of Excellence IT4Innovations, Faculty of Information Technology, Brno University of Technology
Bozetechova 1/2, 612 66 Brno, Czech Republic
icudova@fit.vutbr.cz

***Abstract*—Realistic simulations need for their run very powerful computers. Computing infrastructures are growing in parallelism and becoming more diverse. This heads towards using more sophisticated computational techniques to take full advantage of the machine power. To describe a scientific problem, a number of different and cooperating models is used. This tends to force users to construct, execute, validate and analyse these models. The situation is much more complicated if the user is not an IT specialist. This causes a huge human effort to actions that might be out of a scientist's scope or could be provided automatically. This work presents a tool providing an automated planning, executing and monitoring cooperating and extensive computations. The approach used introduces the HPC as a service. Modular design enables extensions and unifies the access to different HPC systems through a simple client-server interface using standard web services. The dispatch server detects and enables concurrent execution of tasks and offers a level of fault tolerance.**

***Keywords*—Automation, distributed computing, execution planning, job submission, monitoring, HPC, multiscale modelling, model coupling, service.**

I. Introduction

Simulations are considered to be the third pillar of science. To study complex phenomena such as forest fires, weather forecast or fluid dynamics, we need to describe many the processes and communication between them. These simulations are usually very demanding on computational performance and storage. Each process requests different computational resources and the whole computation may be distributed over diverse computational facilities. Big supercomputing centres offer both sufficient amount of computational power and disk space. Computing infrastructures are growing in parallelism and becoming more diverse. This heads towards using more sophisticated computational techniques to take the full advantage of the machine power. However, this tends to force users to construct, execute, validate and analyse a number of different models for each process. Furthermore, advanced knowledge of supercomputer's architecture and submission systems is required. Such a big human effort can become a bottleneck because a non-negligible number of person hours has to be invested daily, especially, if the user is not an IT specialist.

Supercomputers are built on hybrid architectures and integrates CPUs, GPUs or other accelerators. These computational components are grouped into nodes. Supercomputers use the batch system where each queue could offer different equipment, maximum time allowed, amount of allocatable nodes and price.

A usual user workflow contains:

- Creating job scripts based on the simulation workflow. Realistic and complex simulations can be described by many different cooperating models. Usually, each model corresponds to one job script. The job script is a text file specifying the executables, submission specification (e.g., job dependency, queue name, number of nodes and time), outputs and inputs, and software modules (i.e., needed software, compilers).
- Pushing jobs into queues.
- Jobs monitoring. Supercomputers do not allow any convenient notification mechanism. Only email notifications when the job has finished or an error occurred are supported which is not reliable.
- Outputs and results post-processing.

Furthermore, this workflow also requires the knowledge of accessing the supercomputer, command-line tools and manipulation with files since there is no application programming interface available. Therefore, there is a demand for an automated tool offering communication with a supercomputer, data transfer, simulation execution planning, monitoring and notification mechanisms. Such a tool could reduce the complexity of administrative tasks. The dispatch server tool, presented here, follows this approach and presents the HPC as a service. The dispatch server, shortly dispatcher, is being developed as a part of a medical software in a collaboration with the University College London. However, its modular and flexible system design enables various extensions and unifies access to different HPC systems through a simple client-server interface using standard web services. The service enables various companies or institutions to take full advantage of new computing technologies with the user experience of using personal computers only on the level of common PC user.

In Section II, the related tools following the approach of

automating the research tasks and managing computations are presented. In Section III, the architecture design of the dispatch server is presented as well as its benefits for non-developer users. Finally, Section IV summarizes the importance and benefits for ordinary users. This is shown on the application for ultrasound therapy treatment planning. Section V offers conclusions and describes the relation to my PhD thesis and its goals.

II. Related work

The functionality of the dispatch server follows features of middleware toolkits and programming interface, developed over the last decade or more, such as FabSim [1], Globus [2] or SAGA [3]. All of these tools provide job management, e.g., task submission, monitoring and cancelling, and file management on grid computing resources (virtual supercomputers) or high performance computing systems. Similarly to mentioned tools as FabSim, the dispatch server is also built on the cryptographic network protocol *ssh (secure shell)* in terms of the communication with remote machines. Practically, every Unix- or Linux-based system has *ssh*, so there is no need of installation of any additional heavyweight middleware stack on the resources being accessed. In other words, the dispatch server can be used widely on any HPC resource that supports *ssh*. Even though the main motivation comes from FabSim, FabSim is meant to be used especially by IT developers. Its interactive design provides easy single-line commands to control and monitor the computation and perform data transfers. The dispatch server is completely different and points at common user. For details see section III.

Although the idea of using the HPC as a service for various institutions is quite fresh, the dispatch server is not the only tool offering it. In the area of hydroinformatics, service framework [4] providing the HPC as a service has been developed.

Since realistic simulations could be described by a number of different cooperating models, the communication interface between these models needs to be specified. This is called model coupling and it is quite challenging if cooperating models differ in their scales and have dependencies between them. The dispatch server is a service framework providing model coupling. The coupling environments such as MUSCLE 2 [5] or MPWide [6] allow codes to exchange data at runtime efficiently, and can be used to speed up the remote execution of coupled tasks within the dispatch server. This would be completely controlled by the dispatch server.

III. Overview of the dispatch server

The dispatch server presents a middle layer between the user and remote computational resources used for computations. It offers a service providing planning and executing of HPC calculations, job and data management without any extra user interaction. Moreover, it provides user authentication and authorization, encryption, monitoring, reporting, billing and notification mechanism. The dispatch server, its database and the HTTPS server are being implemented in Python programming language. Python was chosen for its extendibility due to a reasonable amount of libraries and fast prototyping behaviour.

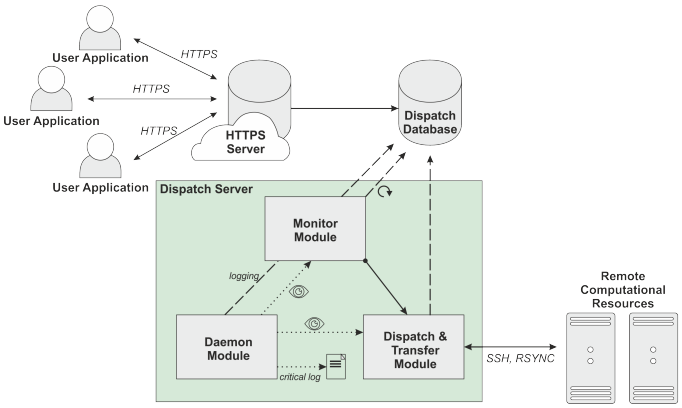


Fig. 1. The architecture of the dispatch server. The planning and monitoring core of the dispatcher consists of the daemon module, monitor module and the dispatch and transfer module (green rectangle). All of these modules are connected to the database to create or update records (dashed line). The daemon module is responsible for checking other modules for being alive (dotted line). The dispatch and transfer module provides a connection with remote resources. The HTTPS server and the dispatch database can be situated physically apart from the dispatch server machine. The HTTPS server serves as an input gate for user interaction. It is the only connection point with user applications.

A. Architecture

The dispatch server consists of three modules – the dispatch and transfer module, the monitor module, the daemon module. The HTTPS server and the dispatch database might be physically decoupled to improve reliability. However, other functions, i.e. monitoring or job management, are disabled. The architecture of the dispatch server is shown in Figure 1.

Following subsections describe details about the dispatch server modules and important terms used in the paper.

Planning file: Planning file serves as a template for dispatch server to create a simulation plan and submit jobs.

Result file: Result file is a product of the whole simulation. There is no other processing of the result file provided by the dispatch server. This file is only downloaded and stored for the user.

Simulation plan: Simulation plan is a task graph defining what simulations should be run on a particular level of a graph, dependencies and file names. This is created by the dispatch and transfer module using a planning file and details stored in the dispatch database.

Progress file: Fault tolerant mechanism stands on the progress files polling. The progress file is a text file continuously written by any running program.

HTTPS server: The HTTPS server handles incoming requests from user applications. It is the only accessible point from outside. The HTTPS server calls stored database procedures to handle requests. It also performs a user authentication and stores received planning file to a reserved temporary storage on dispatch server machine.

HTTPS server is supposed to:

- receive a planning file from the user, e.g., a desktop application, web browser, mobile application.
- provide a result file and progress of simulations.

Dispatch database: The dispatch database defines tables with users, groups of users, allocations, facilities, simulations and their jobs, and so on. It holds the history of all computations executed by the user and allocations. Furthermore, it serves as a log storage.

Daemon module: The dispatch server is supposed to run as a service in the system. This is provided by the daemon module. The daemon module is responsible for checking other modules, i.e. monitor module, dispatch and transfer module, for being alive. If any of these two modules crashes, the daemon module is responsible for restarting the corresponding module. Periodic log events are stored in the dispatch database. If a critical situation occurs, e.g. unavailability of the database, a critical log is written to a file in `/var/log` folder on the dispatch server machine. In such a case, the administrator is responsible for handling the error situation and restarting required modules.

Monitor module: Monitor module is responsible for managing the dispatch database. It periodically checks records in the database and invokes the dispatch and transfer module when

- a new simulation record occurs (to handle this issue),
- going through all simulation job records to update their statuses,
- a simulation has just finished in order to download a result file and store it temporarily on the dispatch server machine.

A couple of stored procedures are implemented. Those routines, e.g., create a new simulation record, download a result file or get simulation status, are used by the HTTPS server after a user request.

Dispatch and transfer module: The dispatch and transfer module can be invoked by the monitor module in order to

- create a simulation plan for a new simulation record, create job scripts, connect to the remote machine or machines, upload all necessary files there and submit jobs.
- check jobs statuses by connecting to the corresponding remote machine and read progress files.
- download a result file to a temporary storage. Before downloading, the estimation of free disk space on the dispatch server machine needs to be calculated.

The dispatch and transfer module is the only module that might be connected to a remote computational resource. Connection is provided using *ssh* protocol and file transfers are provided by *rsync*.

B. Remote execution

Remote executions are completely hidden from the user. Based on the knowledge in the dispatch database, the dispatch server controls the simulation execution planning and running. An example workflow (after handling a user request and receiving a planning file):

- The planning file is temporarily stored on the dispatch server machine. Based on this planning file, the simulation plan is created. According to the simulation plan and based on the information stored in the dispatch database, job scripts are created and temporarily stored in the same folder as the planning file.
- Remote disk space for the simulation and used core-hours estimation are calculated. If there is no problem, the blockage of disk space and core-hours is stored in the particular database record and the workflow continues to the next point. If the simulation cannot be fully performed due to any of these facts, or the machine is unavailable, a particular error message and status is stored in the database record. The whole process is repeated later. If a maximum number of unsuccessful attempts has been made, an error message and status are updated in the particular database record.
- A new folder belonging to the simulation is created on the remote machine. Temporary folder is completely copied to remote one.
- Depending on the complexity of the simulation, job scripts are submitted to the queues concurrently or with dependencies.

After successful job submission, monitoring is periodically made.

C. Security

The security model essentially stands on the *ssh* and *https security models*, i.e. public and private keys are used to authenticate the user and to authenticate all operations on remote machines. Security in terms of a correctly performed simulation is provided by the monitor module and checking progress files.

IV. Use case

The dispatch server is designed to be a modular and easily extendable software. A couple of applications may use it. At this time, the dispatch server will be used mainly within a medical system for administrating the simulation workflows. This software is called *k-Plan* and is being developed in a collaboration with the team from the University College London *(UCL)*. The *k-Plan* software is intended for offline model-based treatment planning for therapeutic ultrasound treatments, e.g., tissue ablation, targeted drug delivery, ultrasonic neuromodulation and opening the blood brain barrier. Simulation calculations particularly use the *k-Wave* toolbox [7]. One of the tissue ablation examples is a cancer treatment in soft tissues, e.g. breast, kidney, or prostate, using HIFU[1]. The aim is to find optimal parameters of ultrasound transducer. A user, i.e. a medical doctor, sets these parameters based on their experience. The output of the simulation tells how successful a treatment would be. In other words, the simulation output tells

[1]High Intensity Focused Ultrasound

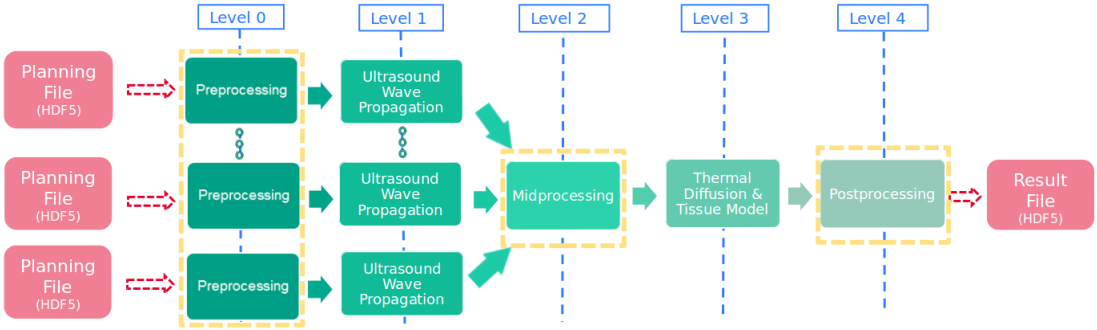


Fig. 2. Simulation plan example. Pink rectangles represent input/output of the simulation. Green rectangles represent individual stages in a pipeline. Each simulation, i.e. ultrasound wave propagation, thermal diffusion and tissue model, differs in its own nature but in computational requirements as well. Therefore, coupling interfaces have to be implemented for a correct communication between models. These coupling interfaces are represented by additional stages in the pipeline (highlighted by orange dashed rectangles). Tasks which are planned at the same level might be executed concurrently (see level 0 and 1 as an example). Tasks in different levels have to be executed sequentially because of dependencies.

whether the affected tissue has been fully destroyed. At least, the simulation consists of the ultrasound wave propagation, the thermal diffusion and the tissue model. These simulations are computed in a pipeline. To provide a correct model coupling, additional stages need to be added to this pipeline. These stages define model coupling interfaces. See Figure 2 for more details.

V. Conclusion

The overview and purpose of the dispatcher were described in this work. The approach offering the HPC as a service could reduce the complexity of daily administrative tasks. The dispatch server is a modular and easily extendable tool providing planning, executing and monitoring of cooperating computations. Furthermore, it provides an authentication, encryption, billing and a notification mechanism. It enables concurrent running of tasks and provides a level of fault tolerance. The dispatch server enables ordinary users to take the full advantage of new computing technologies and speed up their own development. This brings the novelty and the technology progress to other business fields.

The topic of my PhD thesis is to design and create model-coupling framework for complex medical simulations. The partial goals of my PhD thesis are to

1) design a tool allowing to automatically execute complex simulation codes.
2) design a coupling interface between acoustic, thermal and tissue models.
3) optimise the coupling interface to support heterogeneous architectures (CPUs, GPUs, etc.).
4) evaluate the benefits on realistic medical simulations (photoacousting imaging, HIFU treatment planning, etc.).

At this time, I'm working on point 1). The dispatch server serves me as a base component for model coupling and a start point for my research. It's designed to be a part of the *k-Plan* medical system being developed in a collaboration with UCL. However, the generic design of the dispatch server enables connecting various user applications and unifies the access to different computational resources.

Acknowledgment

I would like to thank Bradley E. Treeby and Panayiotis Georgiou for the collaboration on the dispatch server design.

This work was supported by the FIT-S-17-3994 Advanced parallel and embedded computer systems project. This work was supported by The Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science - LQ1602 and by the IT4Innovations infrastructure which is supported from the Large Infrastructures for Research, Experimental Development and Innovations project IT4Innovations National Supercomputing Center - LM2015070.

References

[1] D. Groen, A. P. Bhati, J. Suter, J. Hetherington, S. J. Zasada, and P. V. Coveney, "FabSim: Facilitating computational research through automation on large-scale and distributed e-infrastructures," *Computer Physics Communications*, vol. 207, pp. 375–385, 2016.

[2] I. Foster, "Globus toolkit version 4: Software for service-oriented systems," in *Proceedings of the 2005 IFIP International Conference on Network and Parallel Computing*, ser. NPC'05. Berlin, Heidelberg: Springer-Verlag, 2005, pp. 2–13.

[3] A. Merzky, O. Weidner, and S. Jha, "Saga: A standardized access layer to heterogeneous distributed computing infrastructure," *SoftwareX*, vol. 12, pp. 3–8, 2015.

[4] J. Martinovic, S. Kuchar, V. Svaton, and V. Vondrak, "Hydrological model remote execution and hpc as a service," in *Supercomputing in Science and Engineering*, 1st ed. Ostrava: VSB Technical University of Ostrava, 2017, pp. 97–99.

[5] J. Borgdorff, M. Mamonski, B. Bosak, K. Kurowski, M. Ben Belgacem, B. Chopard, D. Groen, P. V. Coveney, and A. G. Hoekstra, "Distributed multiscale computing with MUSCLE 2, the Multiscale Coupling Library and Environment," *Journal of Computational Science*, vol. 5, no. 5, pp. 719–731, 2014.

[6] D. Groen, S. Rieder, P. Grosso, C. T. A. M. de Laat, and S. P. Zwart, "A lightweight communication library for distributed computing," *Computational Science & Discovery*, vol. 3, no. 1, p. 15002, 2010.

[7] B. E. Treeby and B. T. Cox, "k-Wave: MATLAB toolbox for the simulation and reconstruction of photoacoustic wave fields," *Journal of biomedical optics*, vol. 15, no. 2, p. 021314, 2010.

Systémy odolné proti poruchám – metodika návrhu řadiče rekonfigurace

Richard Pánek
1. ročník, prezenční studium
školitel: Doc. Ing. Zdeněk Kotásek, CSc.

Fakulta informačních technologií Vysokého učení technického v Brně
Božetěchova 2, 612 66 Brno, Česká republika
Tel.: +420 54114-1362
Email: ipanek@fit.vutbr.cz

***Abstrakt*—Pro kritické nejen řídicí systémy je výskyt poruch velice nežádoucí záležitostí. Obzvláště pokud by mohlo dojít k újmě na zdraví nebo finančním ztrátám. Proto se rozvíjely techniky známé pod názvem systémy odolné proti poruchám. Pro zotavování z poruch je využití rekonfigurace obzvláště výhodné. Platformou schopnou rekonfigurace pro návrh a implementaci obvodů je FPGA. Pro zajištění opravy obvodu v FPGA pomocí rekonfigurace je velice výhodné využít řadič částečné dynamické rekonfigurace tedy speciální přidanou komponentu. Dále je žádoucí, aby i řadič byl odolný proti poruchám, obzvláště když bude umístěn na stejném FPGA. Právě vypracováním příslušných kritérií a návrhem tohoto řadiče se bude zabývat metodika, která bude také tématem disertační práce.**

***Klíčová slova*—Řadič rekonfigurace, systémy odolné proti poruchám, částečná dynamická rekonfigurace, FPGA.**

I. Úvod

V dnešní době nás obklopují elektronická zařízení v nejrůznějších přístrojích všeho druhu. Podle jejich určení se klade důraz na výkon, spotřebu, cenu, atd. Ovšem existují také aplikace, kde je potřeba zajistit spolehlivost. Ta je vyžadována obzvláště u systémů, kde by mohlo dojít k újmě na životech nebo na financích. Typickými zástupci takových aplikací jsou řídicí systémy, které se starají o řízení letadel, družic, elektráren, ale také třeba nemocničních přístrojů a mnoha dalších. Je zřejmé, že se jedná o systémy, které musí pracovat bezchybně anebo se musí umět z vlastních poruch zotavit, popř. i přes poruchu pracovat správně.

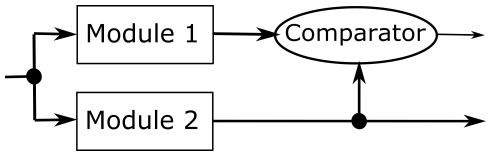
Tento článek je dále uspořádán následovně. Sekce II je zaměřena na uvedení do problematiky a objasnění pojmu *Odolnost proti poruchám*. Je zde vysvětleno možné dělení a také základní přístupy. Sekce III se věnuje řadiči částečné dynamické rekonfigurace pro FPGA. Kromě obecných možností je zaměřena na konkrétní implementaci a také na budoucí výzkum, který bude pokračovat nad touto problematikou. Obzvláště se jedná o možnosti jeho zabezpečení a tím zvýšení jeho spolehlivosti. Sekce IV pojednává o posledním nezabezpečeném prvku volícím majoritu. V sekci V jsou nastíněny cíle disertační práce. Závěrečné shrnutí je v sekci VI.

II. Odolnost proti poruchám

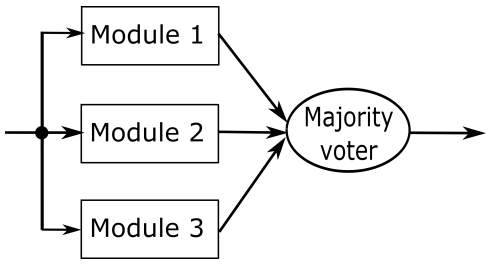
Odolnost proti poruchám (Fault tolerance) [5] je přístup, kdy je systém schopen pracovat dle specifikace i přes výskyt poruch. Snižování dopadu poruch je možné dosáhnout využitím prostorové, časové nebo datové redundance. Jejich volba záleží na požadavcích výsledné aplikace. Prostorová redundance znamená několikanásobný výskyt stejných komponent, které pracují současně. Oproti tomu časová redundance je dána prováděním stejného výpočtu několikrát na stejné komponentě. Datovou redundancí je myšleno opatření dat opravným kódem, který zajistí jejich opravu při výskytu chyb. Samozřejmě je možné výše nastíněné techniky kombinovat a vytvořit tak zabezpečení dle požadavků na výslednou aplikaci.

Dalším možným členěním odolnosti proti poruchám je rozlišování pasivních a aktivních metod. Pasivní metody [4] jsou založeny na předpokladu, že je možné zjistit všechny možné poruchy a pro každou z nich předem nachystat řešení, které snižuje její dopad. U jednoduchých problémů takové omezení nemusí vadit, ale pro rozsáhlé aplikace může být značně náročné analyzovat takové poruchy nebo mít připravené a uložené potřebné opravy a náhrady může být značně prostorově náročné. Ovšem při výskytu poruchy nevzniká zbytečná prodleva do znovuobnovení funkcionality. To je zásadní rozdíl od aktivního přístupu, kdy příslušnou poruchu je třeba analyzovat a následně připravit řešení na míru za běhu. I když odpadá potřeba mít připravené kompenzace napravující poruchu tím, že se počítají za běhu, jejich výpočet zabere určitý výpočetní čas, kdy je zařízení v nedefinovaném stavu. Odstraněním zásadních problémů a kombinací obou přístupů se zabývá hybridní metoda [11], kterou je možné shrnout v následujícím postupu. V aplikaci je detekována porucha. Využije se pasivní přístup, kdy je porucha co nejlépe kompenzována a současně je spuštěna její diagnostika. Po ní je možné připravit opravu na míru. Až je oprava připravena, tak je aplikována pomocí rekonfigurace systému. Tímto způsobem je možné opravit i předem neočekávané poruchy a zároveň je odstraněna doba nedefinované funkce. Jsou tedy využity silné stránky jednotlivých přístupů a současně potlačeny podstatné nedostatky.

Výše zmíněné přístupy lze aplikovat i na programovatelná hradlová pole FPGAs *Field Programmable Gate Arrays* [7], která se stále více uplatňují v nejrůznějších systémech díky svému výpočetnímu výkonu a schopnosti změnit svou konfiguraci pro přizpůsobení se aktuálním podmínkám. Jejich hlavní nevýhodou je náchylnost nejen na kosmické záření, které způsobuje poruchy konfigurační paměti. Dle intenzity a doby působení může způsobit přechodné, ale i trvalé poruchy. Typickou poruchou je překlopení jednoho náhodného bitu libovolné paměti např. té konfigurační. Tato porucha je známa pod označením *Single Event Upset (SEU)*. Odolnost proti poruchám může být zajištěna jejich maskováním pomocí zdvojení s porovnáním *(DwC – Duplication with Comparison)* podle schématu na obrázku 1 nebo pomocí tří-modulové redundance *(TMR – Triple Modular Redundancy)* podle schématu na obrázku 2. V modulech může být obsažen jak celý obvod, tak i jen jeho důležité částí. Dále je také možné využít zřetězení příslušných přístupů a rozdělit tak obvod na menší zabezpečené části.



Obrázek 1. Schéma DwC



Obrázek 2. Schéma TMR

Další možností, která již skutečné provádí opravu poruch, je přístup, kdy je konfigurační informace daného FPGA přepsána pomocí správné konfigurace *(golden bitstream)*. Tento přístup, při kterém dochází k rekonfiguraci FPGA, se nazývá *Scrubbing*. Pro uložení *golden bitstreams* je potřeba mít k dispozici radiačně odolnou paměť. Takový požadavek je možné zajistit využitím speciálního hardware nebo opatřením dat opravným kódem. Jiným přístupem je *Lazy Scrubbing* [3], který využívá násobného výskytu stejných konfigurací v rámci TMR. Když je na FPGA daná část v TMR, pak každý modul je vytvořen díky stejné konfiguraci, která je tudíž uložena v konfigurační paměti FPGA také třikrát. V takovém případě už není potřeba paměť s *golden bitstreams* a v případě poruchy v jednom z těchto modulů, je konfigurační informace příslušného porouchaného modulu přepsána pomocí majoritního výskytu konfiguračních informací z těchto tří stejných modulů. Neboli z konfigurační paměti FPGA jsou přečteny příslušné konfigurační informace všech tří modulů. Dále je pro každý odpovídající si bit určena majoritní hodnota a následně je takto vytvořený bitstream opět nahrán do konfigurační paměti díky částečné rekonfiguraci [9] tzn. možnosti rekonfigurovat jen určitou část FPGA. Obstarat vše potřebné k částečné dynamické rekonfiguraci by měl její řadič, tedy speciální přidaná komponenta.

III. Řadič částečné dynamické rekonfigurace

Pro aktivní přístup k odolnosti proti poruchám je řadič částečné dynamické rekonfigurace přímo nezbytnou součástí, bez které se nelze v systému obejít. Nejjednodušší variantou je řadič, který provádí *Sbrubbing* s předem danou periodou. Preventivně přepisuje konfigurační paměť a tím také opraví případnou poruchu. Je zřejmé, že takový přístup není příliš efektivní, ale zvýší spolehlivost a nezabere příliš velkou plochu na FPGA. Také nebude příliš náročné jej použít, protože lze přidat k libovolnému již vytvořenému systému prakticky bez modifikace tohoto systému. Zásadní vylepšení přineslo IP jádro od Xilinx *Soft Error Mitigation Controller* [10] pro jejich FPGA řady 7. Konfigurační paměť byla opatřena opravným kódem a tudíž je přidána možnost provádět rekonfiguraci jen v případě výskytu poruchy a navíc opravovat jen porouchanou část. Stejně jako u předchozího přístupu není nutné upravovat zabezpečovaný systém. Ovšem vzniká zde prodleva aktivního přístupu, která je způsobena diagnostikou poruchy, tedy počítáním syndromu poruchy opravného kódu a následně určením porouchaného bitu konfigurační paměti. Jinou možností, která již prodlevou s nedefinovaným výstupem systému netrpí, je propojení tří-modulové redundance a řadiče částečné dynamické rekonfigurace [2]. Zabezpečovaný obvod je nutné uzpůsobit do TMR. Což znamená ztrojnásobení komponent a přidání prvků určujících majoritu ze vstupních hodnot. Dále je nutné opatřit tyto prvky také logikou, která je schopná určit také modul s odlišným výstupem (vstupem prvku počítajícím majoritu). Tuto hodnotu je třeba vyvést na další výstup, který je nutné přidat. Tím se samotný prvek určující majoritu značně zesložití, ale je to potřeba, aby řadič rekonfigurace dostal co nejpřesnější informaci o poruše. S touto informací a dobře namapovaným systémem na FPGA je řadič schopen zajistit dynamickou rekonfiguraci jen porouchaného modulu. Současně s opravou zbylé dva moduly TMR pracují dle specifikace a tudíž i celý systém pracuje dle specifikace. Protože prvek určující majoritu bude maskovat výstup jak porouchaného tak i současně opravovaného neboli rekonfigurovaného modulu, jehož výstup nebude po tento čas korektní. Právě takto, jak bylo popsáno, je navržen a implementován *Generic Partial Dynamic Reconfiguration Controller (GPDRC)* [8].

A. Současný stav výzkumu v oblasti řadiče částečné dynamické rekonfigurace

V rámci výzkumné skupiny zabývající se odolností proti poruchám byly postupně navrženy a implementovány dvě verze řadiče částečné dynamické rekonfigurace pro obecné použití na FPGA pod názvem *GPDRC*. První verze [8] je vytvořena tak, aby umožňovala eliminaci přechodných poruch,

jako jsou např. SEU. Pro diagnostiku a přechodné maskování poruch do doby dokončení opravy pomocí rekonfigurace je TMR aplikována na zabezpečovaný systém. V rámci druhé verze řadiče [6] byla původní rozšířena o podporu zotavení systému i z trvalých poruch. Ta je způsobena fyzickým poškozením libovolné části FPGA. Na úrovni tranzistorů dochází k negativní změně jejich vlastností a to např. trvalému zprůchodnění bez ohledu na přiváděné napětí na jejich bázi. U konfigurační paměti to znamená nemožnost změnit stav daného poškozeného uloženého bitu a tedy jeho setrvání v jednom z možných stavů. Důsledkem je nemožnost změnit konfiguraci určité části FPGA, která je ovlivněna právě tímto poškozeným bitem. Pro zotavení se z takové poruchy jsou v FPGA vyhrazeny rezervní bloky. Při odhalení, že se jedná o trvalou poruchu, je příslušný modul vyřazen a nahrazen jinde vyhrazeným náhradním blokem, do kterého je nakonfigurována funkcionalita příslušného vyřazeného modulu. Trvalá porucha je odhalena tak, že i po rekonfiguraci není výstup modulu shodný s ostatními TMR moduly. V případě více se vyskytujících trvalých poruch časem nastane situace, kdy už další rezervní modul není k dispozici. V takovém případě zabezpečení degraduje z TMR na pouhé zdvojení se srovnáním DwC. To už není schopné poruchu lokalizovat přesně, jen ji dokáže oznámit. Následná trvalá porucha způsobí definitivní konec korektní činnosti systému.

B. Budoucí výzkum v oblasti řadiče částečné dynamické rekonfigurace v FPGA

Předchozí výzkum se soustředil na zabezpečení obvodu. Ovšem pokud bude řadič na stejném FPGA jako zabezpečovaný obvod, je velmi pravděpodobné, že může být poruchou zasažen také. V takovém případě není vyloučeno, že řadič s poruchou může způsobit neočekávaným chováním i poškození jinak korektně fungujícího obvodu. Např. by mohl provést rekonfiguraci, která negativně pozmění funkčnost zabezpečovaného obvodu. Proto bude další výzkum směřovat k zabezpečení také samotného řadiče.

I pro řadič rekonfigurace je vhodné využít podobné techniky jako pro zabezpečovaný obvod. A to využít TMR pro maskování poruch jednotlivých modulů s řadiči. Řadič tedy bude v systému třikrát a bude doplněn prvkem určujícím majoritu z jednotlivých výstupů, neboli dat posílaných do konfigurační paměti v rámci rekonfigurace. Tak je vnímána první etapa zabezpečování řadiče. Lze očekávat nepříliš velké zvýšení spolehlivosti, protože řadič zabere trojnásobné místo na FPGA a tudíž se zvýší pravděpodobnost zásahu poruchou. Ovšem oproti tomu bude možná jednonásobná porucha a ve vhodných případech i vícenásobná porucha maskována díky majoritě.

Další etapou zabezpečování bude zavedení rekonfigurace i pro samotné řadiče. Vzhledem ke třem instancím řadiče již se nacházejícím na FPGA by bylo velice vhodné využít právě je. Vize je taková, že předpokládáme poruchu v jednom z modulů ztrojeného řadiče rekonfigurace a tedy v jednom ze tří řadičů v TMR. Tato porucha je rozpoznána díky prvku počítajícímu majoritu a jsou tudíž o ní informovány všechny tyto řadiče. Zbylé dva korektně fungující řadiče se postarají o rekonfiguraci třetího porouchaného. Je vhodné, aby řadič s poruchou do vlastní rekonfigurace nezasahoval. Je nutné zajistit jeho odpojení a tudíž zabránění v činnosti anebo by mělo být postačující využít schopnosti prvku určujícího majoritu, který zajistí maskování chybného výstupu z porouchaného řadiče. I v případě provádění rekonfigurace pouze dvěma funkčními řadiči je možné zjišťovat, zda nenastala porucha. Jedná se o metodu DwC, která ovšem už není schopná určit, ve kterém ze dvou řadičů se porucha projevila. Možným předejitím nastání takové situace je přidat na FPGA ještě jednu instanci řadiče. Celkový počet řadičů by byl čtyři. V případě rekonfigurace jednoho z nich by pořád zbývali tři dobře fungující a tudíž pracující v režimu TMR. I při výskytu další poruchy by ta byla maskována a následně by mohl být nově porouchaný řadič opraven díky opravenému řadiči rekonfigurace a zbylým dvěma korektně fungujícím řadičů. I toto řešení by bylo vhodné otestovat v rámci experimentů.

C. Programová implementace řadiče částečné dynamické rekonfigurace

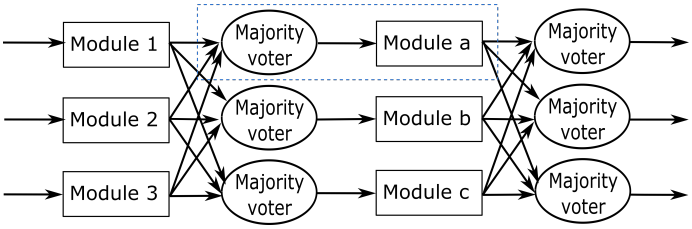
Kromě výše zmíněné implementace GPDRC přímo v hardware se nabízí i alternativa v podobě programové implementace pro procesor. Podstatnou výhodou takového řešení bude odolnost konfigurace proti SEU, pokud tedy nebude procesor implementován v FPGA jako *měkké jádro (soft core)*. V takovém případě by se pro jeho zabezpečení využily stejné techniky, které byly navrženy v předchozím odstavci. Ovšem pokud bude procesor externí součást nebo bude v FPGA v podobě *těžkého jádra (hard core)*, pak mu poruchy konfigurační paměti nehrozí. Jediné, co může SEU způsobit, jsou poruchy v paměťových blocích a tedy chyby v datech potřebných pro činnost. Jedná se o paměť s instrukcemi programu, operační paměť a také registry. Ovšem pro zabezpečení dat proti chybám se nevyužívá rekonfigurace, ale opravné kódy, kterými se data musí opatřit. Lze využít i časové redundance, ale muselo by se zajistit, aby nebylo počítáno opakovaně se stejnými poškozenými daty.

V případě externí součástky jak procesoru tak jiného FPGA s řadičem, je zřejmé, že se prodlouží datové cesty. Tím se prodlouží také čas, který řadič bude potřebovat pro získání informací pro diagnostiku poruchy. Následně bude časově náročnější i samotná rekonfigurace, která bude z poruchy obvod zotavovat. To vše povede k většímu zpoždění opravy a snížení rychlosti výpočtu samotného obvodu. Ovšem záleží na požadavcích na výsledný systém, protože se ušetří plocha na FPGA a také by mohl být procesor využit i k jiné činnosti než pouze pro řadič, když by měl dostatečný výpočetní výkon.

IV. Zabezpečení prvků určujících majoritu

Posledním slabým místem z hlediska spolehlivosti zůstávají prvky určující majoritu u TMR. Ty nejsou nijak zabezpečeny a tudíž při poruše mohou na svém výstupu mít předem neočekávanou hodnotu. Tím mohou způsobit nedefinované chování celého systému. Sice je jejich velikost prakticky zanedbatelná oproti výpočetnímu obvodu, ale i tak při dlouhé době životnosti obvodu šance, že je porucha zasáhne, roste.

Možné zabezpečení je představeno v [1], kde i tyto prvky jsou ztrojeny, což je znázorněno na obrázku 3. Modrou přerušovanou čarou je znázorněn blok, který je třeba rekonfigurovat v případě zjištění poruchy. Oproti původní variantě je v něm zahrnut i jeden prvek určující majoritu. Je to dáno tím, že nelze rozlišit, jestli je porucha v prvku určujícím majoritu anebo v následném modulu. Vše je totiž diagnostikováno až v následující vrstvě prvků určujících majoritu.



Obrázek 3. Schéma TMR se zabezpečenými prvky volícími majoritu

Nelze takto ale zabezpečit poslední instanci po zřetězení, protože je na výstupu potřeba pouze jedna správná hodnota. V tomto případě je výstup ztrojen díky ztrojení prvků počítajících majoritu. Nezbývá než toto poslední zabezpečení omezit a nasadit pouze hlídací obvod popřípadě využít metodu DwC. Při zjištění poruchy je ale nutné výpočet obvodu přerušit do doby, než se pomocí rekonfigurace příslušných zabezpečujících prvků znovu neobnoví korektní funkcionalita. Stejný problém je i u hlídání výstupů z řadičů rekonfigurace v TMR, kdy je opět potřeba jen jeden výstup pro zápis do konfigurační paměti FPGA.

V. Cíle disertační práce

V rámci disertační práce se zaměřuji na vypracování metodiky pro použití řadiče částeční dynamické rekonfigurace pro systémy odolné proti poruchám. Především budu navrhovat a experimentovat s různými kritérii pro návrh, implementaci a samotné používání řadiče. Zatím známá kritéria jsou spolehlivost, rychlost a zpoždění, spotřeba, zabraná plocha na FPGA. Další mohou být identifikována v průběhu výzkumu. Je zřejmé, že jsou vzájemně protichůdná a tak předpokládám vznik různý paretooptimálních řešení, která budou v rámci metodiky diskutována. Zejména jejich přínos pro různé požadavky aplikací.

Vycházím z již vytvořené instance GPDRC, která je výsledkem předchozí práce výzkumné skupiny. Tento řadič budu zabezpečovat pomocí výše nastíněných principů. Dále pro porovnání počítám také s vytvořením implementace pro procesor a variantou s řadičem mimo FPGA s aplikací. Všechny tyto přístupy budou podrobeny experimentům a budou diskutovány přínosy a úskalí, které budou potřeba pro vypracování metodiky.

VI. Závěr

V rámci tohoto článku byly diskutovány základní principy pro odolnost proti poruchám se zaměřením především na FPGA. Obzvláště bylo pojednáváno o prostorové redundanci zajištěné pomocí principu TMR. Ten byl dále rozšířen o možnost částeční dynamické rekonfigurace s využitím přidané diagnostiky do TMR. Tím je možné identifikovat modul TMR, který se má rekonfigurací opravit. Tato rekonfigurace nenaruší činnost obvodu, protože zbylé dva moduly fungují korektně. Vše potřebné k rekonfiguraci musí zajisti její řadič, jehož příkladem je GPDRC vyvinutý v rámci výzkumné skupiny. Ten bude v rámci budoucí práce zabezpečen pomocí zde představených principů. Také bude vytvořena implementace pro procesor, která bude sloužit k porovnání. Vše vede k disertační práci, ve které bude zpracována metodika pro použití řadiče rekonfigurace pro systémy odolné proti poruchám. Zejména půjde o vypracování kriterií pro porovnání jednotlivých přístupů a provedení experimentů, které ukáží, jaký přístup bude pro splnění daných požadavků nejvhodnější.

Poděkování

Tato práce byla podporována Ministerstvem školství, mládeže a tělovýchovy z Národního programu udržitelnosti (NPU II); projektu IT4Innovations excellence in science – LQ1602. Tato činnost byla rovněž podporována projekty řešenými na VUT v Brně pod číslem FIT-S-14-2297.

Reference

[1] Abraham, J. A.; Siewiorek, D. P.: An Algorithm for the Accurate Reliability Evaluation of Triple Modular Redundancy Networls. *IEEE Transactions on Computers*, ročník c-23, č. 7, červenec 1974: s. 682–692.

[2] Bolchini, C.; Miele, A.; Santambrogio, M. D.: TMR and Partial Dynamic Reconfiguration to mitigate SEU faults in FPGAs. *22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT 2007)*, září 2007, ISSN 1550-5774, s. 87–95, doi:10.1109/DFT.2007.25.

[3] Garvie, M.: *Reliable Electronics through Artificial Evolution*. Disertační práce, University of Sussex, leden 2005.

[4] Jiang, J.; Yu, X.: Fault-tolerant control systems: A comparative study between active and passive approaches. *Annual Reviews in Control*, ročník 36, č. 1, 2012: s. 60–72, ISSN 1367-5788, doi: http://dx.doi.org/10.1016/j.arcontrol.2012.03.005.

[5] Koren, I.; Krishna, C. M.: *Fault-Tolerant Systems*. Elsevier, 2007, ISBN 9780120885251.

[6] Miculka, L.; Kotasek, Z.: Generic partial dynamic reconfiguration controller for transient and permanent fault mitigation in fault tolerant systems implemented into FPGA. *17th International Symposium on Design and Diagnostics of Electronic Circuits Systems*, duben 2014, s. 171–174, doi:10.1109/DDECS.2014.6868784.

[7] Siegle, F.; Vladimirova, T.; Ilstad, J.; aj.: Mitigation of Radiation Effects in SRAM-Based FPGAs for Space Applications. *ACM Comput. Surv.*, ročník 47, č. 2, leden 2015: s. 37:1–37:34, ISSN 0360-0300, doi: 10.1145/2671181.
URL http://doi.acm.org/10.1145/2671181

[8] Straka, M.; Kastil, J.; Kotasek, Z.: Generic partial dynamic reconfiguration controller for fault tolerant designs based on FPGA. *NORCHIP 2010*, listopad 2010, s. 1–4, doi:10.1109/NORCHIP.2010.5669477.

[9] XILINX: Partial Reconfiguration User Guide. Dostupné z: http://www.xilinx.com/support/documentation/sw_manuals/xilinx14_1/ug702.pdf, duben 2012 [cit. 2017-06-07].

[10] XILINX: Soft Error Mitigation Controller v4.1. Dostupné z: https://www.xilinx.com/support/documentation/ip_documentation/sem/v4_1/pg036_sem.pdf, duben 2017 [cit. 2017-06-12].

[11] Yu, X.; Jiang, J.: Hybrid Fault-Tolerant Flight Control System Design Against Partial Actuator Failures. *IEEE Transactions on Control Systems Technology*, ročník 20, č. 4, červenec 2012: s. 871–886, ISSN 1063-6536, doi:10.1109/TCST.2011.2159606.

Usmerňovače pre vysofrekvenčný zberač energie integrovaný na čipe

Ing. Miroslav Potočný
1. ročník denného štúdia
prof. Ing. Viera Stopjaková, PhD.

Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Ilkovičova 3, 812 19 Bratislava
miroslav.potocny@stuba.sk

***Abstrakt*— Tento príspevok sa zaoberá usmerňovačmi pre systém vysokofrekvenčného zberača energie integrovaného na čipe v štandardnej CMOS technológii. Dôraz je kladený na porovnanie existujúcich topológií bežne používaných v súčasných bezdrôtových integrovaných obvodoch. Zvláštna pozornosť je venovaná parametrom tranzistorov, ktoré zhoršujú efektivitu výkonovej konverzie a zvyšujú hodnotu minimálneho potrebného vstupného výkonu obvodu. Z týchto parametrov sa ako kľúčové ukazuje najmä prahové napätie tranzistorov. Na jeho potlačenie sa v súčasnosti používajú zapojenia využívajúce externe alebo interne generované jednosmerné napätie na posunutie pracovného bodu tranzistorov, čím je možné významne zlepšiť vlastnosti usmerňovačov a to hlavne pre malé vstupné výkony.**

Kľúčové slová*— *vysokofrekvenčný usmerňovač, zberač vysokofrekvenčnej energie, bezdrôtový prenos výkonu

I. ÚVOD

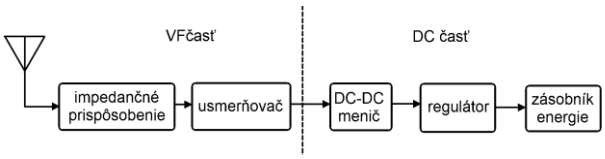
Rozvoj súčasných aplikácií bezdrôtových obvodov, napríklad v oblasti medicíny, rádiofrekvenčnej identifikácie alebo internetu vecí, vytvára požiadavku na neustále zmenšovanie ich rozmerov, ceny, spotreby a zároveň predlžovanie ich životnosti. Rozmery takýchto elektronických systémov sú závislé najmä od použitej antény a batérie, voči ktorým sú rozmery samotného integrovaného obvodu zanedbateľné. Životnosť týchto systémov je daná najmä použitou batériou a spotrebou integrovaného obvodu.

Ako možné riešenie pre bezdrôtové aplikácie elektronických obvodov sa ukazuje využitie systémov na zber energie z okolia. Toto umožňuje výrazne predĺžiť životnosť batérie alebo dokonca jej úplné nahradenie, čím sa podstatne znižuje cena systému. Využiteľných je zvyčajne viacero zdrojov energie, napríklad solárna energia, tepelné zdroje, vibrácie alebo vysokofrekvenčná (VF) energia spôsobená elektromagnetickým poľom. Tento článok sa zaoberá zberom VF energie so zameraním na usmerňovače používané na tento účel. Hlavným cieľom je zvýšiť účinnosť usmerňovačov pre malé vstupné výkony. Zároveň je potrebné aby bol obvod integrovateľný na čip v štandardnej CMOS technológii, čo obmedzuje výber použiteľných súčiastok a topológií obvodov.

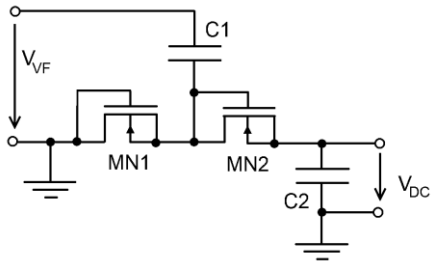
Časti systému zberača energie sú opísané v kapitole II. Stručne sú tu popísané hlavné charakteristiky a problémy usmerňovačov používaných v týchto typoch obvodov. Kapitola III je zameraná na zapojenia usmerňovačov používané v súčasných systémoch bezdrôtového prenosu energie. Hlavným cieľom týchto topológií je potlačenie prahového napätia tranzistorov, ktoré sa javí byť hlavným zdrojom strát v usmerňovačoch. Zaoberá sa aj nežiadúcim vplyvom na záverný prúd tranzistorov, ktorý je často zhoršený znižovaním prahového napätia. Kapitola IV uvádza ciele ďalšieho výskumu v rámci dizertačnej práce.

II. SYSTÉM NA ZBER VYSOKOFREKVENČNEJ ENERGIE

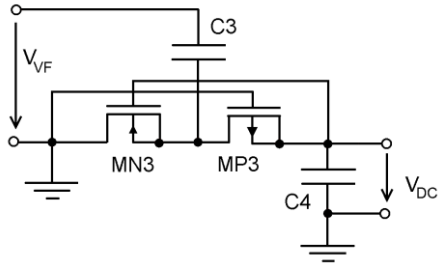
Hlavné časti systému na zber vysokofrekvenčnej energie z okolia sú zobrazené na obrázku 1. Takýto systém sa v súčasnosti využíva napríklad v obvodoch RFID alebo v bezdrôtových senzoroch používaných v zdravotníctve. Obvod DC-DC meniča nie je vždy zahrnutý do takéhoto systému, vzhľadom na problémy s integrovateľnosťou meničov. Celý systém sa dá rozdeliť na VF časť a jednosmernú časť. VF časť sa skladá z antény, obvodu impedančného prispôsobenia a usmerňovača. Typ a usporiadanie použitej antény záleží od toho, či bude systém využívať blízke alebo ďaleké elektromagnetické (EM) pole. Pri využití blízkeho EM poľa sa často používajú vysielacie a prijímacie cievky, ktoré sa správajú ako transformátor. Nejde teda o klasickú anténu a energia je prenášaná induktívnou väzbou. Pri využití ďalekého poľa sa používajú miniaturizované verzie známych antén, najmä dipóly a monopóly. V oboch prípadoch je potrebný obvod na impedančné prispôsobenie, čím sa zabezpečí prenos maximálneho výkonu do usmerňovača. Prispôsobovacie obvody sa odlišujú pre rôzne druhy väzby. Metodika návrhu prispôsobenia pre induktívnu väzbu je opísaná v [1], kde sú definované vzťahy pre dosiahnutie ideálnej hodnoty impedancie záťaže pre prijímaciu cievku. Jednoduché prispôsobenie pre klasické antény je uvedené v [2]. Tu je predpokladaný kapacitný charakter impedancie antény aj usmerňovača, ktoré sú potlačené cievkou. Pre návrh prispôsobenia je potrebné poznať frekvenčné priebehy impedancie antény (alebo cievky) ako aj vstupnej impedancie usmerňovača, takže sa mu môžeme venovať až po navrhnutí



Obr. 1. Bloková schéma zberača vysokofrekvenčnej energie



Obr. 2. Zapojenie usmerňovača založeného na Dicksonovej nábojovej pumpe.



Obr.3. Zapojenie usmerňovača založeného na Dicksonovej nábojovej pumpe s potlačením vplyvu prahového napätia.

samotného usmerňovača. Zvyšok systému tvorí jednosmerná časť ktorá sa zvyčajne skladá z DC-DC meniča, regulátora a napokon zásobníka energie. Pre usmerňovač sú dôležité parametre DC-DC meniča, hlavne jeho minimálne vstupné napätie a vstupná impedancia. Práve tomuto sa plánujeme venovať v budúcnosti. Ako zásobník energie je možné použiť buď batériu alebo kondenzátor.

III. ZAPOJENIA VF USMERŇOVAČOV

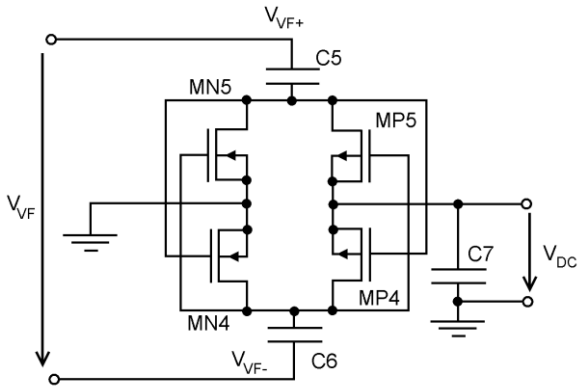
Cieľom súčasného výskumu je nájsť najvhodnejšiu topológiu usmerňovača pre malé vstupné výkony. Ako hlavný parameter z hľadiska porovnávania je efektivita konverzie VF výkonu na jednosmerný výkon, ktorá je zadefinovaná rovnicou (1).

$$EKV=P_{DC}/P_{VF} \quad (1)$$

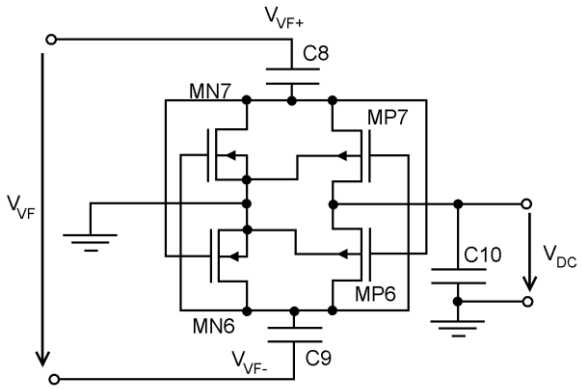
Táto účinnosť je závislá od strát v obvode usmerňovača, ktoré sú podrobne analyzované v [2]. Najväčší vplyv majú parametre polovodičových prvkov, a to ich prahové napätie a záverný prúd. V diskrétnych obvodoch sa preto na usmerňovanie často používajú Schottkyho diódy, ktoré majú lepšie najmä prahové napätie. Tieto diódy však nie sú dostupné v bežných CMOS technológiách, preto uvažujeme iba s použitím MOS tranzistorov zapojených ako diódy. MOS tranzistory majú v bežných technológiách prahové napätie rádovo v stovkách mV, čo je približne rovnaká hodnota ako amplitúda vstupného VF signálu pre usmerňovač. Tieto sa pohybujú rádovo na úrovni stoviek mV až niekoľkých V, v závislosti od aplikácie. Straty spôsobuje aj parazitná kapacita tranzistorov a odpor kanála tranzistora. Odpor sa dá potlačiť použitím tranzistora so širším kanálom, toto ale naopak zvyšuje parazitnú kapacitu, ktorá rastie s plochou tranzistora. Preto je dôležité nájsť optimálnu hodnotu rozmerov tranzistorov. Najdôležitejší parameter pre malé vstupné výkony je však prahové napätie tranzistorov, preto sme sa zamerali tie na topológie usmerňovačov, ktoré potláčajú jeho vplyv.

A. Dicksonova nábojová pumpa

Základné zapojenie Dicksonovej nábojovej pumpy bolo prvý raz uverejnené v [3]. Stalo sa populárnym na usmerňovanie VF signálov počas rozvoja systémov bezdrôtovej identifikácie. Jej najjednoduchšia verzia je zobrazená na obrázku 2. Vstupný VF signál je cez oddeľovací kondenzátor C1 privádzaný medzi dva NMOS tranzistory MN1 a MN2, ktoré sú zapojené ako diódy. Obvod pracuje v dvoch krokoch: počas zápornej polvlny vstupného signálu je otvorený tranzistor MN1, zatiaľ čo MN2 je zatvorený. Na kondenzátore C1 sa zhromažďuje náboj ktorý vytvára jednosmernú zložku napätia, ktorá má veľkosť amplitúdy VF signálu, zníženého o prahové napätie tranzistora MN1. Počas kladnej polvlny sa otvára tranzistor MN2, kým tranzistor MN1 je zatvorený. Vstupné napätie, posunuté o vzniknutú jednosmernú zložku, sa prenáša na výstupný kondenzátor C2. Opäť nastáva úbytok na tranzistore, preto je výstupné napätie rovné dvojnásobku vstupnej amplitúdy zmenšenej o dvojnásobok hodnoty prahového napätia NMOS tranzistora. Detailnejší opis činnosti tohto zapojenia ako aj vplyv parazít je možné nájsť v [3]. Výhodami tohto zapojenia sú hlavne jeho jednoduchosť a skutočnosť že zdvojnásobuje amplitúdu vstupného signálu. Taktiež je to možnosť jednoduchého kaskádovania obvodu, ktoré ďalej zvyšuje výstupné napätie za cenu menšej efektivity. Nevýhodou je naopak veľký vplyv prahového napätia MOS tranzistorov na hodnotu výstupného napätia. Existuje niekoľko možností potlačenia tohto vplyvu. Prvou a najjednoduchšou je použitie tranzistorov s nízkou hodnotou prahového napätia [4]. Táto možnosť je však závislá od použitej technológie, nie je teda univerzálna. Naviac nevýhodou je taktiež väčší záverný prúd takýchto tranzistorov v porovnaní s bežnými NMOS tranzistormi. Ďalšou možnosťou je využitie tranzistorov s dynamickým prahovým napätím (DTMOS) [5]. Na rozdiel od klasického zapojenia MOS tranzistora ako diódy (substrátová elektróda prepojená s emitorom), je substrátová elektróda prepojená s hradlom tranzistora. V takomto zapojení je prahové napätie závislé od napätia medzi emitorom a substrátom (V_{BS}), čo nám ho dovoľuje meniť. Ďalšou výhodou je možnosť využitia PMOS tranzistora ako jednej z diód (namiesto MN2 na obr. 2.). Iný prístup je založený na privedení DC napätia približne rovného prahovému napätiu na hradlo tranzistorov [6]. Takýmto spôsobom sa dá znížiť minimálny vstupný výkon potrebný pre naštartovanie obvodu. Takéto riešenie ale vyžaduje generovanie tohto napätia z externého zdroja, a preto nie je



Obr. 4. Zapojenie CMOS mostíkového usmerňovača so symetrickým vstupným VF signálom.



Obr. 5. Vylepšené zapojenie CMOS mostíkového usmerňovača s potlačením prahového napätia riadením substrátovej elektródy PMOS tranzistorov.

vhodné pre systémy bez batérie. Nevýhodou je aj zvýšený záverný prúd cez tranzistor. V [7] bolo navrhnuté riešenie ktoré využíva usmerňovačom generované jednosmerné napätie na potlačenie vplyvu prahového napätia. Schéma takéhoto zapojenia ja na obrázku 3. Hradlo NMOS tranzistora MN3 je prepojené na výstupný uzol, zároveň je hradlo PMOS tranzistora MP3 pripojené na zem. Týmto sa zvyšuje napätie V_{GS} tranzistora o hodnotu generovaného jednosmerného napätia, čo znižuje hodnotu prahového napätia oboch tranzistorov. Na rozdiel od riešenia v [6] si toto nevyžaduje žiadne ďalšie obvody, ktoré zvyšujú celkovú spotrebu a tým znižujú účinnosť obvodu. Toto riešenie zvyšuje efektivitu pre malé VF signály, ale nezmenšuje minimálnu hodnotu vstupnej amplitúdy signálu, pretože na potlačenie vplyvu prahového napätia sa využíva generované jednosmerné napätie (nulové na začiatku činnosti obvodu). Taktiež pri veľkých amplitúdach VF signálu nastáva nárast záverného prúdu, pretože prahové napätie je príliš potlačené, podobne ako pri riešení v [6].

B. Mostíkové zapojenia

Mostíkové usmerňovače nie sú vhodné na spracovanie malých amplitúd VF signálu, pretože pokles napätia je rovný súčtu hodnôt dvoch prahových napätí tranzistora. Na potlačenie tohto javu bolo v [8] navrhnuté zapojenie mostíka s dvomi NMOS a dvomi PMOS tranzistormi, ktoré majú do kríža prepojené hradlá. Hradlá sú budené v protifáze voči emitorovej elektróde. Schéma takéhoto zapojenia je na obrázku 4. Obvod pracuje podobne ako Dicksonovo zapojenie, ale spracováva symetrický vstupný signál, čo je výhodné pri použití symetrickej antény, ktorá je často využívaná v systémoch bezdrôtového prenosu energie (napríklad dipól pre väzbu ďalekým elektromagnetickým poľom alebo cievka s vyvedeným stredom vinutia pri induktívnej väzbe). Počas kladnej polvlny sú tranzistory MP5 a MN4 otvorené, zatiaľ čo tranzistory MP4 a MN5 sú zatvorené. Kladné vstupné napätie z uzla V_{VF+} je prenášané na výstupný kondenzátor C7, záporné napätie v uzle V_{VF-} nabíja kondenzátor C6, čo má za následok vznik súhlasnej jednosmernej zložky v obvode, podobne ako pri predchádzajúcich zapojeniach. Počas zápornej polvlny sú tranzistory MP4 a MN5 otvorené, zatiaľ čo tranzistory MP5 a MN4 sú zatvorené a napätie z V_{VF-} posunuté o vzniknutú jednosmernú zložku je prenášané na kondenzátor C7, zatiaľ čo záporné napätie V_{VF+} nabíja kondenzátor C5.

Hlavný rozdiel medzi takýmto mostíkom a topológiami založenými na Dicksonovom zapojení je skutočnosť že tranzistory v mostíku nefungujú ako diódy, ale správajú sa ako spínače rozložené do kríža symetricky podľa stredu. Toto je spôsobené diferenciálnym vstupným signálom privádzaným v protifáze na hradlá a emitory tranzistorov. Toto znižuje vplyv prahového napätia tranzistorov pri ich zopnutí a zároveň znižuje záverné prúdy rozopnutých tranzistorov.

Zlepšenie tohto zapojenia bolo navrhnuté v [9]. Spočíva vo využití substrátovej elektródy na zníženie hodnoty prahového napätia PMOS tranzistorov. Toto je dosiahnuté jej pripojením na zem namiesto na emitor, ako tomu bolo v [8]. Takto je

TABUĽKA I. POROVNANIE VLASTNOSTÍ OPÍSANÝCH TOPOLÓGIÍ

Práca	Topológia	Vstupný výkon	Frekvencia	Účinnosť	Záťaž
[4]	Dicksonov usmerňovač	-20 dBm	2,4 GHz	42,3 %	50 kΩ
[5]	Dicksonov usmerňovač	-10 dBm	433 MHz	23 %	10 kΩ
[7]	Dicksonov usmerňovač	-9,9 dBm	953 MHz	29 %	10 kΩ
[8]	Mostíkové zapojenie	-12,5 dBm	953 MHz	67,5 %	50 kΩ
[9]	Mostíkové zapojenie	5,2 dBm	953 MHz	69,5 %	2 kΩ

kladné generované napätie pripojené medzi emitor a substrát PMOS tranzistorov, čo má za následok zníženie ich prahového napätia a zvýšenie efektivity obvodu. Nevýhodov je ale zvýšenie parazitnej kapacity medzi emitorom a subtrátovou elektródou PMOS tranzistorov, má za následok zníženie účinnosti pre frekvencie nad 2 GHz.

Vlastnosti opísaných topológií sú uvedené v tabuľke 1. Najlepšiu účinnosť dosahujú mostíkové zapojenia v [8] a [9]. Avšak majú horšiu účinnosť, pre veľmi malé vstupné výkony, ako zapojenie v [4]. V našom výskume sa preto zameriame hlavne na možnosti zlepšenia účinnosti mostíkového zapojenia pre malé vstupné výkony.

na vysokofrekvenčnú časť systému, teda na prijímaciu anténu, obvod impedančného prispôsobenia a usmerňovač. Súčasný výskum je zameraný na analýzu a porovnanie rôznych topológií usmerňovačov. V budúcnosti budú preskúmané možnosti zlepšenia ich efektivity pre malé vstupné výkony, napríklad pomocou využitia MOS tranzistorov riadených substrátovou elektródou. Tieto je potrebné vyšetriť na výber najvhodnejšieho zapojenia a druhu tranzistora pre danú technológiu, v ktorej sa bude obvod realizovať. Overenie bude založené na simulácii a následnom meraní výrobeného testovacieho čipu. Hlavným parametrom je účinnosť konverzie výkonu, ale dôležité sú aj iné parametre ako vstupná impedancia a amplitúda výstupného jednosmerného napätia, ktorá musí byť dostatočná na správnu funkciu obvodov v jednosmernej časti systému, hlavne DC-DC meniča. Tento tvorí záťaž pre usmerňovač a preto je potrebné poznať jeho vstupnú impedanciu, ktorá vplýva na efektivitu a vstupnú impedanciu usmerňovača. Dôležitý je aj obvod na impedančné prispôsobenie antény a usmerňovača. Na jeho návrh je potrebné poznať impedancie antény i usmerňovača. Zaujímať sa budeme aj o možnosť riadenia systému zberača energie za účelom dosiahnutia maximálneho výkonu. Toto sa dá dosiahnuť buď pomocou laditeľného prispôsobovacieho obvodu [10] alebo riadením zaťažovacej impedancie usmerňovača, teda vstupnej impedancie DC-DC meniča [11].

IV. ZÁVER

V tejto práci bol predstavený systém na zber vysokofrekvenčnej energie, ktorý má slúžiť na napájanie integrovaných obvodov. Takto sa dá výrazne zvýšiť životnosť zariadení bez potreby výmeny batérie. Súčasný výskum je zameraný na výber optimálnej topológie usmerňovača na dosiahnutie čo najlepšej efektivity konverzie prijatého vysokofrekvenčného výkonu na jednosmerný výkon použiteľný ako zdroj napájania elektronického obvodu. Prezentované riešenia usmerňovačov sú založené na dvoch topológiách - Dicksonovej nábojovej pumpe a diferenciálnom mostíkovom zapojení s naprieč spínanými tranzistormi. Cieľom týchto zapojení je minimalizácia prahového napätia použitých tranzistorov, čo je hlavný zdroj strát pre malé vstupné amplitúdy vysokofrekvenčného signálu.

Budúci výskum bude zameraný na návrh vysokofrekvenčnej časti zberača energie, teda hlavne anténu, usmerňovač a obvod impedančného prispôsobenia. Budeme sa zaoberať aj parametrami DC-DC meniča, ktorý ovplyvňuje usmerňovač svojou vstupnou impedanciou. Taktiež sa pokúsime preskúmať možnosť riadenia takéhoto systému na dosiahnutie maximálnej účinnosti konverzie výkonu za pomoci ladenia obvodu impedančného prispôsobenia alebo riadenia vstupnej impedancie DC-DC meniča.

POĎAKOVANIE

Táto práca bola podporená projektami APVV-15-0254 a VEGA 1/0905/17.

REFERENCIE

[1] M. Zargham and P. G. Gulak, "Maximum Achievable Efficiency in Near-Field Coupled Power-Transfer Systems," in IEEE Transactions on Biomedical Circuits and Systems, vol. 6, no. 3, pp. 228-245, June 2012.

[2] R. Barnett, S. Lazar and Jin Liu, "Design of multistage rectifiers with low-cost impedance matching for passive RFID tags," IEEE Radio Frequency Integrated Circuits (RFIC) Symposium, 2006, San Francisco, CA, 2006.

[3] J. F. Dickson, "On-chip high-voltage generation in MNOS integrated circuits using an improved voltage multiplier technique," in IEEE Journal of Solid-State Circuits, vol. 11, no. 3, pp. 374-378, Jun 1976.

[4] Sanjeev K., M. Machnoor, K. J. Vinoy and T. V. Prabhakar, "Some practical considerations of RF to DC converter using low Vth CMOS rectifier," 2015 IEEE Applied Electromagnetics Conference (AEMC), Guwahati, 2015, pp. 1-2.

[5] S. S. Chouhan and K. Halonen, "The design and implementation of DTMOS biased all PMOS rectifier for RF energy harvesting," 2014 IEEE 12th International New Circuits and Systems Conference (NEWCAS), Trois-Rivieres, QC, 2014, pp. 444-447.

[6] T. Umeda, H. Yoshida, S. Sekine, Y. Fujita, T. Suzuki and S. Otaka, "A 950-MHz rectifier circuit for sensor network tags with 10-m distance," in IEEE Journal of Solid-State Circuits, vol. 41, no. 1, pp. 35-41, Jan. 2006.

[7] Koji Kotani and Takashi Ito, "High efficiency CMOS rectifier circuit with self-Vth-cancellation and power regulation functions for UHF RFIDs," 2007 IEEE Asian Solid-State Circuits Conference, Jeju, 2007, pp. 119-122.

[8] K. Kotani, A. Sasaki and T. Ito, "High-Efficiency Differential-Drive CMOS Rectifier for UHF RFIDs," in IEEE Journal of Solid-State Circuits, vol. 44, no. 11, pp. 3011-3018, Nov. 2009.

[9] A. K. Moghaddam, J. H. Chuah, H. Ramiah, J. Ahmadian, P. I. Mak and R. P. Martins, "A 73.9%-Efficiency CMOS Rectifier Using a Lower DC Feeding (LDCF) Self-Body-Biasing Technique for Far-Field RF Energy-Harvesting Systems," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 64, no. 4, pp. 992-1002, April 2017.

[10] K. B. de Brito and R. N. de Lima, "Impedance Network for an Automatic Impedance Matching System," 2007 Asia-Pacific Microwave Conference, Bangkok, 2007, pp. 1-4.

[11] S. Dehghani, S. Abbasian and T. Johnson, "Adjustable Load With Tracking Loop to Improve RF Rectifier Efficiency Under Variable RF Input Power Conditions," in IEEE Transactions on Microwave Theory and Techniques, vol. 64, no. 2, pp. 343-352, Feb. 2016.

Zabezpečenie vnorených systémov proti poruchám

Michal Valiček
1. ročník, denná prezenčná forma štúdia
Školitel: Tibor Krajčovič

Fakulta informatiky a informačných technológií, Slovenská technická univerzita v Bratislave
Ilkovičova 2, 812 19 Bratislava
michal.valicek@stuba.sk

***Abstrakt*—V práci sa zaoberáme detekciou porúch vo vnorených systémoch. Navrhujeme metódu založenú na dvoch prístupoch – kontrolujeme tok programu a časovanie úloh. Riešenie kladie dôraz na použitie v štandardných mikropočítačoch – COTS (Commercial off the Shelf) s minimálnou pridanou réžiou, nakoľko vnorené systémy sú spravidla limitované výkonom, prípadne spotrebou. Vzhľadom na tieto požiadavky je metóda softvérová (všeobecné použitie na COTS mikropočítačoch) a vychádzame z metód s nízkou pridanou réžiou. Na druhej strane, takéto metódy ako napríklad CFCSS (Control-flow Checking Using Branch Sequence Signatures) nedokážu detegovať časť porúch. Domnievame sa ale, že práve spojením s kontrolou časovania vieme pokryť viaceré typy porúch. Metódy sú zapuzdrené v plánovači úloh operačného systému pracujúceho v reálnom čase. Dôvodom je zdieľaný kód pre všetky úlohy v systéme a vedomosť o preempcii, ktorá je významnou prekážkou pri implementácii kontroly časovania. Plánovač má možnosť kontrolovať tok program vhodnou dekompozíciou úloh na podúlohy, ktoré predstavujú superuzly diagramu toku programu.**

Kľúčové slová—kontrola toku programu; kontrola časovania; vnorený systém, detekcia porúch

I. Úvod

Počítačové systémy zasahujú do mnohých oblastí bežného života, objavujú sa v rôznych doménach. Z pohľadu ich kritickosti môže ísť o zariadenia, ktorých výpadok nezapríčiní žiadne riziko, na druhej strane zlyhanie počítačov v zdravotníctve, energetike, vesmírnych misiách môže ohroziť životy, prípadne zapríčiniť rozsiahle škody. Zvlášť s príchodom internetu vecí sa počet zariadení dramaticky zvyšuje. Aj v prípade ak by zlyhanie zariadenia nespôsobilo priamo škody, prichádza častý problém v nárokoch na servisovanie apod. Z týchto dôvodov je často žiaduce do počítačov implementovať doplnkový softvér, hardvér na zvýšenie ich spoľahlivosti. Tieto dodatočné podsystémy predstavujú určitý nadbytok (overhead), nakoľko sa nepodieľajú na primárnej funkcionalite.

Zvýšenú spoľahlivosť vieme zabezpečiť priestorovou, časovou alebo údajovou redundanciou. Dôležitou súčasťou redundancie je detekcia porúch. Porucha môže byť detegovaná softvérovo, hardvérovo alebo hybridne, teda kombinovane. Práve pridaný a teda neštandardný hardvér môže znižovať použiteľnosť metódy, nakoľko je sťažené alebo dokonca znemožnené použitie bežne dostupných obvodov (COTS – Commercial Off the Shelf). Pochopiteľne je riešením implementácia mechanizmov prostredníctvom softvéru, čo môže viesť k neúnosnému výpočtovému, pamäťovému (údajová a aj programová pamäť) zaťaženiu. V neposlednom rade je problematické softvérovo detegovať niektoré typy porúch poruchového modelu uvedenému v druhej kapitole.

V tejto práci kombinujeme 2 rôzne metódy detekcie porúch, ktoré spoločne dokážu pokryť aj problematické typy porúch. Konkrétne pracujeme s kontrolou toku programu a doby vykonávania blokov programu. Vzhľadom na použiteľnosť výslednej metódy je našim cieľom zapuzdrenie do systému, konkrétne do plánovača. Vylepšenie práve plánovača, sa ponúka vzhľadom na to, že plánovač je zdieľaný medzi viacerými úlohami, čo znižuje pridaný kód (overhead). Ďalšou výhodou je jednoduchý prístup k informácii o poradí vykonávania úloh a ich doby vykonávania. V prípade ak plánovač nemá implementované meranie doby vykonávania úloh (to je očakávané hlavne v prípade RTOS – operačného systému pre zariadenia pracujúce v reálnom čase), je možné túto funkcionalitu doplniť (opäť jediný zdieľaný čítač pre všetky úlohy). V neposlednom rade uvažujeme nad jemnejšou granularitou sledovania toku programu a to dekompozíciou úloh na pod-úlohy. Vychádzame z opačného prístupu akým je združovanie blokov programu (súvislá postupnosť inštrukcií bez vetvenia, ktoré je povolené len na konci základného bloku, aj. basic block - BB) do superblokov prostredníctvom grafových algoritmov.

II. Motivácia

Motivácia tejto práce prichádza z dvoch oblastí. Prvou je zabezpečenie kritických systémov proti poruchám. Práve detekcia porúch je kritická v zmysle riadenia ďalších mechanizmov obnovy a zotavenia z porúch a prevencie pred chybami. Zdrojom týchto porúch je často udalosť SEU - single event upset [2]. SEU môže spôsobiť preklopenie logickej hodnoty v pamäťovej bunky, čo môže viesť k neočakávaným výsledkom. Môže dôjsť k zmene parametrov skoku, dokonca zmene inštrukcie skoku na inú a naopak. Špecifickým prípadom je aj zmena hodnoty v programovom počítadle. Práve neustála miniaturizácia prvkov na kremíku integrovaných obvodov vedie k ich nižšej odolnosti voči takýmto javom. Zdrojom nemusia byť len javy v hornej časti atmosféry, ale aj rôzne izotopy, ktoré sa nachádzajú v puzdre integrovaného obvodu.

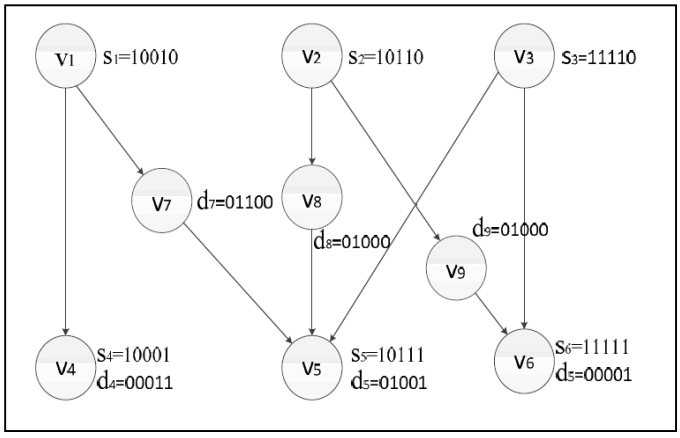
Druhým zdrojom motivácie je zabezpečenie systémov pred škodlivým kódom. Mnohé práce uvažujú dôležitý aspekt, ktorým je rovnaký dôsledok SEU a úspešného útoku. Tým dôsledkom je rozdiel v správaní sa systému oproti špecifikácii. Preto aj mechanizmy zvyšujúce odolnosť proti poruchám môžu zvyšovať odolnosť proti škodlivému kódu. Jedným zo spôsobov môže byť hľadanie charakteristických podstromov diagramu toku údajov, ktoré reprezentujú činnosť škodlivého kódu [3]. Iným prístupom je vynútenie integrity toku programu, ktoré zabráni spusteniu aj injektovaného cudzieho kódu[1][10]. Domnievame sa teda, že detekcia porúch a prevencia pred chybami má význam nielen v oblasti zabezpečenia pred poruchami typu SEU (s miniaturizáciou neustále narastajúci problém) , ale aj v oblasti bezpečnosti (narastajúci problém s príchodom internetu vecí).

III. KONTROLA TOKU PROGRAMU

Základom prakticky každej metódy na prevenciu proti chýbám, zapríčinených poruchou v kontrole programu, je jeho rozdelenie na základné bloky (BB – basic block). Základný blok programu pozostáva z maximálnej postupnosti inštrukcií, ktoré sa v prípade bezporuchovej prevádzky vždy vykonajú postupne, za sebou. Základné bloky:

- Okrem poslednej inštrukcie nesmú obsahovať inštrukcie, ktoré menia poradie vykonávania - podmienené skoky (branch).
- Nesmú obsahovať žiadnu inštrukciu, ktorá je cieľom podmieneného i nepodmieneného skoku, volania podprogramu.

Najkratším základným blokom je teda samotná inštrukcia vetvenia, v prípade nasledujúcich podmienených skokov alebo ak predchádzajúca inštrukcia je cieľom ľubovoľného skoku. Postupnosť inštrukcií v základnom bloku nemusí byť nevyhnutne bezprostredne v programe v jednej neprerušenej sekvencii. špecifickým prípadom je spojenie dvoch sekcií nepodmieneným skokom do jedného základného bloku [9].



Obr. 1. Generované príznaky metódy CFCSS.

Vzniknuté základné bloky sa použijú na zostavenie štruktúry – modelu bezporuchovej prevádzky. Časté a veľmi praktické je použitie orientovaného grafu – diagramu toku programu (CFC – Control Flow Diagram). Počas samotnej prevádzky sú generované príznaky, ktoré reprezentujú tok programu. Tieto príznaky sú porovnávané s referenčným modelom. V prípade, že dôjde k nezhode, je generovaný poruchový signál.

K veľmi efektívnej metóde kontroly toku programu patrí CFCSS (Control-flow Checking Using Branch Sequence Signatures, teda metóda kontroly toku programu používajúca príznaky sekvencií oddelených inštrukciami vetvenia) [6]. Metóda priradí unikátny identifikátor každému bloku v čase prekladania programu, v prevádzke generuje príznak, ktorý je s identifikátormi porovnávaný. Efektivitu zabezpečuje využívaním hlave XOR operácie v kontraste s delením, modulo operáciou a podobne v iných metódach. Príznaky predstavujú jednoduchú postupnosť 1,2...N, pričom N je počet základných blokov. Príklad diagramu toku programu –CFD s priradenými príznakmi CFCSS je uvedený na obrázku 1. V cieľových blokoch sa v čase prekladu vloží ešte rozdiel identifikátorov predchádzajúceho a aktuálneho bloku. Na obrázku 1 je vyobrazené počítanie príznaku z identifikátorov blokov.

Metóda je zaujímavá z pohľadu pridanej réžie. CFCSS spotrebuje od 2,17 % do 2,71 % programového priestoru. Doba vykonávania úloh sa predĺži o 6,02 % do 14,5 %. Pokrytie porúch predstavuje 87,92 %. Pokrytie je mimoriadne vysoké, aj keď hlavným dôvodom je, že SEU poruchy neboli pri testovaní autormi injektované úplne náhodne a v princípe išlo o nasledovné 3 typy:

- odstránenie vetvenia - jeho nahradenie inštrukciou NOP,
- vytvorenie vetvenia - náhodné vloženie nepodmieneného skoku do programu na ľubovoľné miesto,
- zmena operandu vetvenia - náhodná zmena v parametri vetvenia.

Metóda dokáže detegovať väčšinu porúch prvého, tretieho a štvrtého typu z nasledovného poruchového modelu:

- 1. typ - neexistujúci, nepovolený skok na začiatok základného bloku;
- 2. typ - existujúci a povolený skok na začiatok základného bloku, ktorý je ale neplatný, teda výsledkom nesprávneho vyhodnotenia vetvenia;
- 3. typ - skok z konca základného bloku uprostred iného základného bloku;
- 4. typ - skok z konca základného uprostred aktuálneho základného bloku;
- 5. typ - skok z akéhokoľvek miesta základného bloku, kamkoľvek do rovnakého základného bloku[5].

Niektoré poruchy tretieho a štvrtého typu ostanú maskované, ako príklad by bol nepovolený skok z bloku 3 na blok 9 na obrázku 1. Tento problém označili autori pojmom aliasing a ide o maskovanie porúch spôsobené nekontrolovaním výstupu zo základného bloku.

IV. KONTROLA ČASOVANIA

Kontrola doby vykonávania úloh je konceptom, ktorý sa objavuje v publikáciách už niekoľko desaťročí. Základným princípom je watchdog časovač, ktorý slúži na dohľad nad

základnou slučkou a je kompromisným riešením. Komplexnejší dohľad nad dobou vykonávania je watchdog procesor, ktorý sleduje priebeh na riadiacej, údajovej a adresnej zbernici, následne generuje v MISR (posuvný register s lineárnou spätnou väzbou) príznak, ku ktorému je priradený časový údaj získaný často statickou analýzou kódu. Práve nutnosť prístupu k zberniciam je tento spôsob v poslednej dobe implementovatelný v špeciálne navrhnutom hardvéri. Ďalším významným problémom je v prípade viac-úlohových systémov s preemciou, ťažké identifikovanie zmeny vykonávanej úlohy, ktorá sa javí v princípe ako porucha. Riešením je napríklad ovládač do multiúlohového operačného sytému, ktorý tak môže dodatočne riadiť watchdog počitač [8].

Mimo snahy pokrytia viacerých typov porúch je motiváciou použitia kontroly časovania aj dokázaná nemožnosť ochrany proti niektorým útokom len vynucovaním integrity toku údajov. Využívanie platných skokov na systémové služby s vlastnými parametrami nie je detegovateľné kontrolou toku programu, ale vedie k neúmernému predlžovaniu doby vykonávania úloh [4].

Posledným dôvodom použitia kontroly časovania je snaha vyriešiť aliasing, ktorý spôsobuje maskovanie porúch pri metóde CFCSS, ktorá ale prináša tolerovateľný overhead.

V. Ciele dizertačnej práce a tézy

Tak ako aj vyplynulo z predchádzajúcich riadkov, hlavným cieľom dizertačnej práce je navrhnutie, implementácia a overenie kombinovanej metódy zabezpečenia vnoreného systému proti poruchám. Práca v nadchádzajúcom období spočíva v nasledujúcich okruhoch, predbežných tézach dizertačného projektu:

- Navrhnutie metódy kontroly časovania v plánovači úloh OS;
- Navrhnutie metódy na dekompozíciu úlohy na plánovateľné podúlohy, teda opačný proces ako vytváranie superuzlov z uzlov CFD;
- Navrhnutie kontroly toku programu na základe dohľadu plánovača nad vykonávanými podúlohami;
- Implementácia a overenie riešenia v jednoduchom plánovači RIOS;
- Implementácia a overenie v plánovači FreeRTOS;
- Overenie pokrytia ďalších typov porúch uvedených v poruchovom modeli;
- Experimentálne určenie pridanej réžie a pokrytia pri rôznej veľkosti podúloh.

Prvé experimentálne výsledky (proof-of-concept) plánujeme získať implementáciou v jednoduchom plánovači RIOS. Reálne, aplikovateľné výsledky získame v ďalších etapách práce implementáciou metódy do plánovača FreeRTOS [7].

Poďakovanie

Táto práca bola podporená Ministerstvom školstva, vedy, výskumu a športu Slovenskej republiky, operačným programom v rámci projektu „univerzitný park STU“, ITMS 26240220084, spolufinancovaným Európskym regionálnym a rozvojovým fondom. Táto práca bola tiež podporená Slovenskou národnou výskumnou agentúrou, pod projektom VG 1/0836/16..

References

[1] Abadi, M.; Budiu, M.; Erlingsson, .; aj.: Control-Flow Integrity Principles, Implementations, and Applications. In ACM Transactions on Information and System Security (TISSEC), 2009, ISSN 1094-9224.

[2] Baumann, R.: Radiation-induced soft errors in advanced semiconductor technologies. IEEE Transactions on Device and Materials Reliability, ro4ník 5, č. 3, 2005: s. 305{316, ISSN 1530-4388.

[3] Binh, N. T.; amd H M Ngoc, Q. T. T.; Ha, N. M.: Incremental verification of OMEGA-regions on binary control flow graph for computer virus detection. In 2016 3rd National Foundation for Science and Technology Development Conference on Information and Computer Science, 2016, ISBN 978-1-5090-2098-0.

[4] Carlini, N.; Barresi, A.; Payer, M.; aj.: Control-flow bending: on the effectiveness of control-flow integrity. In SEC'15 Proceedings of the 24th USENIX Conference on Security Symposium, 2015, ISBN 978-1-931971-232.

[5] Goloubeva, O.; Rebaudengo, M, .; aj.: Software-Implemented Hardware Fault Tolerance. Springer US, 2006, ISBN 978-0-387-32937-6.

[6] Liu, L.; Ci, L.; Liu, W.: Control-flow checking using branch sequence signatures. In 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016, ISBN 978-1-5090-5880-8.

[7] Miller, B., Vahid, F., Givargis, T.: RIOS: a lightweight task scheduler for embedded systems. In Proceeding WESE '12 Proceedings of the Workshop on Embedded and Cyber-Physical Systems Education. 2012, ISBN: 978-1-4503-1765-8

[8] Vargas, F.; Piccoli, L.; Benfica, J.; aj.: Time-Sensitive Control-Flow Checking for Multitask Operating System-Based SoCs. In On-Line Testing Symposium, 2007. IOLTS 07. 13th IEEE International, 2007, ISBN 0-7695-2918-6.

[9] Yau, S. S.; Chen, F.: An Approach to Concurrent Control Flow Checking. In IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, VOL. SE-6, NO. 2., 1980, ISBN 0098-5589/80/0300-01.

[10] Zhang, C.; Wei, T.; Chen, Z.; aj.: Practical Control Flow Integrity & Randomization for Binary Executables. In 2013 IEEE Symposium on Security and Privacy, 2013, ISBN 9781509056941.

Generování testu s nulovým maskováním poruch

Robert Hülle
2. ročník prezenčního studia
Školitel: Petr Fišer, školitel specialista: Jan Schmidt

České vysoké učení technické v Praze, Fakulta informačních technologií
Thákurova 9, Praha, 16000
hullerob@fit.cvut.cz

***Abstrakt*—V tomto článku shrnuji své výsledky za 2. rok doktorského studia. Prezentuji automatický generátor testovacích vektorů (ATPG), který je schopen vygenerovat test s nulovým maskováním v daném libovolném kompaktoru: ZATPG. Diskutuji silné a slabé stránky tohoto algoritmu, včetně námětů, jak jeho slabé stránky překonat. V závěru nastiňuji další směřování výzkumu, které by mělo vést k disertační práci.**

***Klíčová slova*—test, testování, generování testu, porucha trvalá 0/1, ATPG, SAT, kompakce odezvy, maskování poruch, nulové maskování, LFSR, MISR, BIST.**

I. Úvod

V moderním světě se dennodenně setkáváme s číslicovými systémy, které nám pomáhají organizovat náš čas, poskytují nám zábavu, na nichž závisí naše životy. Není tedy divu, že testování číslicových systémů je velmi důležité. Se zvyšováním složitosti návrhu roste i náročnost generování, aplikace, a vyhodnocení testu.

Jedna z možností, jak snížit cenu aplikace testu a zvýšit pokrytí poruch, je využít prostředků pro vestavěnou diagnostiku (Built-in Self-Test, BIST).

A. Prostředky vestavěné diagnostiky

Prostředky vestavěné diagnostiky (BIST) sestávají z několika částí:

1) Generátor testovacích vektorů: jedná se o sekvenční obvod schopný generovat testovací vektory a ty posléze aplikovat na vstupy testovaného obvodu.

2) Analyzátor odezev: rozhoduje, zda odezva testovaného obvodu odpovídá teoretické odezvě bezporuchového obvodu. Tento obvod lze opět navrhnout mnoha způsoby, časté je jeho rozdělení na statický kompaktor, dynamický kompaktor a komparátor.

Statický kompaktor je kombinační obvod, který zmenšuje počet testovacích výstupních signálů, např. paritní strom.

Dynamický kompaktor je sekvenční obvod, který zpracovává sekvenci odezev testovaného obvodu na sekvenci testovacích vektorů a vytváří z nich *otisk* (signature) obvodu.

Komparátor porovnává výsledný otisk obvodu na konci testu s předpočítaným otiskem bezporuchového obvodu. Pokud se otisk liší, je obvod vyhodnocen jako poruchový.

3) BIST řadič: jedná se o obvod, který řídí běh testu. Tento obvod je zodpovědný za inicializaci generátoru testovacích vektorů i dynamického kompaktoru.

B. Maskování poruch

V případě, že testujeme obvod s poruchou, která se projeví jako chyba na výstupu obvodu, ale výsledný otisk obvodu je shodný s otiskem bezporuchového obvodu, nastalo tzv. *maskování*. Jedná se o jev nanejvýš nežádoucí, neboť snižuje celkové pokrytí poruch testem. Může nastat jak ve statickém, tak v dynamickém kompaktoru.

Ve statickém kompaktoru mohou nastat dva typy maskování. První způsobí maskování při daném testovacím vektoru, ale existuje jiný testovací vektor, který danou poruchu pokryje. Druhý typ maskování do obvodu zavádí dodatečnou redundanci, danou poruchu již nelze otestovat, stala se z ní nedetekovatelná porucha. Pro oba typy maskování existuje mnoho způsobů, jak se jim vyhnout konstrukcí kompaktoru [1]–[5].

V dynamickém kompaktoru maskování nastává, když porucha f_2, která byla detekována již dříve, vyvolá odezvu na testovací vektor pro poruchu f_1 takovou, že vnitřní stav dynamického kompaktoru se překlopí do stavu stejného, jako je stav, ve kterém by se nacházel pro neporuchový obvod. V tomto kompaktoru je předcházení maskování obtížnější, protože již nezáleží na jednom testovacím vektoru, ale na celé testovací sekvenci. I zde existuje celá řada postupů, jak maskování omezit či zcela potlačit. [6]–[11]

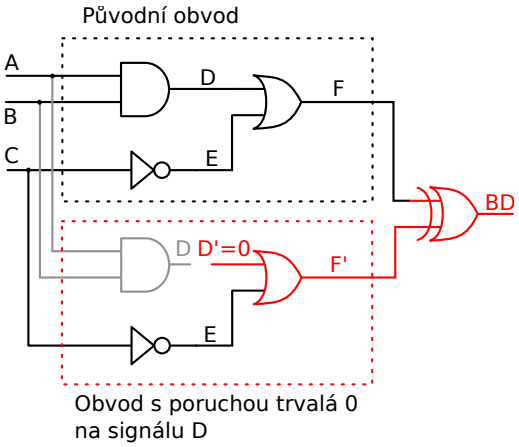
Tyto metody mají vesměs společné to, že vyžadují návrh nového či speciálního kompaktoru. Lze je jen těžko, nebo vůbec, použít na konkrétní, daný kompaktor.

Zde uvedené reference jsou staršího data, ač jsme se snažili, nenašli jsme novější relevantní práce, které neopakovaly fakta a experimenty z dřívějších prací.

C. Dosavadní výstupy

1) Rekapitulace předchozích výstupů: V předchozím ročníku semináře PAD jsem prezentoval dva články, které jsem sepsal během prvního ročníku studia [12]. První článek „SAT-ATPG for Application-Oriented FPGA Testing“ [13] jsem prezentoval na konferenci BEC 2016. Druhý článek „On Properties of ATPG SAT Instances“ jsem odeslal na konferenci DSD 2016, kde nebyl přijat.

2) SAT-based ATPG for Zero-Aliasing Compaction: Pojednává o generování testu s nulovým maskováním poruch v obecném dynamickém kompaktoru. Pro speciální případ kompaktoru, MISR založený na LFSR, experimentálně srovnává pokrytí a maskování vygenerovaného testu.



Obrázek 1: Příklad konceptuálního obvodu pro detekci poruchy trvalá 0. Tento obvod vychází z metod booleovské diference (BD).

II. Generování testu s nulovým maskováním

Jak je uvedeno výše, současné postupy potlačení maskování vyžadují manipulaci s návrhem kompaktoru. Náš přístup umožňuje potlačit maskování pro obecný kompaktor, bez zásahu do jeho architektury a bez zvětšení plochy obvodu.

Potlačení maskování realizujeme pečlivým výběrem testovacích vektorů. Protože je maskování způsobeno nešťastnou odezvou obvodu, nikoli samotným testovacím vektorem, je nutné omezit možné odezvy, tedy mít omezení na výstupech.

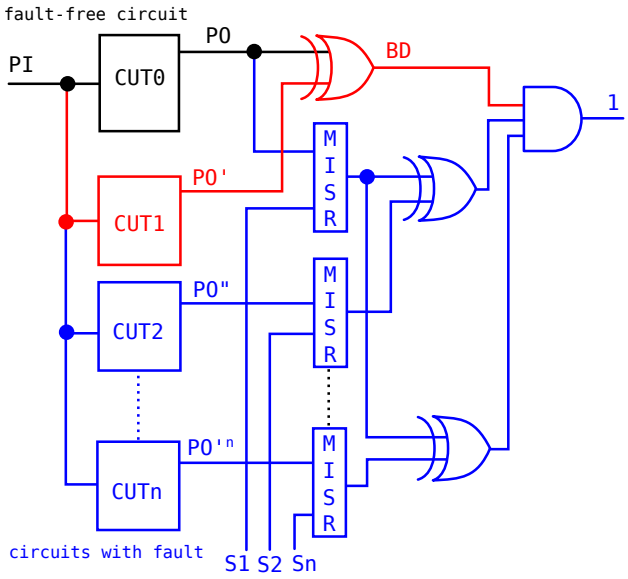
Z toho plyne nevhodnost moderních strukturních ATPG, neboť ty hledají testovací vektory od primárních vstupů (PI). Jakékoli omezení výstupů by se tedy projevilo hluboko v rozhodovacím stromě ATPG algoritmu.

Naopak ATPG založená na řešení problému Booleovské formule (SAT) tímto neduhem netrpí. Přestože SAT patří mezi NP-těžké kombinatorické problémy, moderní SAT řešiče, jako je např. MiniSAT [14], dokáží řešit i obtížné instance pocházející z ATPG procesu efektivně [15]–[17].

A. SAT ATPG

SAT ATPG převádí problém nalezení testovacího vektoru na problém splnitelnosti Booleovské formule. Jedná se o přístup podobný Booleovské diferenci, modelujeme obvod bez a s poruchou, jejich vstupy spojíme, jejich výstupy kombinujeme funkcí XOR, tím vznikne koncepční obvod, příklad je vyobrazen na obrázku 1. Porucha je detekována právě tehdy, když se odezva obvodu s poruchou liší od odezvy obvodu bez poruchy, tedy XORovaný výstup je v logické hodnotě 1 [18].

Tento koncepční obvod je poté převeden Tseitinovou transformací [19] na Booleovskou formuli v konjunktivní normální formě (CNF). Tato formule je dále zpracována SATovým řešičem, který nalezne ohodnocení odpovídající testovacímu vektoru. Takové ohodnocení existuje právě tehdy, když je porucha detekovatelná.



Obrázek 2: Konceptuální obvod pro obecný kompaktor. Bloky *MISR* jsou rozbalené kompaktory, vektory S_1–S_n představují předchozí stav kompaktoru.

B. Omezení testovacích vektorů

Na obrázku 2 je k vidění rozšířený konceptuální obvod. Obvody *CUT0* reprezentující původní testovaný obvod a *CUT1* modelující poruchu f_1, pro kterou hledáme testovací vektor. K tomuto klasickému obvodu dále přidáváme obvody *CUT2* až *CUTn*, které modelují poruchy f_2–f_n. Jedná se o poruchy, které byly otestovány dříve jiným testovacím vektorem, a pro které chceme zaručit, že nenastane maskování po aplikaci právě generovaného vektoru.

Protože maskování nastane až *po aplikaci* vektoru, který není ještě vygenerován, musíme předvídat budoucí stav kompaktoru. Toho dosáhneme rozbalením kombinační části kompaktoru a zpracováním odezvy obvodů *CUT2–CUTn* těmito kompaktory. Dále potřebujeme znát stav kompaktoru pro každou poruchu a pro bezporuchový obvod po aplikaci dosud nalezených testovacích vektorů, ty jsou reprezentovány vektory S_1–S_n. Budoucí stavy kompaktorů pro nemaskované poruchy se musí lišit od stavu bezporuchového obvodu, toto omezení je zaručeno XORováním výstupů rozbalených kompaktorů.

C. Algoritmus

Algoritmus iterativně prochází seznamem nepokrytých poruch, pro které se pokouší nalézt testovací vektor. Pro vygenerovaný vektor se provede simulace, včetně simulace kompaktoru, pro všechny poruchy. Jsou identifikovány poruchy, které byly detekovány i které byly maskovány. Pokud počet nově detekovaných poruch je větší než počet maskovaných a zároveň počet maskovaných poruch je menší než volitelný parametr M, je vektor přijat.

V případě nepřijetí testovacího vektoru jsou do procesu generování přidána omezení na nulové maskování poruch

maskovaných nepřijatým vektorem. Celý proces generování a simulace je poté opakován, dokud není nalezen vyhovující vektor, nebo již žádný vektor splňující všechna přidaná omezení neexistuje.

Celý algoritmus končí v okamžiku, kdy se pro žádnou nedetekovanou poruchu nepodaří najít vyhovující testovací vektor.

Tento algoritmus může vést k testu, který nepokrývá všechny poruchy. To ovšem nevadí ve smyslu, že přidání testovacích vektorů pro nedetekované poruchy by vedlo k maskování většímu, než je počet nově pokrytých poruch, tedy ke snížení pokrytí, pokud detekci měříme až z otisku obvodu v kompaktoru na konci testu, nikoli na testovacích výstupech obvodu.

D. Experimentální vyhodnocení

1) Popis experimentu: Tento algoritmus jsme experimentálně vyhodnotili na vybraných testovacích obvodech ze sady ISCAS'85 [20] a ITC'99 [21].

Jako dynamický kompaktor jsme použili lineární zpětnovazební registr s paralelními vstupy (MISR). Pro každý obvod jsme uvažovali několik velikostí tohoto kompaktoru, vždy jsme použili primitivní charakteristický mnohočlen. Velikost MISRu určujeme jako velikost vnitřního stavu, tedy počet bitů či počet klopných obvodů.

Použitý statický kompaktor byl paritní les, neboli několik paritních stromů (strom XOR hradel). Tyto paritní stromy mají disjunktní vstupy. Celý statický kompaktor má nulové maskování druhého typu, což znamená, že neomezuje množinu testovatelných poruch. Pro účely experimentu byl tento kompaktor součástí obvodu a testovací vektory byly generovány i pro poruchy v kompaktoru.

V experimentu porovnáváme maskování a pokrytí poruch mezi ZATPG a běžným ATPG, které při generování testu nebere v potaz kompaktor. Pro snadnější porovnání různých obvodů bereme v potaz pouze poruchy, které jsou testovatelné. To znamená, že pro každý obvod lze najít test s úplným pokrytím. Jako poruchový model jsme zvolili jednu poruchu trvalá 0/1.

Pro ZATPG i běžné ATPG používáme ATPG založené na SATu [13]. Pro řešení SATu používáme MiniSAT [14].

2) Pokrytí poruch: Jak je ilustrováno v tabulce I, pokrytí dosažené naším přístupem je pro kompaktory velmi malých rozměrů ($s < 4$) srovnatelné s běžným ATPG, ač je mírně nižší. To si vysvětlujeme ukončovací podmínkou běhu ZATPG, kdy pro některé poruchy není vygenerován testovací vektor.

Pokrytí dosažené s většími kompaktory je ovšem vyšší pro ZATPG. Zde jsou výsledky lepší, protože generování testovacích vektorů je vedeno směrem k nižšímu maskování, kdežto u běžného ATPG je maskování náhodné, dané pravděpodobností maskování v kompaktoru.

Prázdné buňky v tabulce I indikují obvody, pro které jsme neměli k dispozici statické kompaktory požadované velikosti.

3) Velikost kompaktoru s nulovým maskováním: Nulového maskování a úplného pokrytí dosáhne ZATPG pro kompaktory menších velikostí, než běžné ATPG. Tento rozdíl je k vidění v tabulce II.

E. Diskuse

Ač ZATPG dosahuje lehce nižšího pokrytí pro velmi malé kompaktory, než klasické ATPG, u větších kompaktorů naopak dosahuje lepšího pokrytí. Úplného pokrytí dosáhne u menších kompaktorů, než ATPG.

Jedním z problémů tohoto algoritmu je jeho nízká robustnost, která platí i pro běžné ATPG. Robustnost by mohlo jít zvětšit vytvořením heuristiky pro řazení poruch a pro selekci poruch bez maskování. Tyto heuristiky jsou jedním z možných budoucích vylepšení algoritmu.

III. Další směřování výzkumu a cíle disertační práce

Další směřování mého výzkumu jakož i cíle mé disertační práce vidím nadále v oblasti generování testu (ATPG) a prostředků vestavěné diagnostiky (BIST). Konkrétně se plánuji zaměřit na následující témata:

A. Omezení ATPG s ohledem na HW generátory

Přestože ZATPG je zajímavým výsledkem, má jeden problém, který jsem záměrně opomíjel. Tímto problémem je generování testovacích vektorů v BIST architektuře. Testovací posloupnost ze ZATPG lze uložit v paměti, aplikovat z ATE, nebo generovat složitým obvodem.

Jedno téma, kterým bych se tedy mohl zabývat, je další omezení ATPG takové, které by vedlo na testovací sekvenci snadno implementovatelnou v logickém obvodu.

B. Komutativní dynamické kompaktory

Velké omezení, na které jsem se ZATPG narazil, je závislost maskování na pořadí testovacích vektorů. Vzhledem k principu generování testovacích vektorů v ZATPG již s testovací posloupností nelze hýbat. To nejenže ztěžuje implementaci TPG v logickém obvodu, ale vylučuje jakékoli další zpracování testu, např. kompakci testovacích vektorů.

Za pozornost by mohlo stát využití komutativních kompaktorů, tedy takových, kde nezáleží na pořadí testovacích vektorů, resp. odezev testovaného obvodu.

C. Nelineární kompaktory

Využití nelineárních kompaktorů a prozkoumání jejich chování v součinnosti s ZATPG je taktéž možný směr výzkumu.

D. Periodické odezvy

Byly popsány metody, jak dosáhnout nulového (nebo sníženého) maskování za pomoci generování periodických odezev. Bylo by zajímavé zkusit upravit ZATPG pro generování testu s periodickou odezvou.

Tabulka I: Pokrytí poruch pro klasický ATPG a ZATPG

délka MISRu [b]		2		3		4		5		6		7		8	
obvod	poruchy	atpg [%]	zatpg [%]	atpg [%]	zatpg [%]	atpg [%]	zatpg [%]	atpg [%]	zatpg [%]	atpg [%]	zatpg [%]	atpg [%]	zatpg [%]	atpg [%]	zatpg [%]
b04	2846	78.71	79.20	91.10	87.80	94.65	95.11	97.99	97.89	98.80	99.68	99.54	99.96	99.86	99.96
b11	2382	78.30	75.99	89.37	92.02	94.58	96.09	98.02	98.86	98.78	99.71	99.41	99.96	99.87	100
c499	970	82.37	70.10	90.70	86.78	95.65	77.23	97.82	90.87	98.86	96.57	98.85	99.90	99.79	100
c880	1582	79.33	83.19	90.95	92.34	93.79	98.86	97.65	100	98.48	100	99.30	100	99.24	100
c1355	2618	79.76	75.63	90.56	86.05	95.45	92.85	97.01	97.09	99.27	100	99.65	99.73	99.77	100
c1908	2581	78.61	72.99	92.25	84.41	96.39	89.99	98.29	94.95	99.03	98.33	99.69	99.88	99.65	99.96
c2670	3613	79.91	73.04	92.69	90.78	94.54	93.85	98.11	98.45	98.61	98.00	99.83	100	99.53	100
c5315	7964			89.45	90.29	94.75	95.01	97.01	98.52	98.53	99.90	99.57	99.82	99.90	100
c7552	10921			90.62	88.42	94.85	92.97	97.55	96.35	98.28	98.94	99.32	99.95	99.65	100
průměr		79.57	75.73	90.85	88.77	94.96	92.44	97.72	97.00	98.74	99.01	99.46	99.91	99.70	99.99

Tabulka II: Nejmenší kompaktor s nulovým maskováním

		délka MISRu [b]		pokrytí [%]	
obvod	poruchy	ATPG	ZATPG	ATPG	ZATPG
b04	2846	11	9	100	100
b11	2788	10	8	100	100
c499	970	12	8	100	100
c880	1582	12	5	100	100
c1355	2618	10	6	100	100
c1908	2581	12	9	100	100
c2670	3613	12	7	100	100
c5315	7964	14	8	100	100
c7522	10921	12	8	100	100

E. Shrnutí

Jako jeden cíl své disertační práce si stanovuji rozšířit algoritmus popsaný v tomto článku o schopnost generovat test, který bude snadno implementovatelný v HW generátoru testovacích vektorů pro obvody BIST (bod III-A).

Další cíl, který si stanovuji je analýza představeného algoritmu při použití s jinými typy kompaktorů. Případně návrh nových kompaktorů, které by mohly lépe využít vlastností ZATPG (body III-B – III-D).

PODĚKOVÁNÍ

Autor děkuje za poskytnuté výpočetní a úložné zdroje programu Metacentra CESNET LM2015042 a CERIT-CS CZ.1.05/3.2.00/08.0144.

Tento výzkum byl částečně podporován z projektu ČVUT SGS17/213/OHK3/3T/18.

Tento výzkum byl částečně podporován z grantu české grantové agentury GA16-05179S.

LITERATURA

[1] K. Chakrabarty, "Zero-aliasing space compaction using linear compactors with bounded overhead," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 5, pp. 452–457, 08 2002.

[2] S. R. Das, M. Sudarma, M. H. Assaf, E. M. Petriu, W. B. Jone, K. Chakrabarty, and M. Sahinoglu, "Parity bit signature in response data compaction and built-in self-testing of VLSI circuits with nonexhaustive test sets," *IEEE Transactions on Instrumentation and Measurement*, vol. 52, no. 5, pp. 1363–1380, Oct 2003.

[3] K. Chakrabarty and J. P. Hayes, "Test response compaction using multiplexed parity trees," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 15, no. 11, pp. 1399–1408, Nov 1996.

[4] B. Pouya and A. Touba, Nur, "Synthesis of Zero-Aliasing Elementary-Tree Space Compactors," in *IEEE VLSI Test Symposium*, 1998, pp. 70–77.

[5] Y. Liu and A. Cui, "An Efficient Zero-Aliasing Space Compactor Based on Elementary Gates Combined with XOR Gates," in *IEEE/ACM International Conference on Computer-Aided Design*, 11 2013, pp. 95–100.

[6] K. Pradhan, D., M. Reddy, Sudhakar, and K. Gupta, Sandeep, "Zero aliasing compression," in *Fault-Tolerant Computing: 20th International Symposium*, 06 1990, pp. 254–263.

[7] G. Edirisooriya and P. Robinson, John, "Test Generation to Minimize Error Masking," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, no. 4, pp. 540–549, 04 1993.

[8] T. Bogue, M. Gossel, H. Jurgensen, and Y. Zorian, "Built-in self-test with an alternating output," in *Proceedings Design, Automation and Test in Europe*, Feb 1998, pp. 180–184.

[9] G. Edirisooriya, P. Robinson, John, and S. Edirisooriya, "On the performance of augmented signature testing," in *IEEE International Symposium on Circuits and Systems*, 05 1993, pp. 1607–1610.

[10] K. Pradhan, D. and K. Gupta, Sandeep, "A new framework for designing and analyzing BIST techniques and zero aliasing compression," *IEEE Transactions on Computers*, vol. 40, no. 6, pp. 743–763, 1991.

[11] M. Kopec, "Can nonlinear compactors be better than linear ones?" *IEEE Transactions on Computers*, vol. 44, no. 11, pp. 1275–1282, Nov 1995.

[12] R. Hülle, "Generování testu pro prostředky vestavěné diagnostiky," in *14. Pracovní seminář Počítačové architektury a diagnostika*, 9 2016.

[13] R. Hülle, P. Fišer, J. Schmidt, and J. Borecký, "SAT-ATPG for Application-Oriented FPGA Testing," in *15th Biennial Baltic Electronics Conference*, 10 2016, pp. 83–86.

[14] N. Eén and N. Sörensson, "An Extensible SAT-solver," in *Theory and Applications of Satisfiability Testing*, ser. Lecture Notes in Computer Science, E. Giunchiglia and A. Tacchella, Eds. Springer Berlin Heidelberg, 2004, vol. 2919, pp. 502–518.

[15] M. Prasad, P. Chong, and K. Keutzer, "Why is ATPG easy?" in *36th Design Automation Conference*, 1999, pp. 22–28.

[16] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, "2+p-SAT: relation of typical-case complexity to the nature of the phase transition," *Random Structures & Algorithms*, vol. 15, no. 3-4, pp. 414–435, 1999.

[17] S. Eggersglüß, R. Krenz-Baath, A. Glowatz, F. Hapke, and R. Drechsler, "A new SAT-based ATPG for generating highly compacted test sets," in *15th IEEE Design and Diagnostics of Electronic Circuits and Systems*, April 2012, pp. 230–235.

[18] T. Larrabee, "Test pattern generation using Boolean satisfiability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 11, no. 1, pp. 4–15, Jan. 1992.

[19] G. Tseitin, "On the Complexity of Derivation in Propositional Calculus," in *Automation of Reasoning*, ser. Symbolic Computation, J. Siekmann and G. Wrightson, Eds. Springer Berlin Heidelberg, 1983, pp. 466–483.

[20] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," in *IEEE International Symposium Circuits and Systems (ISCAS'85)*. IEEE Press, Piscataway, N.J., 1985, pp. 677–692.

[21] F. Corno, M. S. Reorda, and G. Squillero, "RT-level ITC'99 benchmarks and rst ATPG results," in *IEEE Design Test of Computers*, Jul 2000, pp. 44–53 vol.17.

Komunikačný modul pre implantovateľné senzorické systémy

Ing. Šimon Danko
2. ročník, denná prezenčná forma
prof. Ing. Viera Stopjaková, PhD.

Fakulta elektrotechniky a informatiky, STU
Ilkovičova 3, Bratislava
simon.danko@stuba.sk

***Abstrakt*—Tento príspevok prezentuje výsledky doterajšieho výskumu v rámci dizertačnej práce zameranej na bezdrôtový komunikačný modul (s dôrazom na jeho spotrebu) určený pre implantovateľné zariadenia. Tento systém bude pracovať v tzv. Ultra-wideband pásme, čo prináša zníženie energie potrebnej na prenesenie jedného bitu oproti štandardným úzkopásmovým typom bezdrôtovej komunikácie, a taktiež značné zjednodušenie vysielača. Na druhej strane to však prináša nové výzvy pri návrhu vhodného prijímača. V tomto článku je prezentovaný návrh prijímača pre Ultra-wideband, ktorý je určený pre implantovateľné senzorické zariadenia slúžiace na dlhodobé monitorovanie zdravotného stavu.**

Keywords—komunikačný modul; ultra-wideband; aktívne bio-senzorické implantáty; spotreba energie

I. Úvod a Motivácia

Súčasné riešenia dlhodobého monitorovania zdravotného stavu pacienta sú založené na pravidelných prehliadkach a vyšetreniach. Aktívne bio-senzorické implantáty (ABSI) ponúkajú možnosť sledovať stav pacienta kontinuálne a dlhodobo, bez potreby externého zariadenia. Implantát v tele neobmedzuje pacienta z hľadiska pohybu a bežného každodenného života, jeho zavedenie do tela je však invazívne. Namerané údaje sú vyhodnotené a pravidelne odosielané do zberného bodu (tzv. HUB-u), ktorým môže byť domáci počítač prípadne mobilný telefón. Tieto dáta sa následne môžu odosielať lekárovi a informovať o prípadnom zhoršení zdravotného stavu.

Moderné aktívne elektronické implantáty sú komplexné zariadenia, ktoré pre svoju správnu funkciu vyžadujú počiatočnú konfiguráciu a monitorovanie správnej funkčnosti počas celej životnosti implantátu. Počiatočná konfigurácia sa vykoná buď priamo na operačnej sále pri prvotnom zavedení zariadenia do tela (napr. kardiostimulátory) alebo až po zahojení miesta zákroku (napr. kochleárne implantáty). Štandardný spôsob komunikácie je založený na magnetickej indukčnosti (RFID), kedy sa vonkajšia časť systému musí nachádzať v tesnej blízkosti implantátu. Najčastejšie priamo na koži nad miestom implantovania. Postupné pridávanie funkcií implantátov vyžaduje čoraz vyššie prenosové rýchlosti a mobilitu, bez nutnosti priameho kontaktu s pacientom.

Alternatívou k predchádzajúcemu spôsobu sa stal rádio-komunikačný štandard Medical Implant Communication Service (MICS / MEDS) z roku 1999, ktorý je špeciálne určený pre komunikáciu s medicínskymi zariadeniami pracujúcimi vo vnútri alebo v blízkosti ľudského tela ako napr. defibrilátory, neurostimulátory, inzulínové pumpy. Tieto štandardy sú však primárne určené len na telemetriu implantátu a nie kontinuálny zber dát [1].

II. Ultra-Wideband Technológia

Ultra-wideband (UWB) je komunikačný štandard pre bezdrôtové systémy pracujúce v pásme od 3,1 GHz do 8,5 GHz so šírkou pásma minimálne 50 MHz v Európe a 500 MHz vo zvyšku sveta (alebo ±20% relatívne z frekvencie nosnej vlny) a maximálnou spektrálnou hustotou vyžarovaného výkonu na úrovni –41,3 dBm/MHz (74,13 nW/MHz). UWB umožňuje komunikovať vysokou prenosovou rýchlosťou za použitia minimálneho množstva energie, v rádoch pJ až nJ na pulz. Pre porovnanie, Bluetooth LE vyžaduje pri 960 b/s 153 nJ/bit [2]. Podľa spôsobu vysielania môžeme UWB systémy rozdeliť na dve základné kategórie.

A. FM-UWB

Frekvenčne modulované UWB (FM-UWB) využíva dvojitú FM moduláciu. V prvom kroku sa na dátový signál aplikuje digitálna FSK modulácia s nízkym modulačným indexom. Tým vznikne tzv. podnosná vlna (sub-carrier). Na ňu sa následne aplikuje analógová FM modulácia s vysokým modulačným indexom. Tým vznikne RF signál s konštantnou UWB obálkou. Spotreba FM-UWB vysielačov je v rádoch desiatok nJ/bit [3].

Pre demoduláciu FM-UWB je možné použiť buď auto-koreláciu alebo hetero-dynový prijímač. V prvom prípade sa prijímaný signál násobí so svojim obrazom oneskoreným o čas τ. Takýto prístup si nevyžaduje RF lokálny oscilátor (LO) na strane prijímača, čo značne znižuje jeho spotrebu. Avšak návrh vhodného oneskorovacieho člena je komplikovaný. Ten je možné implementovať ako oneskorovacie vedenie vytvorené paralelným rezonančným obvodom, all-pass filter (APF) alebo pásmový priepust. Tieto riešenia si však vyžadujú niekoľko cievok, ktoré zaberajú značnú plochu na čipe.

B. IR-UWB

Impulse radio UWB (IR-UWB) využíva krátke elektromagnetické impulzy bez nosnej vlny. Tvar impulzu nie je definovaný, normou je definovaná iba spektrálna maska. V prípade Gaussových impulzov sa veľká časť energie nachádza

v spektre nízkych frekvencií, avšak pre dodržanie normy je nutná filtrácia týchto zložiek, čo vedie k zníženiu účinnosti takéhoto systému. Určitou výhodou môže byť fakt, že vhodne navrhnutá anténa pracuje ako filter, ale návrh UWB antény je značne komplikovaný a je samostatnou časťou výskumu. Riešením problému filtrovania sú tvarovacie obvody, ktoré dokážu vygenerovať impulz s presne danou spektrálnou charakteristikou, ktorá umožňuje zúžiť spektrum a tým využívať viacero kanálov súčasne. Tento prístup však prináša zvýšenie zložitosti obvodu a veľmi vysoké nároky na časovanie jednotlivých blokov [4].

IR-UWB impulzy majú niekoľko násobne širšie spektrum, čo komplikuje ich príjem. Koherentné IR-UWB prijímače sú založené na detekcii energie impulzu alebo porovnaním prijatého signálu s vzorom impulzu uloženým v prijímači [5]. Aj keď takýto prístup s môže javiť ako energeticky výhodnejší, je potrebné si uvedomiť, že takéto prijímače vyžadujú digitálnu logiku na spracovanie a dekódovanie, prípadne vytvorenie vzoru impulzu. Vzor impulzu sa vytvára na základe známeho modelu komunikačného kanála. Ďalším typom prijímačov sú tzv. nekoherentné prijímače, najčastejšie pracujúce na princípe detekcie nábežnej hrany pulzu, prenášanej referencie (*transmitted reference*) alebo diferenciálneho príjmu. Absencia nosnej vlny limituje počet dostupných typov modulácie signálu a taktiež činí takého prijímače náchylné na detekciu falošných impulzov, vytvorených rušením z iných zdrojov.

Tab. 1 porovnáva FM-UWB a IR-UWB systémy z hľadiska niektorých kľúčových aspektov pre návrh systému.

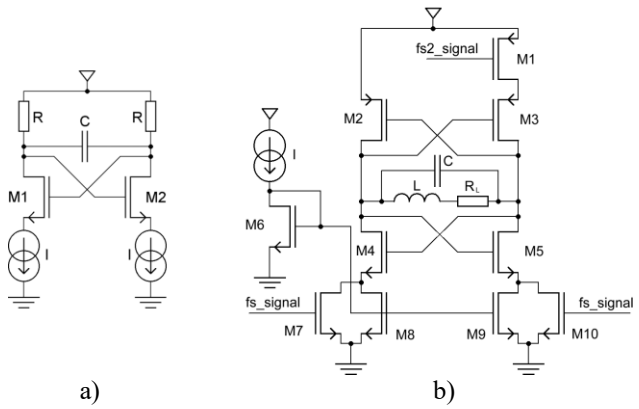
Tab.1: Porovnanie FM-UWB a IR-UWB.

Vlastnosti	*FM-UWB*	*IR-UWB*
Energetická účinnosť		✓
Možnosť ohraničiť spektrum (kanály)	✓	
Širka spektra impulzu		✓
Menšia zložitosť vysielača		✓
Menšia zložitosť prijímača	✓	
Menšia zložitosť návrhu antény	✓	
Prenosová rýchlosť		✓

Kompromisom medzi FM a IR-UWB hlavne z pohľadu spotreby a zložitosti návrhu obvodov sú tzv. tónové impulzy. Sú to krátke impulzy určitej frekvencie, ktorých obálka má tvar Gaussovej funkcie. Tieto tónové impulzy, trvajúce niekoľko nanosekúnd, majú veľkú šírku pásma zo stredom v danej frekvencii, čo umožňuje rozdeliť dostupné spektrum na viacero kanálov a zároveň ohraničením spektra zjednodušiť konštrukciu prijímača.

III. Vysielač

Z pohľadu spotreby energie bezdrôtového komunikačného systému sú najkritickejšie jeho vysokofrekvenčné časti. Spotreba obvodu stúpa úmerne s frekvenciou, a okrem toho, pri vysokých frekvenciách zohrávajú dôležitú úlohu rôzne parazitné vlastnosti komponentov, napr. kapacita hradlového oxidu, kapacita kanála voči substrátovej elektróde, tranzitná frekvencia tranzistorov (f_T), ale aj samotná topológia obvodu (layout).



Obr. 1: a) Schéma jednoduchého relaxačného oscilátora, b) schéma navrhovaného relaxačného oscilátora.

Kryštálové oscilátory poskytujú veľmi dobrú frekvenčnú, fázovú a tepelnú stabilitu spolu s nízkou spotrebou. Avšak ich integrácia na čip je značne komplikovaná a vyžaduje si špeciálny výrobný proces [6]. Navyše majú dlhý čas rozbehu, až niekoľko tisíc períod. Pre integrované obvodov (IO) je možné použiť napr. ring oscilátor alebo relaxačný oscilátor, ale frekvencia ich oscilácií je silne závislá od PVT (*process, volage, temperature*) fluktuácií. Pre bezdrôtový komunikačný systém, ktorý dokáže pracovať na viacerých kanáloch, je dôležitá schopnosť LO meniť frekvenciu oscilácií v určitom rozsahu. Oscilátory sa tradične navrhovali v bipolárnej alebo v BiCMOS technológii, ktoré sú rýchlejšie a vykazujú nižší šum, ale sú drahšie ako CMOS. Zo zvyšujúcou sa maximálnou pracovnou frekvenciou CMOS technológie sa ich výhoda postupne stráca.

Relaxačný oscilátor v CMOS technológii, ktorého schéma je znázornená na obr. 1a, sa v súčasnosti javí ako najvhodnejšie riešenie pre UWB systémy (ak je tepelne kompenzovaný) [7]. Frekvencia oscilácií jednoduchého relaxačného oscilátora (obr. 1a) je daná vzťahom $f_{osc} \approx 1/8RC$. V tomto prípade je možné ovplyvňovať frekvenciu aj prúdom I a preto je toto zapojenie veľmi citlivé na zmeny teploty. Je dôležité pripomenúť, že pre frekvencie v rádoch GHz, je nutné použiť veľmi malé hodnoty R a C, približne 100 Ω a 100 fF. Takto nízke hodnoty C sa v praxi blížia parazitným kapacitám prepojov na čipe, ktoré okrem samotnej topológie obvodu závisia aj od parametrov výrobného procesu, čo môže viesť k nežiadúcim zmenám f_{osc}.

Zmenou RC za LC člen v komplementárnom zapojení (obr. 1b), získame obvod s frekvenciu danou (1), zároveň pre g_m tranzistorov musí platiť (2) [7].

$$f_{osc} = \frac{1}{2\pi\sqrt{LC}} \tag{1}$$

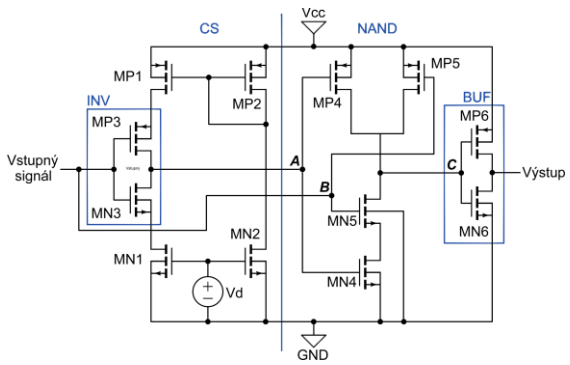
$$g_m \geq \frac{LC}{R_L} \tag{2}$$

Ak nahradíme fixnú kapacitu C poľom spínaných kondenzátorov, je možné vytvoriť digitálne laditeľný oscilátor (DCO). Malé zmeny frekvencie je taktiež možné dosiahnuť zmenou prúdu cez oscilátor, ale za cenu zmeny amplitúdy

výstupného signálu. Ďalšou nespornou výhodou relaxačného oscilátora je jeho diferenciálny výstup, ktorý je vhodné použiť v nízkonapäťových IO.

Zníženie spotreby energie LO je možné dosiahnuť vypínaním LO v čase kedy nie je potrebný, prípadne medzi jednotlivými impulzmi. To si vyžaduje krátky čas nábehu LO, ktorý je nepriamo úmerný energii dodávanej do systému. Obr. 1b zobrazuje schému relaxačného oscilátora s modifikáciou pre skrátenie času potrebného pre ustálenie amplitúdy oscilácií. Krátke zopnutie tranzistorov M7 a M10, riadených signálom „*fs_signal*“, skratuje prúdové zrkadlo tvorené tranzistormi M6, M8 a M9, čo umožňuje zvýšenie prúdu tečúceho oscilátorom. To vedie k rýchlejšiemu nábehu oscilácií. Pre ďalšie zníženie tohto času sa signálom „*fs2_signal*“ na krátku dobu uzavrie tranzistor M1, čo krátkodobo naruší pomery v jednotlivých vetvách oscilátora. Alternatívnym prístupom je možné obmedziť prúd oscilátorom tesne pod hraničnú hodnotu, pri ktorej je oscilátor schopný pracovať a následne dátovým signálom zvyšovať prúd nad túto hodnotu. Keďže oscilátor sa nerozbieha z nulových počiatočných podmienok, čas nábehu oscilácií sa skráti. Týmto spôsobom je možné generovať priamo oscilátorom už spomínané tónové impulzy.

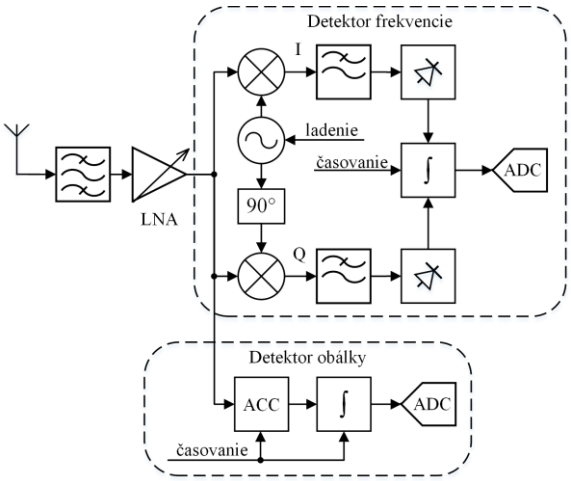
Obvod generovania Gaussových impulzov je znázornený na obr. 2. Pracuje na princípe detektora nábežnej hrany s prúdovo-obmedzeným invertorom. Prúdové zrkadlo CS (tranzistory MP1 a MP2) obmedzuje maximálny nabíjací prúd hradlových kapacít tranzistorov MP4 a MN4 cez invertor INV. Tým sa vytvára oneskorenie medzi signálmi A a B, vstupujúcimi do NAND hradla. Vďaka oneskoreniu tak vznikne na krátky okamih na výstupe NAND-u log.0. Dĺžku vygenerovaného impulzu je možné meniť napätím V_d od 0,4 ns do 4 ns. Posledný invertor slúži ako budič nasledujúceho stupňa. Vlastná spotreba obvodu je približne 150 fJ/pulz pri opakovaní s frekvenciou 1 MHz. Obvod bol simulovaný v 90 nm CMOS technológii.



Obr. 2: a) Schéma generátora Gaussových impulzov.

IV. Prijímač

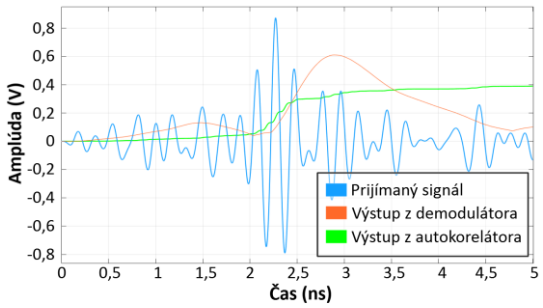
Generovanie presnej a stabilnej frekvencie je v súčasnosti stále komplikované a je mu venovaný rozsiahly výskum. Pretože súčasné úzkopásmové (NB – *narrow band*) systémy pracujú s veľmi malou šírkou pásma, je pre správnu demoduláciu veľmi dôležitá synchronizácia frekvencie a fázy LO prijímača a vysielača, čo je možné dosiahnuť použitím napr. fázového závesu, ktorý sa „zamkne“ na nosnú vlnu. Samotný fázový záves potrebuje pre synchronizáciu LO s prijímaným signálom určitý čas, ktorý je rádovo väčší ako čas trvania UWB impulzu. Použitím IQ demodulátora môžeme odstrániť nutnosť presnej synchronizácie LO prijímača a vysielača, avšak za cenu použitia dvoch, prípadne viacerých zmiešavačov.



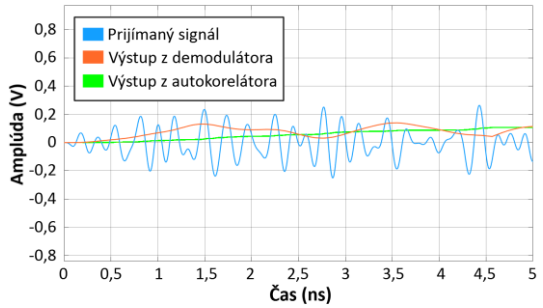
Obr. 3: Bloková schéma navrhovaného prijímača UWB.

Navrhovaný komunikačný modul bude využívať OOK moduláciu tónových impulzov. Takýto spôsob sa radí niekde medzi IR-UWB a FM-UWB a je veľmi podobný DS-UWB (*direct sequence*). Výhodami sú zúženie spektra, ktoré nepriamo súvisí s dĺžkou impulzu a jeho posunutie na určitú nosnú frekvenciu (*tón*), ktoré umožňuje využívať viacero kanálov v dostupnom spektre. Bloková schéma navrhovaného prijímača je zobrazená na obr. 3.

Okrem LNA je najkritickejšou časťou prijímača zmiešavač, ktorého návrh je z dôvodu vysokých frekvencií komplikovaný. Veľmi široké spektrum UWB signálov uvoľňuje požiadavku na presnú synchronizáciu prijímača a vysielača. V praxi to znamená, že ak frekvencie LO a prijímaného signálu nie sú rovnaké, ale sú dostatočne blízko seba a ich rozdiel získaný zmiešaním spadá do pásma priepuste filtra za zmiešavačom, tento signál je možne demodulovať. Takýto prístup sa nazýva približná medzifrekvencia (*Approximate Intermediate Frequency* – AIF) [8]. Grafy na obr. 4 a 5 zobrazujú časové priebehy signálu pre bity s hodnotou *1* a *0*. Súčasná detekcia obálky a nosnej frekvencie impulzu zvyšuje odolnosť voči rušeniu ako aj pravdepodobnosť správnej detekcie signálu. Prijatý impulz musí obsahovať danú frekvenciu a zároveň správne množstvo energie, ktoré sa získava auto-koreláciou prijatého signálu. Prijímač bol simulovaný v prostredí Simulink a dosiahol BER = 10^{-4} (*Bit Error Ratio – pomer chybných bitov k celkovému počtu prijatých*) v AWGN kanále (*Additive White Gaussian Noise – aditívny biely Gaussov šum, ktorého spektrálna hustota výkonu je konštantná*) s SNR = 3 dB. Finálna demodulácia signálu bola implementovaná metódou *hard-decision*, teda porovnaním výstupov z detekcie obálky a nosnej frekvencie s pevne nastavenou prahovou hodnotou.



Obr. 4: Graf vstupného RF signálu zodpovedajúcemu bitu „1“ výstupných signálov zo simulovaného prijímača pre SNR = 6 dB.



Obr. 5: Graf vstupného RF signálu zodpovedajúcemu bitu „0“ a výstupných signálov zo simulovaného prijímača pre SNR = 6 dB.

Tab. 2: Porovnanie BER voči SNR pre UWB prijímače.

SNR [dB]	Rýchlosť	BER	Spotreba	Referencia
12	100 kbps	10^{-3}	22 nJ/bit	[9]
8	750 kbps	10^{-3}	6.7 nJ/bit	[3]
3	1 Mbps	10^{-4}	N/A	Táto práca

Z tab. 2 je možne pozorovať, že navrhovaný prijímač má vyššiu pravdepodobnosť správneho príjmu pri menšom SNR. Táto vlastnosť je výhodou pre implantáty pretože tkanivo okolo nich má vysoký útlm v UWB pásme. Tento útlm je spôsobený jednak vysokým obsahom vody, ale aj kompozíciou tela skladajúcou sa z rôznych vrstiev, ako napr. sval, tuk a koža. Tieto materiály majú rôzne elektrické vlastnosti a na ich rozhraní vznikajú odrazy. Útlm tkaniva môže dosiahnuť až 80 dB pri hĺbke implantácie 120 mm. Aj napriek tomuto faktu práca [10] dokazuje, že je prenos UWB cez tkanivo je možný. Nakoľko simulácia bola vykonaná za použitia všeobecných blokov vyššej úrovne, informácia o spotrebe v tejto fáze návrhu nie je dostupná a bude určená samotným obvodovým riešením jednotlivých blokov, technológiou a topografiou čipu.

V. Bezpečnosť komunikácie

Nakoľko informácie o zdravotnom stave pacienta sú považované za dôverné, ich bezpečnosť je prvoradá. Nielen prenášané dáta, ale aj samotný elektronický implantát a jeho použitie a nastavenie musia byť zabezpečené proti útokom tak, aby sa predišlo možnému zneužitiu a ohrozeniu zdravia alebo života pacienta. Vlastnosti samotnej UWB komunikácie poskytujú taktiež istú mieru zabezpečenia. Veľmi nízka spektrálna hustota energie a krátky čas trvania impulzu, bez kontinuálnej prítomnosti nosnej vlny činia takúto komunikáciu veľmi ťažko zistiteľnou a odpočúvateľnou. Ďalšie zvýšenie bezpečnosti sa dá dosiahnuť preskakovaním frekvencií a časových slotov vo vopred dohodnutej pseudo-náhodnej sekvencii, čo je možné uskutočniť bez nutnosti úpravy zapojení vysielača a navrhovaného prijímača. Konečná implementácia zabezpečenia závisí od požiadaviek konkrétnej aplikácie a je mimo rámca tejto práce.

VI. Záver

UWB komunikácia bola zvolená pretože umožňuje znížiť jednak energiu potrebnú na prenesenie informácie, tak aj spotrebu obvodových častí modulu. Aj keď exituje niekoľko konštrukcií UWB vysielačov, ktoré je možné použiť v ABSI, príjem signálov v reálnom prostredí je stále problematický. Navrhovaný prijímač prijíma UWB signál na základe kombinácie detekcie obálky energie impulzu a hetero-dynového príjmu nosnej frekvencie. Tento prístup znižuje BER a zároveň zvyšuje odolnosť voči vonkajšiemu rušeniu. Schopnosť spoľahlivo prijímať signál s malým SNR je pre implantovateľné systémy dôležitá, nakoľko živé tkanivo podstatne utlmuje elektromagnetické žiarenie v UWB pásme.

Ďalší výskum bude zameraný na návrh topografie jednotlivých obvodových blokov UWB vysielača a prijímača, implementáciu na čip a verifikáciu ich funkčnosti.

Poďakovanie

Táto práca bola podporená projektami APVV-15-0254 a VEGA 1/0905/17.

Referencie

[1] European Telecommunications Standards Institute, „Short Range Devices (SRD);Medical Body Area Network Systems (MBANSs) operating in the 2 483,5 MHz to 2 500 MHz range; Harmonised Standard covering the essential requirements of article 3.2 of the Directive 2014/53/EU”

[2] Y. Zhu, J. Hu a H. Wu, „A 2.5 Gpulse/s, 25 pJ/pulse, 0.18um CMOS Impulse Radio UWB Transmitter Based on Dual-Polarity Distributed Waveform Generator“.

[3] W. R. Z. W. Fei Chen, „A 5 mW 750 kb/s Noninvasive Transceiver for,“ *IEEE Transactions on Circuits and Systems II: Express Briefs,* %1. vyd.99, 2017.

[4] A. P. C. David D. Wentzloff, „A 47pJ/pulse 3.1-to-5GHz All-Digital UWB Transmitter in 90nm CMOS,“ rev. *IEEE International Solid-State Circuits Conference*, 2007.

[5] B. Latré, B. Braem, I. Moerman, C. Blondia a P. Demeester, „A survey on wireless body area networks,“ *Wirel. Netw.,* zv. 17, pp. 1-18, 2011.

[6] N. A. M. B. I. R. T. I. B. Mohammad Marufuzzan, „Design of Low Power Crystal Oscillator in 0.13 um CMOS Technology,“ rev. *2016 International Conference on Advances in Electrical, Electronic and Systems Engineering (ICAEES)*, Putrajaya, Malaysia , 2016.

[7] N. Maheshwari, „Analysis and characterization of VCO for UWB Application,“ *International Journal of Latest Trends in Engineering and Technology (IJLTET),* zv. 2, %1. vyd.1, pp. 226-233, 2013.

[8] D. B. C. C. E. Vladimir Kopta, „A 420 µW, 4 GHz Approximate Zero IF FM-UWB Receiver for Short-Range Communications,“ rev. *IEEE Radio Frequency Integrated Circuits Symposium*, 2016.

[9] J. R. L. N. Saputra, „A Short-Range Low Data-Rate Regenerative FM-UWB Receiver,“ *IEEE Transactions on Microwave Theory and Techniques,* zv. 59, %1. vyd.4, pp. 1131-1140, 2011.

[10] D. Anzai, K. Katsu, R. Chavez-Santiago, Q. Wang, D. Plettemeier a J. Wang, „Experimental evaluation of implant UWB-IR transmission with living animal for body area networks,“ *IEEE TRANSACTIONS ON MICROWAVE THEORY AND TECHNIQUES,* zv. 62, pp. 183-192, 2013.

Číslicový návrh spojující odolnost proti poruchám a odolnost proti útokům

Vojtěch Miškovský
2. ročník, prezenční studium
Školitel: Hana Kubátová, Školitel specialista: Martin Novotný

České vysoké učení technické v Praze, Fakulta informačních technologií
Thákurova 9, 160 00 Praha 6
miskovoj@fit.cvut.cz

***Abstrakt*—Odolnost proti poruchám a odolnost proti útokům jsou návrhové vlastnosti, které mohou být u některých zařízení vyžadovány současně. Pro obě tyto vlastnosti existují návrhové metody, které ovšem vyžadují poměrně velkou režii plochy či spotřeby. Vzhledem k této režii by se mohlo stát, že návrh odolný proti poruchám sníží odolnost proti útokům nebo naopak návrh odolný proti útokům sníží odolnost proti poruchám. Cílem našeho výzkumu je prozkoumat tyto vzájemné vlivy a navrhnout nové metody spojující odolnost proti poruchám a odolnost proti útokům.**

***Klíčová slova*—odolnost proti útokům, odolnost proti poruchám, FPGA, DPA, AES, spolehlivost, bezpečnost**

I. Úvod

Moderní elektronické systémy se čím dál více stávají součástí běžného života a nasazují se i v oblastech, kde na jejich správné funkčnosti může záviset lidské zdraví i majetek. Takové systémy pak kladou velké nároky na spolehlivost. Musí odolávat vlivům stárnutí, nevhodnému zacházení, ale také cíleným útokům. Proti těmto vlivům je potřeba citlivá zařízení chránit.

A. Motivace

Každé digitální zařízení může postihnout porucha. Ta může vést k chybě a případně k selhání funkce zařízení. Chceme-li takovému selhání zabránit, musíme zařízení navrhnout odolné proti poruchám. Metody číslicového návrhu odolného proti poruchám jsou široce zkoumané téma a existuje mnoho metod, jak zařízení ochránit. Tyto metody jsou obvykle založené na využití redundance [1]. Nevýhodou takového řešení je režie plochy, času či spotřeby, kterou návrh odolný proti poruchám přináší.

Dalším problémem může být útočník snažící se narušit chod zařízení či získat citlivá data. Číslicovým návrhem můžeme ovlivnit pouze takové útoky, které jsou založené na fyzické implementaci zařízení. Mezi takové útoky patří například útoky postranními kanály (side-channel attacks) [2] nebo útoky injekcí poruchy (fault injection attacks) [3]. Existují návrhové metody, pomocí kterých se lze těmto útokům bránit. Jejich nevýhodou je opět režie.

Pokud navrhujeme zařízení tak, aby bylo odolné proti poruchám i proti útokům, můžeme se setkat s různými komplikacemi.

B. Definice problému

Odolnost proti poruchám i odolnost proti útokům jsou hojně zkoumaná témata, nicméně jsme se nesetkali s výzkumem, který by se věnoval kombinaci těchto vlastností a jejich vzájemnému ovlivňování.

Problémem může být například redundance využívaná v návrzích odolných proti poruchám. Tato redundance by mohla mít negativní vliv na odolnost proti útokům postranními kanály, neboť může zvýšit množství informace, které se v postranních kanálech objevuje, čímž může usnadnit útok.

Opačným problémem je režie metod pro odolnost proti útokům. Větší komplexnost takového návrhu může negativně ovlivnit jeho spolehlivost.

Dalším úskalím současného použití metod odolných proti útokům a odolným proti poruchám je režie. Jak již bylo zmíněno, režie plochy i spotřeby může být u těchto metod značná a při jejich kombinaci by mohla být neúnosná. Proto by bylo vhodné prozkoumat společné vlastnosti návrhových metod pro odolnost proti útokům a odolnost proti poruchám a tyto společné vlastnosti využít k vytvoření nových návrhových metod kombinujících oba požadavky, které výslednou režii sníží.

C. Cíle dizertační práce

Pro svou dizertační práci jsem si vytyčil tři základní cíle:

- Prozkoumat vliv architektur odolných proti poruchám na odolnost proti útokům
- Prozkoumat vliv architektur odolných proti útokům na spolehlivost
- Navrhnout nové návrhové metody (nebo přizpůsobit existující), které budou odolné proti poruchám i proti útokům a zároveň sníží režii oproti kombinaci existujících metod

Vzhledem k rozsahu výzkumu se chci ve své práci zaměřit na útok rozdílovou příkonovou analýzou (differential power analysis — DPA) [4]. Tento útok je jeden z nejběžnějších útoků postranním kanálem. Navíc metody pro odolnost proti

tomuto útoku jsou často účinné i proti jiným postranním kanálům [5]. Dále se chci omezit především na šifru AES jakožto nejběžnější blokovou šifru. Implementační platformou budou FPGA, která, která umožňují snadné a levné nasazování různých implementací.

Výstupem práce by mělo být podrobné porovnání vzájemného vlivu současných architektur odolných proti poruchám a architektur odolných proti útokům. Dále by výstupem měl být návrh jedné, nebo více architektur spojujících odolnost proti poruchám a odolnost proti útokům.

II. Související práce

Existuje mnoho návrhových technik pro zajištění odolnosti proti poruchám. Kromě běžných metod využívajících informační, časové nebo plošné redundance [1] [6] existují i takové, které využívají možnosti rekonfigurace FPGA [7] [8] [9].

Podobně existuje značné množství metod, jak učinit číslicový návrh odolný proti příkonovým útokům (power attacks), například maskování [10] [11], dvoudrátová (dual-rail) logika [12] [13], nebo prahová implementace (threshold implementation) [14] [15].

Naproti tomu si nejsme vědomi žádného výzkumu, který by se věnoval vzájemnému vlivu a společným vlastnostem návrhu pro odolnost proti poruchám a návrhu pro odolnost proti útokům. Částečnou podobnost jsme objevili pouze ve dvou studiích, které se zabývají vlivem návrhu odolného proti chybovým útokům na odolnost proti příkonovým útokům. Architektury odolné proti chybovým útokům na rozdíl od architektur odolným proti poruchám obvykle zabezpečují proti chybám pouze takové části obvodu, které lze při injekci poruchy použít k získání šifrovacího klíče, což je v případě šifry AES zejména funkce S-Box.

Regazzoni a spol. [16] [17] porovnávají odolnost proti DPA různých implementací funkce S-Box šifry AES, které jsou zabezpečeny různými kódy detekujícími či opravujícími chyby. Tyto implementace jsou syntentizovány pomocí standardní knihovny buněk pro 90nm proces ST-Microelectronics. Odolnost proti DPA je vyhodnocena pomocí simulace v programu SPICE.

Další podobné vyhodnocení provedli Dofe a spol. [18], kteří porovnávali funkce S-Box a MixColumns šifry AES zabezpečené proti chybovým útokům. Vyhodnocení odolnosti proti DPA v tomto případě probíhalo měřením reálné spotřeby na FPGA Spartan-6.

Výsledkem obou studií je závěr, že architektury pro odolnost proti chybovým útokům negativně ovlivňují odolnost proti příkonovým útokům.

III. Dosavadní postup prací

V současné době se věnujeme zkoumání vlivu metod odolných proti poruchám na odolnost proti útoku rozdílovou příkonovou analýzou (DPA). Využíváme přitom korelační variantu útoku [19]. Experimentálně vyhodnocujeme, jak různé architektury pro odolnost proti poruchám ovlivňují náročnost útoku DPA na šifru AES [20] implementovanou v FPGA.

Pro experimentální vyhodnocení bylo potřeba nejdříve připravit softwarovou a hardwarovou platformu pro útok pomocí DPA a zvolit vhodnou metodu pro porovnávání náročnosti útoku jednotlivých implementací odolných proti poruchám.

A. DPA hardware

Při útoku DPA je potřeba naměřit průběhy spotřeby s různými otevřenými texty. Pro účely měření je potřeba vhodně připravit FPGA desku. V tuto chvíli máme pro DPA připravené tyto platformy:

- Evariste II s FPGA modulem Altera Cyclone III [21]
- Xilinx Spartan-3E Starter board
- Nízkonákladovou desku s čipem Xilinx Artix-7 [22]

Dosavadní výsledky byly naměřeny na platformě Evariste II, nicméně ostatní platformy budeme využívat za účelem porovnání výsledků na různých FPGA.

B. DPA software

Při útoku je potřeba zvolit vhodný model spotřeby a pomocí něj určit odhadovanou spotřebu (hodnotu závislou na otevřeném nebo šifrovaném textu a klíči) pro každý použitý otevřený text a každou možnou hodnotu klíče (klíč je pro tento účel rozdělen na jednotlivé byty). Poté vypočítáme korelace mezi naměřenou reálnou spotřebou a odhadovanou spotřebou. Nejvyšší absolutní hodnota korelace by pak měla odpovídat správnému klíči.

Pro účely výpočtů korelací jsme používali software Wolfram Mathematica, který ovšem nesplňoval naše požadavky na rychlost výpočtu. Proto jsme se rozhodli ve spolupráci s mým diplomantem implementovat pro tento účel vlastní aplikaci.

Zvolili jsme několik různých přístupů, implementovali je a porovnali s ohledem na výpočetní a paměťovou efektivitu, numerickou stabilitu a paralelizovatelnost. Výsledky tohoto snažení jsme publikovali na konferenci DDECS 2017 [VM.2].

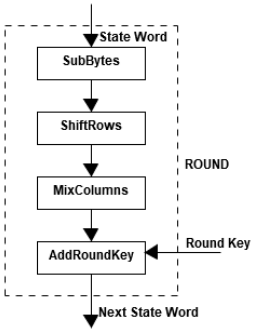
Nejlepší z porovnávaných algoritmů je časově efektivní, má velmi nízké paměťové nároky a je numericky stabilní. Je téměř lineárně paralelizovatelný i pro desítky výpočetních jader. Navíc umožňuje výpočet přerušit, prozkoumat výsledky a v případě potřeby pokračovat ve výpočtu. Momentálně pracujeme na úpravě tohoto algoritmu pro výpočet na GPU.

Aplikaci se chystáme dále rozšířit a vytvořit komplexní univerzální framework pro výpočty pro DPA.

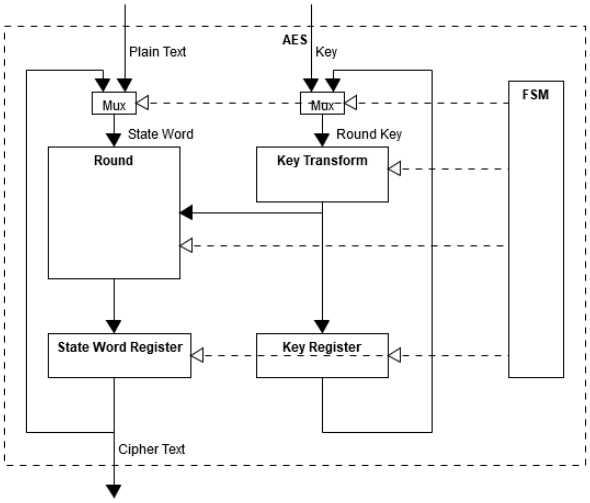
C. Experimenty

Provedli jsme dvě porovnání odolnosti proti DPA různých architektur odolných proti poruchám. V prvním jsme porovnávali využití plošné redundance s různým počtem šifrovacích modulů. Ve druhém jsme se věnovali různým druhům redundance implementovaným na různé úrovni návrhu.

Použité architektury vycházely z implementace šifry AES s šířkou klíče 128 bitů. Jedná se o sekvenční implementaci, kdy runda je implementována jako kombinační obvod (viz Obrázek 1) a stavový registr. Podobně je implementován také generátor rundovních klíčů. Vše je řízeno automatem. Celé šifrování trvá



Obrázek 1. Schéma implementace rundy šifry AES



Obrázek 2. Schéma implementace modulu AES

11 hodinových taktů (10 rund + inicializační runda). Schéma základní implementace AES je vyobrazeno na Obrázku 2.

1) Porovnání plošné redundance: Při vyhodnocování vlivu plošné redundance jsme se zaměřili na tři implementace:

- Základní AES
- Duplex (2 moduly + komparátor)
- TMR (3 moduly + volič majority)

Pro každou implementaci jsme naměřili 50 sad průběhů spotřeby, každá sada obsahovala 2000 průběhů. Následně jsme prováděli pro každou sadu výpočet korelací pro DPA při různém počtu použitých průběhů a zjišťovali, jaké je minimální množství použitých průběhů, při kterém získáme všech 16 bytů správného klíče. Porovnání výsledků pro jednotlivé implementace (různé počty použitých AES modulů) je vyobrazeno v Tabulce I.

Z výsledků je patrné, že zvolené metody plošné redundance mají velmi malý vliv na počet průběhů spotřeby potřebný k získání správného klíče. Jelikož rozdíly mezi jednotlivými výsledky jsou menší než směrodatná odchylka, jsou tyto rozdíly statisticky nevýznamné.

Základní myšlenky a metodika vyhodnocování tohoto experimentu byly publikovány v článku na konferenci MECO 2016 [VM.1]. Rozšířený o podrobné výsledky byl tento článek publikován v časopise Microprocessors and Microsystems [VM.3].

Tabulka I
POROVNÁNÍ PRŮMĚRŮ A ROZPTYLŮ MINIMÁLNÍCH MNOŽSTVÍ ZPRACOVANÝCH PRŮBĚHŮ SPOTŘEBY NUTNÝCH PRO ZÍSKÁNÍ SPRÁVNÉHO KLÍČE PRO RŮZNÉ MNOŽSTVÍ POUŽITÝCH AES MODULŮ

	Základní AES	Duplex	TMR
Průměr	745.28	741.74	719.22
Rozptyl	13,281.59	18,222.03	14,802.38

Tabulka II
POROVNÁNÍ MEDIÁNŮ A MEZIKVARTILOVÝCH ROZPĚTÍ MINIMÁLNÍCH MNOŽSTVÍ ZPRACOVANÝCH PRŮBĚHŮ SPOTŘEBY NUTNÝCH PRO ZÍSKÁNÍ SPRÁVNÉHO KLÍČE PRO RŮZNÉ DRUHY REDUNDANCE

Architektura	Medián	Mezikvartilové rozpětí	Rozdíl oproti AES
AES	850	175 (20.5%)	0%
AES-SPC	950	250 (26.3%)	+12%
AES-HR-R	900	275 (30.5%)	+6%
AES-HR-A	812	150 (18.5%)	-4%
AES-TR-R	1025	250 (24.4%)	+21%
AES-TR-A	1037	275 (26.5%)	+22%

2) Porovnání různých druhů redundance: Pro porovnávání vlivu různých druhů redundance na odolnost proti DPA jsme zvolili tyto varianty AES:

- Informační redundance — ověřování vstupní a výstupní parity funkce S-Box (AES-SPC)
- Plošná redundance — TMR na dvou různých úrovních:
  - na úrovni rundy — kombinační obvod rundy a stavový registr jsou ztrojeny a výstupy jsou porovnávány voličem majority (AES-HR-R)
  - na úrovni algoritmu — celý šifrovací modul je ztrojen a výstupy jsou porovnávány voličem majority (AES-HR-A)
- Časová redundance — výpočet je prováděn třikrát na dvou různých úrovních
  - na úrovni rundy — každá runda se provádí třikrát, stavový registr je ztrojen a výsledky jsou porovnávány voličem majority (AES-TR-R)
  - na úrovni algoritmu — celé šifrování se provádí třikrát a výsledky jsou porovnávány voličem majority (AES-TR-A)

Pro každou implementaci jsme naměřili 50 sad průběhů spotřeby, každá sada obsahovala 2000 průběhů. Následně jsme prováděli pro každou sadu výpočet korelací pro DPA při různém počtu použitých průběhů a zjišťovali, jaké je minimální množství použitých průběhů, při kterém získáme všech 16 bytů správného klíče. Porovnání výsledků pro jednotlivé implementace je vyobrazeno v Tabulce II.

Z výsledků je patrné, že zvolené metody redundance mají velmi malý vliv na počet průběhů spotřeby potřebný k získání správného klíče. Jelikož rozdíly mezi jednotlivými výsledky jsou menší než mezikvartilové rozpětí, jsou tyto rozdíly statisticky nevýznamné.

Článek s výsledky tohoto porovnání byl přijat na konferenci DSD 2017 [VM.4]

IV. Závěr

Vytvořili jsme hardwarové a softwarové prostředí pro účely porovnávání odolnosti proti útoku DPA různých architektur implementovaných v FPGA. Porovnali jsme metody plošné redundance lišící se počtem modulů a dále jsme porovnali různé metody redundance. Výsledkem experimentů bylo zjištění, že rozdíl mezi zvolenými architekturami odolnými proti poruchám je statisticky nevýznamný. Dále bychom chtěli tato zjištění ověřit v obsáhlejším porovnání, které bude zahrnovat více použitých FPGA, větší množství architektur odolných proti poruchám, různé syntézní parametry a různé metody vyhodnocení. Toto porovnání by mělo být finálním výsledkem prvního cíle mé disertační práce. Poté se chystáme věnovat dalším cílům práce vytyčeným v sekci I-C.

Poděkování

Tento výzkum byl podporován grantem GA16-05179S Grantové agentury České Republiky, „Výzkum vztahů a společných vlastností spolehlivých a bezpečných architektur založených na programovatelných obvodech“ (2016-2018) a projektem ČVUT SGS17/017/OHK3/1T/18.

Literatura

[1] D. Pradhan, *Fault-tolerant computer system design*. Upper Saddle River, N.J: Prentice Hall PTR, 1996.

[2] T.-H. Le, C. Canovas, and J. Clédiere, “An overview of side channel analysis attacks,” in *Proceedings of the 2008 ACM symposium on Information, computer and communications security*. ACM, 2008, pp. 33–43.

[3] H. Ziade, R. A. Ayoubi, R. Velazco *et al.*, “A survey on fault injection techniques,” *Int. Arab J. Inf. Technol.*, vol. 1, no. 2, pp. 171–186, 2004.

[4] P. Kocher, J. Jaffe, and B. Jun, “Differential power analysis,” in *Annual International Cryptology Conference*. Springer, 1999, pp. 388–397.

[5] E. Peeters, F.-X. Standaert, and J.-J. Quisquater, “Power and electromagnetic analysis: Improved model, consequences and comparisons,” *Integration, the VLSI journal*, vol. 40, no. 1, pp. 52–60, 2007.

[6] I. Koren, *Fault-tolerant systems*. Amsterdam Boston: Elsevier/Morgan Kaufmann, 2007.

[7] J. Emmert, C. Stroud, B. Skaggs, and M. Abramovici, “Dynamic fault tolerance in FPGAs via partial reconfiguration,” in *Field-Programmable Custom Computing Machines, 2000 IEEE Symposium on*. IEEE, 2000, pp. 165–174.

[8] A. Alaghi, M. S. Yarandi, and Z. Navabi, “An optimum ORA BIST for multiple fault FPGA look-up table testing,” in *2006 15th Asian Test Symposium*. IEEE, 2006, pp. 293–298.

[9] J. Lach, W. H. Mangione-Smith, and M. Potkonjak, “Low overhead fault-tolerant FPGA systems,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 2, pp. 212–221, 1998.

[10] J. Blömer, J. Guajardo, and V. Krummel, “Provably secure masking of AES,” in *International Workshop on Selected Areas in Cryptography*. Springer, 2004, pp. 69–83.

[11] T. S. Messerges, “Securing the AES finalists against power analysis attacks,” in *International Workshop on Fast Software Encryption*. Springer, 2000, pp. 150–164.

[12] J.-L. Danger, S. Guilley, S. Bhasin, and M. Nassar, “Overview of dual rail with precharge logic styles to thwart implementation-level attacks on hardware cryptoprocessors,” in *Signals, Circuits and Systems (SCS), 2009 3rd International Conference on*. IEEE, 2009, pp. 1–8.

[13] K. Tiri, D. Hwang, A. Hodjat, B.-C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, “Prototype IC with WDDL and differential routing–DPA resistance assessment,” in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2005, pp. 354–365.

[14] S. Nikova, C. Rechberger, and V. Rijmen, “Threshold implementations against side-channel attacks and glitches,” in *International Conference on Information and Communications Security*. Springer, 2006, pp. 529–545.

[15] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, “Pushing the limits: a very compact and a threshold implementation of AES,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2011, pp. 69–88.

[16] F. Regazzoni, T. Eisenbarth, L. Breveglieri, P. Ienne, and I. Koren, “Can knowledge regarding the presence of countermeasures against fault attacks simplify power attacks on cryptographic devices?” in *Defect and Fault Tolerance of VLSI Systems, 2008. DFTVS'08. IEEE International Symposium on*. IEEE, 2008, pp. 202–210.

[17] F. Regazzoni, L. Breveglieri, P. Ienne, and I. Koren, “Interaction between fault attack countermeasures and the resistance against power analysis attacks,” in *Fault Analysis in Cryptography*. Springer, 2012, pp. 257–272.

[18] J. Dofe, H. Pahlevanzadeh, and Q. Yu, “A Comprehensive FPGA-Based Assessment on Fault-Resistant AES against Correlation Power Analysis Attack,” *Journal of Electronic Testing*, vol. 32, no. 5, pp. 611–624, 2016.

[19] E. Brier, C. Clavier, and F. Olivier, “Correlation power analysis with a leakage model,” in *International Workshop on Cryptographic Hardware and Embedded Systems*. Springer, 2004, pp. 16–29.

[20] N. F. Pub, “197: Advanced encryption standard (AES),” *Federal Information Processing Standards Publication*, vol. 197, no. 441, p. 0311, 2001.

[21] V. Fischer, F. Bernard, and P. Haddad, “An open-source multi-FPGA modular system for fair benchmarking of true random number generators,” in *Field Programmable Logic and Applications (FPL), 2013 23rd International Conference on*. IEEE, 2013, pp. 1–4.

[22] M. Bartík and J. Buček, “A low-cost multi-purpose experimental fpga board for cryptography applications,” in *Advances in Information, Electronic and Electrical Engineering (AIEEE), 2016 IEEE 4th Workshop on*. IEEE, 2016, pp. 1–4.

Publikace autora

Přijaté recenzované publikace

[VM.1] V. Miškovský, H. Kubátová, M. Novotný *Influence of fault-tolerant design methods on differential power analysis resistance of AES cipher: Methodics and Challenges*. 5^{th} Mediterranean Conference on Embedded Computing (MECO 2016), Bar, Montenegro, 2016.

[VM.2] P. Socha, V. Miškovský, H. Kubátová, M. Novotný *Optimization of Pearson correlation coefficient calculation for DPA and comparison of different approaches*. 20^{th} International Symposium on Design and Diagnostics of Electronic Circuit & Systems (DDECS), Dresden, Germany, 2017.

[VM.3] V. Miškovský, H. Kubátová, M. Novotný *Influence of Passive Hardware Redundancy on Differential Power Analysis Resistance of AES Cipher implemented in FPGA*. Microprocessors and Microsystems – Elsevier B.V 2017, Volume 51, pp. 220-226, ISSN 0141-9331 2017.

[VM.4] J. Říha, V. Miškovský, H. Kubátová, M. Novotný *Influence of Fault-Tolerance Techniques on Power-Analysis Resistance of Cryptographic Design*. Euromicro Conference on Digital System Design (DSD 2017), Vienna, Austria, 2017. (přijato)

Další publikace

[VM.5] V. Miškovský, H. Kubátová, M. Novotný *Influence of Fault-tolerant Design Methods on Resistance against Differential Power Analysis*. The 4^{th} Prague Embedded Systems Workshop, Roztoky u Prahy, Česká republika, 2016.

[VM.6] V. Miškovský, H. Kubátová, M. Novotný *Číslicový návrh spojující odolnost proti útokům a odolnost proti poruchám*. Počítačové architektury a diagnostika, Bořetice, Česká republika, 2016.

[VM.7] V. Miškovský, H. Kubátová, M. Novotný *Influence of Fault-tolerant Design Methods on Resistance against Differential Power Analysis in FPGA*. Conference on Trustworthy Manufacturing and Utilization of Secure Devices, Barcelona, Spain, 2016.

[VM.8] V. Miškovský *Influence of Fault-Tolerant Design Techniques on Resistance against Differential Power Analysis*. CryptArchi, Smolenice, Slovakia, 2017.

Configurable Reprogramming Methodology for Embedded Low-Power Devices

Ondrej Kachman
3rd year, full-time study
Supervisor: Doc. Ing. Ladislav Hluchý, PhD.; Consultant: Ing. Marcel Baláž, PhD.

Institute of Informatics, Slovak Academy of Sciences
Dúbravská cesta 9, Bratislava, Slovak Republic
ondrej.kachman@savba.sk

***Abstract*— The embedded low-power devices are very important part of any modern intelligent system. With the large amounts of sensors and actuators used, it is a good practice to implement remote reprogramming capabilities into the firmwares of these devices. This paper presents a new configurable reprogramming methodology that can be applied to various platforms. It is built on the best reprogramming practices while giving developers more control over firmware outline, updated functions and modules. It also refers energy efficiency, as the data shared over the network and memory operations on the devices are minimal. The multiplatform capabilities make this methodology ideal for smart systems.**

Keywords*— *reprogramming, multiplatform, embedded, low-power, configurable, over-the-air

I. Introduction

The area of remote and efficient reprogramming of the embedded devices has been researched since the introduction of the low-power devices and the wireless sensor networks. Various reprogramming methods have been developed over the past 15-20 years. The main goal of these methods is to replace the old firmware version with the new one while keeping the procedure energy efficient and secure.

A. Motivation and challenges

Recent progress in the internet of things technologies, cyber-physical systems and smart systems requires these methods to be evaluated and adjusted to the new trends in these areas. Some of the low-power devices used in the modern systems may be physically inaccessible and battery powered. When a firmware update is required, small, wirelessly transferred delta files (deltas) are used to save energy. These deltas encode only the difference between the old and the new firmware version [1]. The new challenges for the reprogramming mechanisms in the modern systems are following:

- Multiplatform use – The algorithms used by the mechanism should be applicable to every platform used by the system
- Configurability – For the different platforms, different approaches to reprogramming may be better.
- Energy efficiency – keep the amount of data shared on the network during update minimal.

This paper describes methods and algorithms for the remote firmware updates that address these challenges. Together, they create a configurable reprogramming methodology suitable for modern intelligent systems.

II. Related work

The area of remote reprogramming for the low-power devices can be split to multiple subareas. Existing works can be split to two main groups:

1. Delta file generation – methods and algorithms focused on the generation of as small delta file as possible
2. Delta dissemination – methods and algorithms that are used to propagate the delta files through the network to the target devices

Recent advances in the network protocols for internet of things make it possible to use its standard protocols to safely and securely disseminate the delta files [2]. Delta dissemination is therefore out of scope of this paper.

This paper is focused on the delta generation methods and algorithms. These can be split into three groups:

1. Firmware similarity improvement algorithms
2. Delta file generators
3. Update agents

Some existing works are focused on a single group, some target multiple groups.

A. Methods and algorithms used to generate a delta file

The following subchapters describe related work for the process of the delta file generation.

1) Firmware similarity improvement algorithms

The authors of [3] introduced changes to the compiler in order to preserve register allocations for the variables shared between the different firmware versions. The method helped to generate smaller delta files but worsened the execution time of the firmware by inserting more MOV instructions into the code.

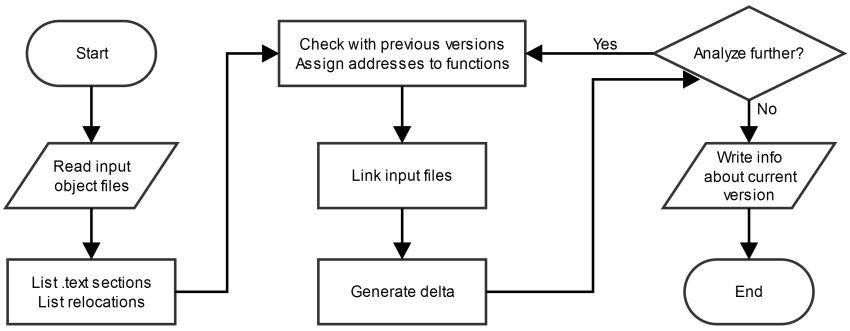


FIG. 1 FLOWCHART OF THE PROPOSED METHODOLOGY

The method called 'remote incremental linking' [4] introduced slop regions – a free space in the program memory between the firmware functions. This approach enabled functions to grow and shrink in their slop regions, reducing the shifts and relocation changes in the firmware, making it possible to generate smaller delta files.

The reprogramming mechanism called R3 [5] uses the ELF files to remove relocation entries (jumps, calls, branches) from the firmware before generating the delta file. Afterwards, the relocations are added to the delta as compressed metadata.

2) Delta file generators

These generators compare old and new versions and encode their differences into the delta files. Older algorithm that had a wider use was called RMTD [6]. It was a block based algorithm. R3 [5] used R3diff algorithm, a byte based algorithm with very good results. We developed our own algorithm, Delta Generator [1], that introduced multiple optimizations and performs better than R3diff.

3) Update agents

Update agent is a function or a module present on the target device that applies the delta file. It may be a part of the device's bootloader or reside in its own flash section. There are two main update agent strategies:

- External memory updates [5] – firmware is rebuilt in the external memory and bootloader loads it into the program memory
- On-the-fly updates [1][7] – operations encoded within the delta file are applied on the fly.

B. Towards the multiplatform solutions

The authors of [8] created a framework for reprogramming of AVR microcontrollers. This framework can be configured to evaluate different reprogramming mechanisms. This helps to determine the best mechanism for a chosen firmware. Our methodology includes various configurations to address the different firmware reprogramming problems.

MobileDeluge [9] is a mobile network reprogramming tool for heterogeneous wireless sensor networks. It supports wireless devices based on different platforms, thus it addresses the problem of multiplatform support. It is designed to communicate with the target devices directly to issue reprogramming commands. Our methodology relies on the networking protocols used by the system to disseminate the delta files.

III. CONFIGURABLE METHODOLOGY FOR THE REMOTE REPROGRAMMING

This chapter presents our remote reprogramming methodology. It consists of methods that form a complete, platform independent solution capable of learning about the changes in any firmware and generating as small delta files as possible. Our methods are applied in a software tool we developed. Our methodology includes methods responsible for the following tasks:

- Analyzing the object files, listing .text sections and relocations
- Tracking changes for each function, detecting new or deleted functions
- Assigning addresses to the functions based on their analysis
- Invoking linker, checking for relaxed entries in the final files
- Generating delta files

Our tool includes a wrapper for the GCC compiler and linker. We aim to test our method and tool on the three different platforms – 8-bit microcontroller from the AVR family, 16-bit microcontroller from the MSP430 family and a system-on-chip with the 32-bit ARM Cortex-M0 core. Each platform has its version of the GCC available and supports ELF files. We do not alter the instructions generated by the compiler. The flowchart of our method is in the Fig. 1.

A. Description of the processes and configurability options

This subchapter describes the configurable processes from the Fig. 1. These are the processes within our proposed method that contribute to its good performance.

1) List text sections, list relocations

Compilers enable developers to generate a .text section for each function, for example .text.function1, .text.function2. However, both will be linked into the .text section in the executable file. To prevent this, programmer must define a target section for each function in the source code. Our method includes two configurations:

- Split functions into the sections defined explicitly by the developer
- Edit string tables of the object files, place each function in its own section

The second option will simply iterate through the ELF string tables and add the number to each .text section, putting previously mentioned functions into sections .text1 and .text2. This does not affect any firmware instructions.

Relocation entries can be read at this point, even though they are not resolved. List of these relocations may be later used before the effective delta generation by setting them to zero and making the firmware images more similar – inspired by [4].

2) Check with previous versions, assign addresses to the functions

In this step, our method collects information about every function that was present in the previous version. We evaluate its growth, previous positions, free space in its slop region (if present) and memory shifts its change may cause. The results of these evaluations can result in the different outline of the firmware's functions depending on the following possible configurations:

- Static addresses – developer can assign some functions to a static address
- No fragmentation – No function will be provided by a slop region
- Partial fragmentation – Functions that would cause a lot of shifts are copied to a free space where they can be further edited
- Full fragmentation – Every function is provided with a slop region.

3) Generate delta.

After linking, when the relocations are resolved, there are two possible configurations:

- Set all the relocations to zero, generate insert operations for the delta
- Do not alter the relocations, run the differencing algorithm directly

Differencing algorithm also supports two different modes. These modes are relevant when the firmware is being fragmented or defragmented:

- Dirty mode – unused sections that previously contained data are not erased
- Clean mode – unused sections are erased and filled with 0xFF symbols.

B. *Contribution and innovation of our solution*

None of the methods used in our methodology alter any ELF sections that hold instructions or relocations. There is no need to specify instruction formats or relocation types for any platform. This makes our solution platform independent.

Our methodology provides developers with various configurable options described in the previous subchapter. Produced output files can be reverse analyzed and provide feedback on how the current configuration can be improved. This is a new approach. Various configurations may be the best for different scenarios.

Our methodology generates smaller deltas than basic differencing algorithms. Less data shared on the network improves the energy efficiency of our solution.

IV. Experimental results and discussion

This section provides experimental results that show, how various configurations of our methodology may be the most suitable for the different scenarios of over-the-air updates. In [1], we presented our differencing algorithm that outperforms existing solutions. Therefore, we use it for the following experiments executed on the ATmega32u4 microcontroller. We consider three scenarios:

1. In the first scenario, the devices on the network require one big update of the functions that sense, store and send data – the sensor module
2. The second scenario updates these three sensor module functions incrementally, one after another
3. The third scenario involves 9 incremental updates to various functions in the firmware code – the sensor module (3 functions), the communication module (3 functions), the flash storage module (3 functions)

We configure our tool to edit string tables of the ELF files, every function is placed in its own section. We do not set relocations to zero. For each scenario, we evaluate three different configurations:

Table 1 Experimental results for the three different scenarios and configurations

Config	Scenario 1 - single update			Scenario 2 -3 updates			Scenario 3 - 9 updates		
	Delta files	Total bytes	Frag. overhead	Delta files	Total bytes	Frag. overhead	Delta files	Total bytes	Frag. overhead
No frag.	1	98 (best)	0	3	216 (+7%)	0	9	1398 (+65%)	0
Partial frag.	2	164 (+64%)	112	4	202 (best)	102	10	844 (best)	260
Full frag.	3	680 (+593%)	634	5	638 (+215%)	576	11	1036 (+22%)	568

Table 2 Delta file sizes in bytes for the Scenario 3

Config	Δ_{frag}	Δ1	Δ2	Δ3	Δ4	Δ5	Δ6	Δ7	Δ8	Δ9	Δ_{defrag}	Total
No frag.	-	74	84	58	106	162	150	244	102	418	-	1398
Partial frag.	-	12	40	48	26	126	50	200	58	24	260	844
Full. frag.	302	14	14	34	26	126	20	200	20	14	266	1036

1. No fragmentation; All deltas are generated in clean mode
2. Partial fragmentation; Changed functions are provided with a slop region (dirty mode), final delta generated in the clean mode defragments the memory
3. Full fragmentation; All functions are provided with slop regions before any update (clean mode), functions are then updated (dirty mode) and finally, the firmware is defragmented (clean mode).

A. Results

The results of our experiments are shown in the Table 1. Column "Delta files" shows, how many delta files were used for the update. Column "Total bytes" shows the sum of the delta file sizes used for reprogramming. Column "Frag. overhead" shows, how many of those bytes were used for memory fragmentation, defragmentation and cleanup. Table 2 shows the sizes of every delta file from the 3rd scenario. The sizes are in bytes.

B. Discussion

The experiment shows, that the single firmware update does not require any memory fragmentation to be efficient. With more incremental updates required, configurations with memory fragmentation become more efficient – partial fragmentation performs the best for both 3 and 9 incremental updates. Full fragmentation configuration has the penalty of the big deltas that are used to fragment the memory at the beginning and to defragment it at the end. It is performing significantly better for more updates and it can be expected, that for a great number of incremental updates, this configuration will perform the best. Table 2. shows, that the delta sizes for the full fragmentation approach are mostly the smallest for the single function updates (Δ1- Δ9).

V. Aims of the Dissertation Thesis

This chapter list the aims of the dissertation thesis and describes how they are or will be addressed.

A. Definition of parameters that influence performance and energy consumption of over-the-air firmware updates

The thesis will provide in-depth analysis of the chosen problem. It will list all known challenges in this area along with their solutions. Some of these problems have been mentioned throughout this paper.

B. Proposal of an energy consumption estimation model for over-the-air updates of low-power devices

Energy consumption estimation models can help evaluate any reprogramming scheme. With the more possible configurations of the firmware, these models can help choose the most effective update strategy. We published paper that describes how these models can help evaluate the energy efficiency of the reprogramming schemes – [10].

C. Design of a reprogramming scheme for fast and energy effective over-the-air updates of low-power devices

This paper described the methodology we use for the efficient over-the-air reprogramming of low-power devices in the modern intelligent systems. We use the best methods in the area and build on them.

D. Implementation of a proposed scheme and its evaluation on a chosen hardware platforms

We implemented a software tool to help us evaluate our methodology. Currently, most of the experiments carried out were executed for the AVR microcontroller. Future work includes experiments on the different platforms.

VI. Conclusion

This paper presents a configurable reprogramming methodology that is built on the best practices in the area, and introduces configurations that make remote firmware updates more efficient for the different reprogramming scenarios. Experiments have shown, that various configurations reduce the total amount of data shared on the network. The solution is multiplatform, which makes it ideal for the modern intelligent systems.

VII. Acknowledgement

This work has been supported by Slovak national project VEGA 2/0192/15.

VIII. References

[1] O. Kachman and M. Balaz, "Optimized Differencing Algorithm for Firmware Updates of Low-Power Devices," in *19th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems*, Kosice, 2016.

[2] V. Kumar and M. Sanjay, "Efficient and Secure Code Dissemination in Sensor Clouds," in *IEEE 15th International Conference on Mobile Data Management (MDM)*, Brisbane, 2014.

[3] Y. Huang, M. Zhao and C. J. Xue, "WUCC: Joint WCET and Update Conscious Compilation for cyber-physical systems," in *18th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Yokohama, 2013.

[4] J. Koshy and R. Pandey, "Remote Incremental Linking for Energy-Efficient Reprogramming fo Sensor Networks," in *Proceeedings of the Second European Workshop on Wireless Sensor Networks*, 2005.

[5] W. Dong, B. Mo, C. Huang, Y. Liu and C. Chen, "R3: Optimizing relocatable code for efficient reprogramming in networked embedded systems," in *IEEE INFOCOM Proceedings*, Turin, IEEE, 2013, pp. 315 - 319.

[6] J. Hu, C. J. Xue, Y. He and E. H.-M. Sha, "Reprogramming with Minimal Transferred Data on Wireless Sensor Network," in *IEEE 6th International Conference on Mobile Adhoc and Sensor Systems (MASS '09)*, Macau, IEEE, 2009, pp. 160 - 167.

[7] N. B. Shafi, K. Ali and H. S. Hassanein, "No-reboot and Zero-Flash Over-the-air Programming for Wireless Sensor Networks," in *9th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON)*, Seoul, 2012.

[8] B. Mazumder and J. O. Hallstrom, "VSPIN: A Framework for Developing Incremental Sensor Network Reprogramming Strategies," in *4th International Workshop on Software Engineering for Sensor Network Applications (SESENA)*, San Francisco, 2013.

[9] X. Zhong, M. Navarro, G. Villalba, X. Liang and Y. Liang, "MobileDeluge: Mobile Code Dissemination for Wireless Sensor Networks," in *11th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, Philadelphia, 2014.

[10] O. Kachman and M. Balaz, "Effective Over-the-Air Reprogramming for Low-Power Devices in Cyber-Physical Systems," in *Technological Innovation for Cyber-Physical Systems*, Lisbon, 2016.

State Synchronization of Faulty Soft Core Processors in Reconfigurable TMR Architecture

Karel Szurman
5th year, part-time study
Supervisor: Zdeněk Kotásek

Faculty of Information Technology, Brno University of Technology
Božetěchova 1/2, 612 66 Brno
iszurman@fit.vutbr.cz

***Abstract*—Fault-tolerant systems implemented into SRAM-based FPGA are frequently protected by combination of triple modular redundancy and partial dynamic reconfiguration. When a part of the SRAM configuration memory with the copy of the protected circuit is reconfigured on the run, the system restart is the easiest way how to bring all three copies of the circuit back to fully synchronous and operating state. Soft core processors are complex systems which require more precise technique for synchronization of the system state space and data gained from previous calculations without disruption of processors functionality and executed program. This paper presents current state of our research focused on the state synchronization methodology for soft core processors.**

Keywords—Fault tolerant system, FPGA, state synchronization, partial dynamic reconfiguration, recovery, soft core processor.

I. INTRODUCTION

Emerging technologies used in avionics and space systems have growing demands on computing frequency and data throughput. Examples of such systems can be LiDAR (Light Detection and Ranging) system for 3D sensing of the Earth surface in real time, software defined radio system for space telecommunication or research satellite using a number of highly accurate sensors for data acquisition during its exploration mission. These digital systems are usually based on combination of Digital Signal Processors (DSPs), Field Programmable Gate Arrays (FPGAs), Application Specific Integrated Circuits (ASICs) and custom-made electronic hardware. Avionics and space systems are safety-critical systems the failure of which can lead to catastrophic consequences. These systems are exposed to various failure conditions during their lifetime. Radiation effects, caused by energetic particles in the space radiation environment, are one of the most serious. Nowadays, SRAM-based FPGAs become broadly used inside these systems for their low price, high performance, ability for reprogramming and flexibility even when they are very sensitive to radiation effects and mainly to Single Event Upset (SEU) effect. SEUs can cause changes in a state of bistable element and affect configuration memory and user logic. Usage of SRAM-based FPGA requires implementation of SEU mitigation technique and employing of fault tolerance strategy to operate correctly even in the presence of failure.

Two main SEU mitigation strategies for Fault Tolerant Systems (FTSs) exist [1]. Both approaches employ hardware redundancy. The most used form is Triple Modular Redundancy (TMR) due to its fault-masking ability, possibility to scale the TMR protection by changing its granularity, tolerable overhead and the availability of tools for automated TMR generation [2]. The first technique, referred as scrubbing, is based on periodic writing a known copy of the original bitstream to configuration memory in order to correct corrupted bits. The copy is stored usually in external radiation-hardened memory. The scrubbing has various implementations and it is also often combined with hardware redundancy applied automatically on netlist of the circuit. The second technique is based on the usage of hardware modular redundancy together with Partial Dynamic Reconfiguration (PDR). In our terminology, TMR consists of replicated circuits, the circuits contain components. Fault detection in a TMR is operating by means of majority voting from copies of protected circuit. When a failure caused by SEU in one of the circuit copies is detected then corresponding TMR module located in FPGA configuration memory is reconfigured through PDR process. In our research, the PDR is controlled by Generic Partial Dynamic Reconfiguration Controller (GPDRC) [5]. After the faulty circuit reconfiguration is finished, its operational state is not up-to-date and needs to be synchronized with correctly operating circuit copies in TMR architecture before it is incorporated back into the system.

The aim of our research is to propose a new methodology for a state synchronization of faulty soft core processors in reconfigurable TMR architecture implemented into SRAM FPGA. We targeted soft core processors as a computation platform alternative to microprocessor, DSPs or general-purpose processor, which can be synthetized for any FPGA design and easily modified and customized by implementation of additional features or application of a specific optimization. The new methodology should complete existing FT methodologies which address fault masking, fault detection and fault recovery based on hardware redundancy and PDR.

II. State Synchronization Methodology

At the beginning of the research, we focused on methods for state synchronization for digital system implemented by random logic and state machines, synchronization strategies and parameters which could be evaluated. Then, we started to deal with the state synchronization for soft-core processors as the platform with complex functionality and structure where the state synchronization is more demanding.

A. Initial development phase

Results gained during the initial development phase of our state synchronization methodology are as follows [3] [4]:

- We defined a set of parameters (criteria) for evaluation and optimization of various state synchronization techniques.
- We developed fundamental state synchronization methodology and described the basic principles of state synchronization.
- We implemented reconfigurable fault tolerant CAN bus control system together with implementation of specific synchronization strategy based on our methodology.

We realized that the principles of state synchronization and its implementation have a strong impact on the FTS and its parameters. Therefore, all aspects including requirements on its real time behavior, principles of performing its function, the type and volume of the synchronized context must be taken into account when the method of a state synchronization after fault occurrence is developed.

B. State synchronization parameters

As the first, we identified set of dynamic and static parameters. The dynamic parameters reflect the impact of the synchronization on operation and function of the system and overall timing. According to the synchronization impact, the methods can be divided into function blocking and function non-blocking methods. Another dynamic parameter is the time needed to perform the synchronization which is closely associated with other parameters, requirements on the synchronization implementation and the volume of data which needs to be synchronized. On the other hand, the static parameters have an indirect impact on system features and have a close relation to an algorithm used to implement the synchronization procedure. This set of parameters include area and FPGA resource overhead, the power demands and reliability of implemented synchronization.

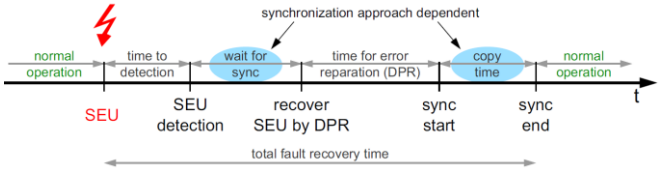


Figure 1: SEU recovery process [2]

The time needed for system resynchronization is the one of the most important criteria since the goal is to maximize Mean Time Between Failures (MTBF) metric. The overall timing for SEU recovery process in shown in Figure 1. The total fault recovery time is given by the sum of time durations of all following recovery phases:

- the time needed for error detection,
- the wait for synchronization time,
- the reconfiguration time for repairing the SEU,
- the synchronization time.

The time needed for error detection is the time between SEU occurrence and the moment when an error caused by this fault is detected in majority voter. The reconfiguration time is proportional to the size of reconfigured partition and speed. The wait for synchronization time is the time needed for finishing ongoing calculations. The synchronization time is the time required to copy the correct values from reference circuit instance to a recently reconfigured circuit.

C. Essential design considerations

We determined the set of essential design considerations which must be satisfied by the implemented synchronization method. The design considerations are following:

1) The selection of the state in which the synchronization of a reconfigured circuit copy will be performed – the proper state must be considered with respect to the synchronization method feasibility, real-time requirements, the system architecture and data consistency.
2) The definition of the system context which will be used for the synchronization – the context is defined by data type and its volume, the two types of data can be distinguished: the data which are processed and reproduced (i.e. application-related data) and the data which are important for system function and operation (i.e. system related data).
3) The design of the interconnection between redundant components and the control mechanism which will be needed for performing the synchronization process.

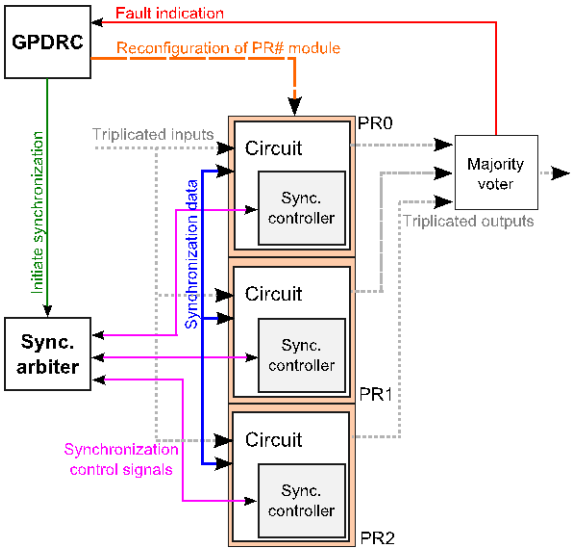


Figure 2: Generic architecture for state synchronization

D. *State synchronization implementation*

We designed generic architecture for synchronization with arbiter and controllers [3]. This architecture is shown in Figure 2. It was designed on the basic of consideration that the FTS architecture can consists of various components with complex hierarchy and in each of components, registers with data of the system state can be stored. For this reason, the synchronization process of a reconfigured circuit copy in the TMR architecture has to be controlled on two levels. From the outside, on the level of individual circuits, and inside on the level of circuits for synchronization of their internal components.

The basic principle of the fault recovery strategy is as follows. A protected circuit is implemented within coarse grained TMR architecture. Majority voter performs voting from input signals and indicates any mismatch to GPDRC and initiate reconfiguration of faulty circuit. After the process of reconfiguration is finished, GPDRC indicates the index of reconfigured unit to the synchronization arbiter which needs to be synchronized. The arbiter determines the roles of replicated units in TMR architecture (it specifies which of them is synchronized, referenced or paused during the synchronization process). The arbiter controls all state synchronization phases for components and data objects inside the synchronized circuit. In the end, the arbiter exits the synchronization process and switches the circuit copies back into its operational mode.

The crucial task for state synchronization strategy is the implementation of the interconnection between redundant modules which must be designed with compromise between FPGA resource utilization overhead, implementation complexity and the goal to complete the synchronization in the shortest possible time.

III. RECONFIGURABLE FT CAN BUS CONTROL SYSTEM

The proposed state synchronization methodology for digital system implemented on random logic and state machines was evaluated by experiments with reconfigurable FT CAN bus control system, published in [3] and [4].

IV. FAUL TOLERANCE AND SYNCHRONIZATION OF SOFT CORE PROCESSORS

Our current research is focused on soft core processors, aspects of their fault recovery with the usage of PDR process and state synchronization. Till now, we were considering only coarse grained TMR architecture for protection of target system in our experiments. The scheme of processors protected by coarse grained TMR and the illustration of fault recovery strategy is shown in Figure 3.

Moreover, our goal is also to explore possibilities of fine grained TMR architecture used for protecting internal components within soft core processor.

A. *The state of the art methods*

The paper [2] describes methods for synchronization of faulty processors in coarse grained TMR architecture. The aim of authors was to research essential steps for synchronization in more details since this topic is insufficinelty addressed by researchers studying various FT methodologies and PDR. They proposed and evaluated four different methods for Xilinx PicoBlaze soft core processor which span from an implementation with minimal hardware overhead to completely hardware-based synchronization technique.

- Synchronization by reset – processors are brought into synchronized and known state of program execution by the system reset.
- Synchronization with shared memory – processors are synchronized by concurrent writing into TMR protected memory by all processors, followed by a concurrent reading of the data. Synchronization has to be triggered externally by program at the moment when no interrupt is executed.
- Synchronization with shared memory driven by interrupt – processors are synchronized by interrupt and it can be performed almost immediately after reconfiguration is finished. However, it requires hardware synchronization of the stack context.
- Complete hardware-based synchronization – processors are synchronized through hardware synchronization interface which allows copying of processor core registers with flags, stack pointe, the stack data and the scratchpad memory stored in block RAM (BRAM). Multiplexers and synchronization counters are implemented for each memory element enabling synchronization of one element in one cycle.

The evaluation of synchronization methods and experiments demonstrate significant increase in utilization FPGA resources and impact on the system frequency for synchronization methods exploiting hardware implementation and modifications inside processors [2].

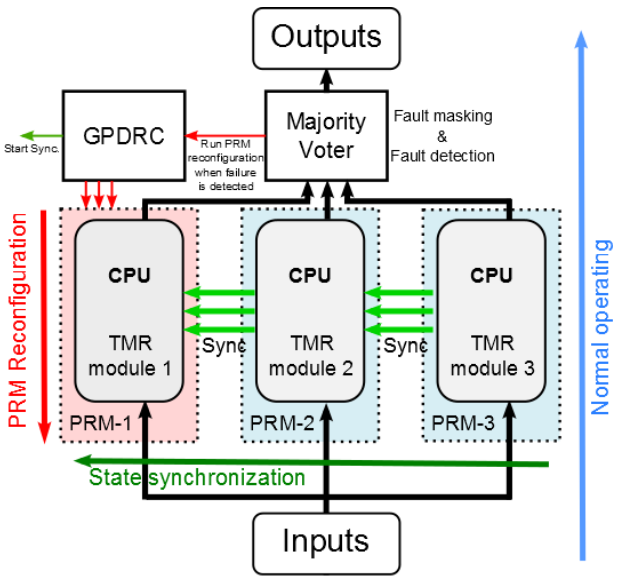


Figure 3: Recovery process for soft core processors protected by TMR architecture

B. *Design of soft core processor synchronization*

We evaluated several open-source soft core processors available for our research including LEON3, Plasma, ZPU and neo430. The neo430 was selected as the main platform for our experiments since it is a small, powerful and customizable 16-bit soft-core microcontroller, compatible to TI's MSP430 [6]. This microcontroller uses separated instruction and data memory and integrates high precision timer, watchdog timer, serial interface UART, GPIOs, Wishbone bus interface and internal bootloader.

Synchronization procedure for a soft core processor must synchronize all processor registers, program stack pointer and stack data. Moreover, instruction and data caches and program memory can be synchronized.

The scope of synchronization strategy depends on soft core processor complexity and application requirements for the synchronization itself and the recovery process.

V. PH.D. THESIS GOALS

The aim of our research is to propose new methodology for design and implementation of state synchronization procedure for reconfigurable FTS based on soft core processors protected by coarse grained or fine grained TMR architecture with respect to defined criteria for state synchronization procedure.

Recent results and goals satisfied during previous research were summarized in section II.A. The future goals of the research and Ph.D. thesis are as follows:

1. Implementation of CAN bus control system in neo430 soft core processor protected by coarse grained reconfigurable TMR architecture. Design state synchronization method, perform experiments and compare results with previous hardware implementation of the FT CAN bus control system.
2. Implementation of the robot controller algorithm [7] in neo430 soft core processor protected by fine grained reconfigurable TMR architecture. The design of state synchronization method and execution of the experiments. The motivation is also in further participation on collective research activities within Fault Tolerant Systems Design, Diagnostics and Testing group at Brno FIT. The robot controller is an integral part of the verification environment for evaluating impacts of faults in electro-mechanical systems.
3. Comparison and generalization of advantages and disadvantages of various state synchronization methods for soft core processor with respect to selected granularity of the TMR architecture based on performed experiments.
4. Evaluation of various synchronization methods for soft core processors against defined design criteria for synchronization method and execution (impact on the system function, speed, overhead).
5. Evaluation of reliability of unprotected soft core processor and soft core processor protected by coarse grained and fine grained TMR architecture, the system with and without reconfiguration and implementation of a state synchronization strategy.
6. Comparison of different ways for construction of synchronization circuit for digital system based on random logic and soft core processors.

VI. CONCLUSION AND FUTURE RESEARCH

In this paper, the present research related to development of new state synchronization methodology for FTS based on soft core processors was described.

The future work will be mainly focused on the implementation of various synchronization methods for soft core processors with respect to selected TMR architecture granularity, defined synchronization criteria and the evaluation of experiments and results with the aim to generalize gained knowledge for any soft core processor platform.

VII. ACKNOWLEDGEMENTS

This work was supported by The Ministry of Education, Youth and Sports from the National Program of Sustainability (NPU II); project IT4Innovations excellence in science - LQ1602. This work was also supported by Brno University of Technology under number FIT-S-14-2297 and by ARTEMIS JU under grant agreement no 641439 (ALMARVI).

REFERENCES

[1] Siegle, F.; Vladimirova, T.; Ilstad, J.; Emam, O.: Mitigation of Radiation Effects in SRAM-Based FPGAs for Space Applications. ACM Comput. Surv. 47, 2, Article 37, pp 1-34, January 2015. ISSN 0360-0300.

[2] Kretzschmar, U.; Gomez-Cornejo, J.; Astarloa, A.; Bidarte, U.; Del Ser, J.:Synchronization of faulty processors in coarse-grained TMR protected partially reconfigurable FPGA designs. Reliability Engineering & System Safety, Volume 151, 2016, pp.1-9. ISSN 0951-8320.

[3] Szurman, K.; Mičulka, L.; Kotásek, Z.: State Synchronization after Partial Reconfiguration of Fault Tolerant CAN Bus Control System. 17th Euromicro Conference on Digital Systems Design. Verona: IEEE Computer Society, 2014, pp. 704-707. ISBN 978-0-7695-5074-9.

[4] Szurman, K.; Mičulka, L.; Kotásek, Z.: Towards a State Synchronization Methodology for Recovery Process after Partial Reconfiguration of Fault Tolerant Systems. 9th IEEE International Conference on Computer Engineering and Systems. Cairo: IEEE Computer Society, 2014, pp. 231-236. ISBN 978-1-4799-6593-9.

[5] Mičulka, L.; Kotásek, Z.: Generic Partial Dynamic Reconfiguration Controller for Transient and Permanent Fault Mitigation in Fault Tolerant Systems Implemented Into FPGA. 17th IEEE Symposium on Design and Diagnostics of Electronic Circuits and Systems, Warszawa, PL, 2014, pp. 171–174, ISBN 978-0-7695-5074-9.

[6] Nolting, S.: The NEO430 Processor - A small, powerful and customizable open-source 16-bit soft-core microcontroller, compatible to TI's MSP430 ISA. https://opencores.org/project,neo430, available online, april 2017.

[7] Podivínský, J.; Čekan, O.; Lojda, J.; Kotásek, Z.: Verification of Robot Controller for Evaluating Impacts of Faults in Electro-mechanical Systems. Proceedings of the 19th Euromicro Conference on Digital Systems Design. Limassol: IEEE Computer Society, 2016, pp. 487-494. ISBN 978-1-5090-2817-7.

Hardvérový kernel pre systémy reálneho času

Lukáš Kohútka
2. ročník, denné štúdium
Školiteľ: Viera Stopjaková

Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave
Ilkovičova 3, 812 19 Bratislava
lukas.kohutka@stuba.sk

***Abstrakt*—Tento príspevok prezentuje nové výsledky dosiahnuté v rámci výskumu v oblasti operačných systémov reálneho času realizovaných na hardvérovej úrovni, získané počas druhého ročníka doktorandského štúdia. V tomto období došlo k zúženiu zámeru dizertačnej práce, boli zadefinované ciele dizertačnej práce a taktiež boli tieto ciele postupne plnené. Tento príspevok tiež obsahuje opis nových častí celkového riešenia spolu s novými publikáciami za dané obdobie.**

Kľúčové slová—systém reálneho času; hardvérová akcelerácia; koprocesor; architektúra; determizmus; FPGA; operačný systém

I. Úvod a Motivácia

S rastúcou hustotou integrácie integrovaných obvodov (IO) a systémov na čipe (SoC) sa postupne rozširujú možnosti ich využitia pre rozličné aplikácie, vrátane aplikácií určených pre systémy reálneho času. No napriek zvyšujúcej sa hustote integrácie, rast pracovnej frekvencie číslicových IO sa za posledných pár rokov výrazne spomalil, až takmer zastavil. Aby bolo možné aj naďalej zvyšovať výpočtový výkon počítačových systémov, tento nepriaznivý trend je kompenzovaný používaním viacjadrových procesorov, ktoré vykonávajú viacero inštancií programov paralelne. Prístup použitia viacjadrových procesorov je vhodným riešením najmä pre bežné systémy a dobre paralelizovateľné aplikácie, pretože s rastúcim počtom jadier sa znižuje priemerný čas potrebný na dokončenie paralelizovateľných častí programu. Avšak pre systémy reálneho času predstavuje tento prístup menší prínos [1].

Alternatívnym prístupom ako využiť zvyšujúce sa množstvo tranzistorov na čipe je hardvérová akcelerácia. Tá predstavuje hardvérovú realizáciu rozličných výpočtových algoritmov, ktoré sa realizujú zvyčajne softvérovo. Hardvérová realizácia môže znížiť časovú asymptotickú zložitosť vybraného algoritmu. Napríklad ak najlepšia známa softvérová realizácia daného algoritmu má lineárnu časovú zložitosť, tak hardvérová realizácia môže dosiahnuť až konštantnú časovú zložitosť. Konštantná časová zložitosť znamená, že bez ohľadu na množstvo dát v systéme, daná operácia alebo algoritmus trvá vždy rovnako dlhý, čiže konštantný časový úsek. Práve konštantná časová zložitosť je veľmi dôležitá pre systémy reálneho času, pretože prispieva k tomu, aby bol systém reálneho času viac deterministický. Deterministickosť je pre systémy reálneho času mimoriadne dôležitá vlastnosť [1].

Charakteristickou črtou systémov reálneho času je to, že obsahujú aspoň jednu úlohu, ktorú možno označiť za úlohu reálneho času. Úloha reálneho času je inštancia programu alebo jej časť (t.j. proces alebo vlákno), pričom takáto úloha musí byť dokončená najneskôr do stanoveného času. V opačnom prípade môže byť výsledok tejto úlohy považovaný za nepresný alebo dokonca úplne nepoužiteľný. Úlohy reálneho času je potrebné naplánovať v systéme takým spôsobom, aby bolo zabezpečené a zaručené ich splnenie v správnom čase. Čím je celý systém deterministickejší, tým ľahšie je možné tento cieľ dosiahnuť. Vysoká miera deterministickosti navyše umožňuje vytvárať komplexnejšie systémy reálneho času s väčším množstvom úloh [1].

Systémy reálneho času sa využívajú v mnohých oblastiach priemyslu ako napríklad letecký, vesmírny, automobilový, železničný, výrobný, energetický, chemický a ďalšie. Celkovo sú teda systémy reálneho času dosť rozšírené a preto má zmysel sa zaoberať tým, ako ich vylepšiť. V mnohých prípadoch sa využívaju systémy reálneho času na kritické aplikácie, kedy je nutné sa zaoberať aj spoľahlivosťou systému a jeho kritických úloh. Ak sa totiž nejaká kritická úloha nevykoná včas, môže to mať fatálne následky.

Hardvérová akcelerácia prináša pre systémy reálneho času nielen zrýchlenie systému, ale zároveň umožňuje zvýšiť mieru deterministickosti, spoľahlivosti a v neposlednom rade aj robustnosti celého systému, čiže možnosť disponovať väčším množstvom úloh a s tým súviacej širšej funkcionality systému.

II. Zámer dizertačnej práce

Zámerom dizertačnej práce, o ktorej je tento príspevok, je navrhnúť, implementovať a vyhodnotiť hardvérovú architektúru jadra operačného systému reálneho času. Takéto jadro by malo byť univerzálne, efektívne, modulárne, konfigurovateľné a prispôsobiteľné na základe potrieb konkrétnych aplikácií v rámci systémov reálneho času. Na základe tohto zámeru sme zadefinovali nasledovné rámcové ciele dizertačnej práce:

- Vyhodnotenie hardvérových zoraďovacích architektúr a ich efektívnosti, vrátane vlastných implementácií cez porovnanie výsledkov syntézy.
- Návrh modulu plánovania úloh reálneho času založeného na FPGA s vylepšenou flexibilitou a efektívnosťou.
- Návrh modulu správy pamäte reálneho času založeného na FPGA s vylepšenou flexibilitou a efektívnosťou.

- Návrh novej hardvérovej zoraďovacej architektúry vhodnej pre použitie v jadre operačného systému reálneho času.
- Návrh robustného a modulárneho jadra operačného systému reálneho času pozostávajúceho z hardvérového plánovania úloh, správy pamäte a novej zoraďovacej architektúry.
- Vyhodnotenie navrhnutých modulov ako samostatných celkov a celého jadra pozostávajúceho z týchto modulov.

Práca sa bude zaoberať nielen architektúrou jadra na systémovej úrovni, ale aj jednotlivými komponentmi platformy na úrovni RTL (úroveň medzi-registrových prenosov). To umožňuje optimalizáciu platformy na viacerých úrovniach abstrakcie – ako komponentov, tak aj ich vzájomného prepojenia. Najdôležitejším parametrom optimalizácie je miera deterministickosti, ktorá je v ideálnom prípade dosiahnutá konštantnou časovou zložitosťou operácií, ktoré má jadro poskytovať. Ďalšími parametrami optimalizácie sú štandardné parametre, na ktoré sa prihliada pri vývoji číslicových obvodov. Medzi tie patrí napríklad pracovná frekvencia systému, počet hodinových cyklov potrebný na vykonanie operácie, plocha čipu (pre FPGA množstvo spotrebovaných logických elementov, registrov a pamäte) a spotreba energie.

III. Systémová úroveň návrhu

Štandardné riešenie pre systémy reálneho času pozostáva z jedného alebo viacero jadier CPU, softvérovej implementácie operačného systému reálneho času a jednej alebo viacerých aplikácií. V našom prípade je softvérová implementácia jadra RTOS nahradená hardvérovým jadrom RTOS, ktoré môže byť implementované buď v hradlových poliach (FPGA) alebo vo forme zákazníckeho obvodu (ASIC), rovnako ako samotné CPU. Samotné jadro RTOS realizované na hardvérovej úrovni pozostáva z dvoch hlavných častí, ktorými sú: Správca úloh a Správca pamäte. Pritom obidve tieto časti používajú na realizáciu svojich operácií zoraďovanie dát a preto sme sa hlbšie zaoberali aj samotnými architektúrami pre zoraďovanie dát pre systémy reálneho času. Na základe vyššie uvedených informácií môžeme zhrnúť, že v našom prípade výsledný systém reálneho času disponuje týmito komponentmi:

- CPU
- Hardvérovo realizované jadro RTOS
  - Správca úloh
  - Správca pamäte
- Softvérovo realizované zvyšné časti RTOS
- Aplikácie

IV. Komponenty systému

A. CPU

V rámci dizertačnej práce bude použitý iba existujúci procesor. K dispozícii sú dva typy procesorov: hard-core CPU a soft-core CPU. Hard-core CPU je štandardný procesor realizovaný v ASIC technológii, pričom v rámci existujúcich vývojových dosiek sa jedná vo väčšine prípadov o ARM procesor. V prípade soft-core CPU sa jedná o IP jadro procesora, ktoré sa realizuje v technológii FPGA. Výsledné riešenie na konci tejto dizertačnej práce by malo byť použiteľné pre oba typy procesorov, pretože každý má iné výhody a nevýhody.

B. Správca úloh

Vývoj správy úloh bol vykonaný najmä v prvom ročníku doktorandského štúdia, ale v druhom ročníku bol tento správca úloh ešte vylepšený. Vyvinutý správca úloh je zameraný na preemptívne plánovanie úloh s konštantnou časovou zložitosťou, nezávislou od množstva úloh v systéme. V prvom ročníku sa podarilo vyvinúť viacero rôznych verzií plánovača úloh, ako pre jednojadrový procesor, tak aj pre dvojjadrový. Takýto plánovač bol založený na algoritmoch EDF (Earliest Deadline First) a FCFS (First-Come First-Served), rozšírených o možnosť odstrániť ľubovoľnú úlohu na základe jej ID. Plánovanie úloh sa teda v tomto prípade realizuje nielen na základe doby odozvy, do ktorej je potrebné dokončiť úlohy reálneho času, ale aj na základe priority. Úloha je buď hard real-time úloha, ktorá je naplánovaná podľa hodnoty „deadline“, alebo sa jedná o bežnú úlohu (teda nie real-time úloha), ktorá je naplánovaná podľa priority. Pritom je fixne stanovené, že všetky real-time úlohy sú uprednostňované pred bežnými úlohami. Vďaka tomu je možné pomerne jednoducho kombinovať v tom istom systéme úlohy reálneho času s bežnými úlohami, ktoré nemajú stanovenú dobu odozvy, ale prioritu. Celkovo možno skonštatovať, že tieto nové verzie plánovača úloh sú flexibilnejšie a časovo efektívnejšie v porovnaní s doterajšími existujúcimi riešeniami [2-6].

Pre dvojjadrové procesory boli taktiež navrhnuté dve techniky riešenia konfliktov, pričom pojem konflikt je vnímaný ako situácia, kedy viacero procesorových jadier chce použiť koprocesor v tom istom čase. Prvá metóda nazývaná *semaforová technika*, vyberá v prípade konfliktu víťazné jadro procesora, ktoré môže ihneď použiť daný koprocesor, zatiaľ čo ostatné jadra procesora sú zatiaľ pozastavené. Táto technika je pomerne jednoduchá a efektívna z hľadiska plochy na čipe. Druhá technika nazvaná ako *simultánne spracovanie* je založená na vnútornom rozšírení logiky v koprocesore takým spôsobom, aby bol koprocesor schopný prijímať a vykonávať inštrukcie od viacerých jadier procesora súčasne. Tým sa zamedzí vzniku konfliktov a tým pádom ani nedochádza k pozastavovaniu alebo oneskoreniu v procesore. Nevýhodou tohto prístupu je iba zložitejší návrh koprocesora a s tým spojená väčšia plocha čipu potrebná na realizáciu.

Navrhnuté verzie tohto koprocesora sú už zároveň implementované, verifikované metodológiou Universal Verification Methodology, zosyntetizované pre FPGA Altera Cyclone II, otestované funkčným BIST testom, a v neposlednom rade publikované vo viacerých príspevkoch na rôznych konferenciách [7-12].

V druhom ročníku štúdia bol správca úloh rozšírený o dve nové verzie plánovania úloh. Prvou novinkou je rozšírenie plánovača úloh o podporu pre 4-jadrové procesory. Na túto podporu bola použitá vyššie spomenutá technika zvaná semaforová technika. Vďaka podpore pre 4-jadrové CPU je možné dosiahnuť ešte väčší výkon pre koncové aplikácie,

vďaka čomu je možné vykonávať viac úloh reálneho času tak, aby boli splnené všetky časové ohraničenia jednotlivých úloh. O tomto vylepšení boli pripravené dva vedecké články, z toho jeden je akceptovaný [13] a druhý bol odoslaný na publikovanie v rámci IEEE konferencie SOCC 2017.

Druhým vylepšením, ktoré bolo dosiahnuté v tomto akademickom roku, je pridanie podpory pre garanciu, že budú dodržané všetky časové ohraničenia vybraných úloh reálneho času. Táto garancia je dosiahnutá realizáciou algoritmu GED (Guaranteed Earliest Deadline), ktorý je vhodný pre elimináciu potenciálnych domino efektov, ktoré majú nepriaznivý vplyv na soft real-time úlohy. Vďaka tomuto je algoritmus GED veľmi vhodný pre periodické soft real-time úlohy. Kombináciou algoritmov EDF, GED a FCFS je možné podporovať všetky tri hlavné typy úloh (hard RT, soft RT aj non-RT úlohy). Týmto spôsobom bola dosiahnutá vysoká miera flexibility navrhovaného plánovača úloh.

V budúcnosti je plánovane správcu úloh ešte ďalej rozšíriť o algoritmus RED, ktorý poskytuje optimálny plán úloh pre komplexnejšiu množinu úloh rôzneho typu. Taktiež sa skombinujú riešenia pre viacjadrové systémy (zatiaľ vyvinuté len pre EDF a FCFS) s realizáciou algoritmov GED a RED. V pláne práce je tiež vytvorenie automatickej správy periodických, prípadne aj sporadických úloh, keďže momentálne je správca úloh zameraný len na aperiodické úlohy, pričom periodickosť úloh musí byť zatiaľ zabezpečená softvérovo.

C. Správca pamäte

Pod pojmom správa pamäte sa myslí najmä dynamická správa pamäte, teda alokácia a dealokácia blokov pamäte rozličnej veľkosti, podobne ako je tomu v prípade funkcií jazyka C: *malloc* a *free*.

Aj keď už existuje aj realizácia dynamickej správy pamäte pre systémy reálneho času, táto realizácia je žiaľ iba softvérová a pomerne heuristická, a teda málo univerzálna a málo deterministická. Naviac štatistické výsledky ukazujú, že využitie pamäte je len na úrovni 75% a menej v dôsledku veľmi vysokej fragmentácie pamäte, čo je neefektívne a ťažko škálovateľné riešenie [14].

Riešením by mohol byť koprocesor, ktorý by realizoval existujúci algoritmus známy ako *worst fit*. Tento algoritmus bol zvolený z toho dôvodu, že jediná zložitejšia operácia, ktorú obsahuje je zoraďovanie blokov pamäte podľa jej veľkosti. Keďže operáciu zoraďovania sa podarilo implementovať v konštantnom čase už pre plánovanie úloh, je zrejmé, že rovnako aj tento algoritmus je možné realizovať tak, aby mal konštantnú časovú zložitosť, a teda aby bol maximálne deterministický. Nevýhodou tohto algoritmu je potenciálna náchylnosť na externú fragmentáciu, ktorá znižuje využitie pamäte. Na druhej strane v prípadoch, keď aplikácia používa podobne veľké bloky pamäte, dosahuje tento algoritmus najlepšiu mieru efektivity využitia pamäte.

V rámci tohto ročníka bola vyvinutá testovacia platforma pre vyhodnotenie rôznych prístupov správy pamäte v rámci bakalárskej práce pod vedením doktoranda. Výsledky boli publikované a prezentované na konferencii IIT.SRC 2017. Vďaka tejto testovacej platforme bude možné jednoduchšie porovnať nového, hardvérovo-akcelerovaného správcu pamäte voči existujúcim riešeniam [15].

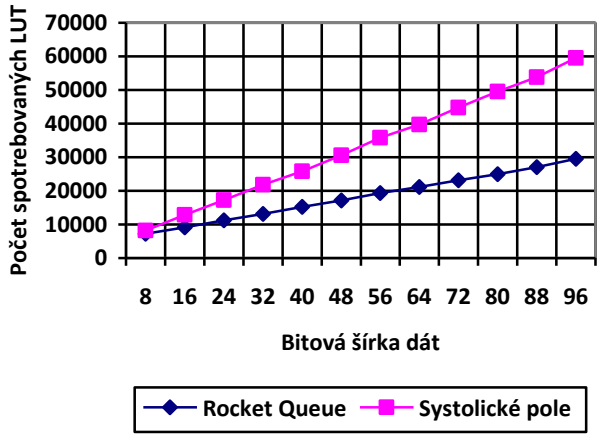
D. Zoraďovacia architektúra

Podstatnú časť druhého ročníka tvoril výskum a vývoj zameraný na vytvorenie novej zoraďovacej architektúry, ktorá by bola vhodnejšia pre správu úloh aj pamäte v systémoch reálneho času v porovnaní s existujúcimi architektúrami. Výsledkom tejto práce je nová architektúra, ktorú sme nazvali Rocket Queue. Táto architektúra podporuje vloženie a zatriedenie nového prvku, zmazanie ľubovoľného existujúceho prvku a čítanie prvku s minimálnou alebo maximálnou hodnotou. Prínos Rocket Queue architektúry v porovnaní s existujúcimi architektúrami spočíva v šetrení až 40% logických elementov, ktoré je potrebné použiť na realizáciu danej architektúry. Vďaka tomu je možné buď zvýšiť kapacitu zoraďovacej štruktúry alebo znížiť spotrebu plochy na čipe, spotrebu energie a tiež zvýšiť priemernú dobu bezporuchovej prevádzky (t.j. spoľahlivosť) [16, 17].

Výsledky tejto práce boli prezentované na medzinárodných konferenciách, pričom na konferencii MECO 2017 bola autorom práce udelená cena The Best Paper za najlepší článok na konferencii [17]. Okrem toho bola autorom ponúknutá možnosť publikovať rozšírené výsledky v dvoch vedeckých časopisoch (články sú v štádiu prípravy).

Architektúra Rocket Queue je efektívnejšia z hľadiska spotreby výpočtových zdrojov najmä z toho dôvodu, že táto architektúra nevyužíva jeden komparátor na každú bunku zoraďovacej štruktúry, ale iba jeden komparátor na jednu úroveň štruktúry. Práve výraznou redukciou počtu komparátorov je dosiahnutá spomínaná úspora výpočtových zdrojov, pretože práve komparátor bol identifikovaný ako najzložitejší blok v rámci existujúcich zoraďovacích architektúr. Použitím iba jedného komparátora pre niekoľko buniek je potrebné zabezpečiť spoľahlivé zdieľanie komparátora v rámci týchto buniek tak, aby kompárator potrebovala použiť vždy len jedna bunka. Preto sú bunky zaradené do úrovní (levelov), v rámci ktorých je komparátor bezpečne a spoľahlivo zdieľaný medzi bunkami danej úrovne.

Na obrázku č. 1 je znázornená spotreba logických elementov (LUT) vzhľadom na bitovú šírku zoraďovaných údajov pri použití architektúry Rocket Queue a architektúry založenej na systolických poliach. Systolické pole je považované za najvhodnejšie doterajšie existujúce riešenie. Syntéza sme vykonali pre FPGA Altera Cyclone V, pre pracovnú frekvenciu 125 MHz. Z porovnania výsledkov pre obe architektúry vyplýva, že zvyšovaním počtu bitov pre zoraďované dáta sa zvyšuje percentuálna miera úspory potrebnej logiky, čím sa zvyšuje prínos použitia novej architektúry Rocket Queue. V prípade 32-bitových údajov je na implementáciu potrebných až o 39% menej LUT a v prípade 64-bitových údajov až o 47% menej LUT.



Obr. 1 – Porovnanie architektúr vzhľadom na počet logických elementov voči bitovej šírke dát

Ďalším krokom bude aplikácia vyvinutej zoraďovacej architektúry do modulov správa úloh, ako aj správa pamäte, aby sa tak využil prínos tejto architektúry pre vylepšenie týchto modulov. Vďaka tomu budú moduly správy úloh aj správy pamäte výrazne efektívnejšie ako existujúce riešenia vzhľadom na množstvo logiky (t.j. LUT a registre pre FPGA), ktorá je potrebná na realizáciu daných modulov.

E. Softvérovo realizované zvyšné časti RTOS

Táto časť nebude vyvíjaná. Plánujeme použiť existujúci kód operačného systému FreeRTOS, v ktorom sa namiesto správy úloh a správy pamäte použijú špecifické inštrukcie vyvinutých modulov vo forme koprocesorov. Táto časť práce už bola predspracovaná vo forme integrácie softvérovo implementovaných algoritmov (konkrétne EDF, GED pre správu úloh a algoritmy worst-fit a best-fit pre správu dynamickej pamäte) do FreeRTOS bez použitia hardvérovej akcelerácie na FPGA.

F. Aplikácie

Táto časť systému zatiaľ nie je rozpracovaná. Úlohou aplikačnej časti má byť primárne overenie kompletného riešenia a jeho škálovateľnosti vzhľadom na počet úloh, rôznorodosť typov úloh, zložitosť aplikácie a počet jadier použitého procesora.

V. ZÁVER

Bola navrhnutá nová architektúra hardvérovo akcelerovaného operačného systému ako platforma pre systémy reálneho času. Taktiež boli opísané aj jednotlivé komponenty navrhovaného riešenia systému, dosiahnutého v rámci druhého ročníka štúdia. Prácu sme zamerali na realizáciu správy úloh a správy pamäte. V rámci druhého ročníka bola vyvinutá hlavne nová architektúra pre zoraďovanie dát a taktiež bol vylepšený plánovač úloh. Riešenia boli odsimulované, implementované a otestované, a dosiahnuté výsledky boli publikované na medzinárodných konferenciách.

Z hľadiska plnenia cieľov dizertačnej práce môžeme skonštatovať, že doteraz sa podarilo splniť tri zo šiestich cieľov. Zostáva ešte zrealizovať správcu pamäte, následne spolu s vyvinutým správcom úloh integrovať tieto moduly do FreeRTOS, vybrať vývojovú dosku s CPU a FPGA, a nakoniec implementovať celý systém a otestovať ho na danej platforme.

POĎAKOVANIE

Táto práca bola podporená projektom APVV-15-0245 a VEGA 1/0905/17.

REFERENCIE

[1] G.C. Buttazzo, “Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications,” 2011.

[2] Y. Tang, and N.W. Bergmann, “A Hardware Scheduler Based on Task Queues for FPGA-Based Embedded Real-Time Systems,” IEEE Transactions on Computers, 2015.

[3] J. Starner, J. Adomat, J. Furunas, and L. Lindh, “Real-Time Scheduling Co-Processor in Hardware for Single and Multiprocessor Systems,” Proceedings of the EUROMICRO Conference, 1996.

[4] C. Ferreira, and A.S.R. Oliveira, “Hardware Co-Processor for the OReK Real-Time Executive,” 2010.

[5] S.E. Ong, and S.C. Lee, “SEOS: Hardware Implementation of Real-Time Operating System for Adaptability,” Computing and Networking (CANDAR), 2013 First International Symposium, 2013.

[6] K. Kim, D. Kim, and Ch. Park, “Real-Time Scheduling in Heterogeneous Dual-core Architectures,” Proceedings of the 12th International Conference on Parallel and Distributed Systems, 2006

[7] L. Kohutka, “Hardware Task Scheduling in Real-Time Systems,” Proceedings of IIT.SRC, 2015.

[8] L. Kohutka, M. Vojtko, and T. Krajcovic, “Hardware Accelerated Scheduling in Real-Time Systems,” Engineering of Computer Based Systems Eastern European Regional Conference, 2015.

[9] L. Kohutka, V. Stopjakova, “Hardware-Accelerated Task Scheduling in Real-Time Systems: Deadline Based Coprocessor for Dual-Core CPUs,” International Symposium on Design and Diagnostics of Electronic Circuits and Systems, 2016.

[10] L. Kohutka, V. Stopjakova, “Hardware Accelerated Task Scheduling in Real-Time Systems,” Conference on Advances in Electronic and Photonic Technologies, 2016.

[11] L. Kohutka, V. Stopjakova, “Task Scheduler for Dual-Core Real-Time Systems,” International Conference on Mixed Design of Integrated Circuits and Systems, 2016.

[12] L. Kohutka, V. Stopjakova, “Improved Task Scheduler for Dual-Core Real-Time Systems,” Euromicro Conference on Digital System Design, 2016.

[13] L. Kohutka, V. Stopjakova, “Real-Time Task Scheduler for Quad-Core CPUs,” Conference on Advances in Electronic and Photonic Technologies, 2017.

[14] M. Masmano, I. Ripoll, A. Crespo, J. Real, “TLSF: a new dynamic memory allocator for real-time systems,” Real-Time Systems 2004. ECRTS 2004. Proceedings. 16th Euromicro Conference on, pp. 79-88, 30 June-2 July 2004.

[15] D. Beňo “Testing Platform for Memory Management in Real-Time Systems,” Proceedings of IIT.SRC, 2015.

[16] L. Kohutka, V. Stopjakova, “Rocket Queue: New Data Sorting Architecture for Real-Time Systems,” International Symposium on Design and Diagnostics of Electronic Circuits and Systems, 2017.

[17] L. Kohutka, V. Stopjakova, “A New Efficient Sorting Architecture for Real-Time Systems,” Meditteranean Conference on Embedded Computing, 2017.

Automatizace návrhu systémů odolných proti poruchám pomocí vysokoúrovňové syntézy

Jakub Lojda
2. ročník, prezenční studium,
Školitel: Doc. Ing. Zdeněk Kotásek, CSc.

Fakulta informačních technologií Vysokého učení technického v Brně
Božetěchova 1/2, 612 66 Brno, Czech Republic
Email: {ilojda, kotasek}@fit.vutbr.cz

***Abstrakt*—S vyšší úrovní integrace přichází výzva maximálně využít dostupnou kapacitu na čipu. Efektivní využití zdrojů je námětem pro vznik nových metod návrhu číslicových obvodů. Jednou z těchto metod je tzv. *vysokoúrovňová syntéza* (HLS), která je využitelná ve spojení s hradlovými poli *Field Programmable Gate Array* (FPGA). Obecným cílem našeho výzkumu je nalézt metodu automatického návrhu systémů odolných proti poruchám. Tento článek shrnuje první kroky výzkumu, které se zabývají možností vkládání redundance s pomocí HLS a zejména vyhodnocením tohoto přístupu.**

***Klíčová slova*—Automatizace návrhu, HLS, vysokoúrovňová syntéza, Catapult C, odolnost proti poruchám, systém odolný proti poruchám.**

I. Úvod

Vyšší úroveň integrace vede na možnost realizovat složitější obvody uvnitř čipů, pro návrháře je k dispozici více zdrojů, které je ale stále těžší efektivně využít. Efektivní využití těchto zdrojů je námětem pro vznik nových metod návrhu obvodů. Jednou z těchto metod je tzv. *vysokoúrovňová syntéza*, angl. *High Level Synthesis* (HLS). K realizaci výsledných obvodů je pak často využívána technologie hradlových polí FPGA. U těchto obvodů však vede rostoucí úroveň integrace na čipu ke zvyšování náchylnosti k tzv. jevům *Single Event Upset* (SEU). Obecným cílem našeho výzkumu je nalézt metodiku pro automatizovanou volbu zajištění *odolnosti proti poruchám* (OPP), která by byla schopna pokrýt jak nové metody návrhu (HLS), tak konvenční přístupy.

Tento článek popisuje konkrétní snahy o zavádění OPP do systémů tvořených pomocí HLS. Text je organizován do šesti částí, Sekce II a Sekce III se věnují souvisejícím výzkumům. Sekce IV popisuje současný stav práce na metodě zavádění spolehlivosti do systémů založených na HLS, Sekce V uvádí vyhodnocení představené metody a Sekce VI uvádí obecné cíle disertace a závěrečné zhodnocení.

II. Související práce: Automatizace návrhu systémů OPP

Autoři příspěvku [7] shrnují náležitosti, které by nástroj pro automatizaci návrhu systémů OPP měl mít. Dále zvažují fakt, že dosavadní pokusy o automatizaci byly vedeny na nižších úrovních abstrakce. Autoři poukazují, že i na vyšších úrovních abstrakce je možné dosáhnout stejně kvalitních výsledků s dodatečnými výhodami v podobě snazší práce návrháře s popisem na úrovni HDL. Další shrnutí tohoto přístupu nabízí hlavní autor předchozí publikace v příspěvku [6]. Hradlová pole FPGA, na která je náš výzkum primárně orientován, je možno dle místa projevu SEU rozdělit do třech úrovní [14]: 1) *vrstva konfigurační paměti*; 2) *vrstva blokových pamětí*; a 3) *vrstva uživatelské logiky*. Proto i následující text je rozdělen tak, aby korespondoval s jednotlivými úrovněmi.

A. Konfigurační paměť

Jedním ze způsobů ošetření konfigurační paměti je tzv. *čištění paměti*, neboli *memory scrubbing*. Autoři příspěvku [13] uvádějí metodu spouštění *scrubbingu* na základě predikce poruch. Jinou variantu představuje tzv. *Frame-Level Redundancy* (FLR) *scrubbing* [19], který je vhodný pro *obnovu* systému založeného na architektuře TMR s jemnou granularitou. Příspěvek [12] uvádí metodu pro snížení *Mean Time To Reapir* (MTTR). Metoda překládá *chybovou signaturu* na startovní rámec, od kterého je při detekci poruchy zahájen *scrubbing*.

B. Paměťové elementy

V [20] je prezentována metodika návrhu číslicových obvodů s ohledem na náchylnost k přechodným poruchám v paměťových elementech. Autoři vyvíjejí nástroj zvaný *FT-PRO*, který tuto metodiku automatizuje. Aplikační poznámka [11] uvádí realizaci *čištění paměti* pro blokové paměti BRAM.

C. Uživatelská logika

Metoda pro modifikaci sekvenčních obvodů tak, aby se staly *samočinně kontrolované*, je prezentována v [10]. Příspěvek [1] se zaměřuje na tvorbu nástroje pro vložení redundance na principu modifikace již existujícího VHDL kódu dle předložených specifikací. Nástroj prohledávající stavový prostor možných konfigurací OPP prezentují autoři příspěvku [16].

III. Související práce: HLS

Následující část textu je věnována popisu aktuálních metod zavádějících OPP do systémů syntetizovaných pomocí HLS, což je také aktuálním předmětem zaměření našeho výzkumu.

A. Datové cesty

V příspěvcích [3] a [5] se autoři zabývají poruchami datových cest, jež se vyskytují přechodně a trvají několik taktů. Heuristika uváděná v [17] umožňuje volit kompromis mezi latencí a mírou redundance výsledného systému. Na zvolené testovací sadě bylo dosaženo 18% až 49% úspory zdrojů při 70% pokrytí poruch a zdvojnásobení přípustné *latence*.

B. Řadič datových cest

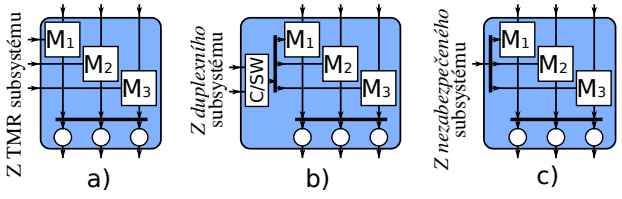
Pro dosažení vysoké míry OPP je mimo zabezpečení datových cest vhodné věnovat dostatečnou pozornost také řadiči, kterým jsou datové cesty ovládány. V případě, že datové cesty podporují detekci přechodných poruch, je možné využít koncept představený v [4].

IV. Současný stav výzkumu

Aktuální snahy spočívají ve vyhodnocení alternativní metody pro vkládání redundance do systémů navržených pomocí HLS. Protože k modifikacím zde dochází na úrovni algoritmu, který je vstupem do vysokoúrovňové syntézy, přístup zachovává výhody, kterými HLS disponuje. Je nutné zdůraznit, že tato metoda nezvažuje zabezpečení řadiče datových cest a příslušejících ostatních prvků, které nejsou přímo ovlivnitelné vstupním popisem algoritmu. První nástin naší metody byl publikován v [8].

Při výzkumu využíváme komerčně dostupný nástroj HLS *Catapult C* [9], který je schopen zpracovat jazyk C (příp. C++) a na svém výstupu poskytuje popis obvodu v jazyce VHDL vyjádřením na úrovni RTL. V algoritmu popsaném v jazyce C++ můžeme identifikovat celkem tři typy modifikací, jsou to: 1) *datové typy* (*úložiště*); 2) *aritmetické a logické operace*; a 3) *příkazy pro řízení toku*. Tato fáze práce je zaměřena na výzkum dosažení OPP pomocí změny *úložišť* a *operací*. Metoda využití nových *datových typů* (DT), které nazýváme *redundantní datové typy* (RDT), bude vyhodnocena na známém principu TMR. Koncept RDT využívá již existující DT (které označujeme jako *originální* DT). Každý RDT vyjadřuje jednu metodu OPP (např. TMR) a může zastřešovat operace nad libovolným *originálním* DT. Tento způsob umožňuje měnit sémantiku operací příslušejících k danému RDT a tak implementovat OPP do těchto operací. Modifikace algoritmu pak spočívá v pouhých záměnách DT za RDT implementující konkrétní metody OPP.

U *binárních* operací může docházet ke komunikaci odlišných subsystémů, na které je aplikován odlišný druh zajištění OPP, může tak docházet k operacím s těmito kombinacemi DT a RDT: **a) intra-DT** operace (RDT vs. RDT zajišťující totožnou metodu OPP); **b) inter-DT** (RDT vs. RDT zajišťující odlišnou metodu OPP); a **c) original-DT** operace (RDT vs. jeho *originální* DT). Příklady možných kombinací pro systém TMR shrnuje Obrázek 1.



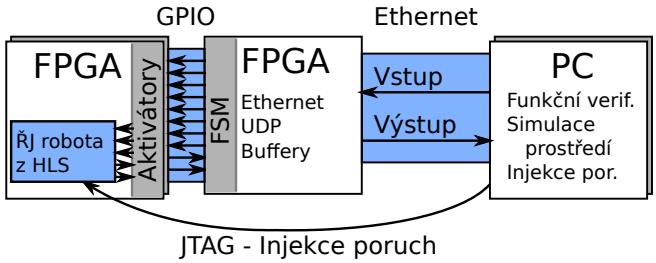
Obrázek 1. Příklad situací, které mohou nastat mezi operacemi v HW pro systém TMR v případě (a) operací *intra-DT*; (b) operací *inter-DT* (v tomto příkladě subsystém TMR komunikuje se *zdvojeným* (*duplexním*) subsystémem); a (c) operace s *originálním* DT.

V. Případová studie: Řadič robota

Vyhodnocení parametrů prezentovaných principů představuje případová studie na řadiči robota pro navigaci v bludišti.

A. Verifikační prostředí s řadičem robota

Pro zkoumání vlastností výsledných implementací bylo využito verifikační prostředí pro vyhodnocení vlivu poruch na zkoumanou jednotku [15]. Verifikační prostředí je vybudováno dle metodiky *Universal Verification Methodology* (UVM) a je zaměřeno na vyhodnocení *elektromechanických* aplikací. Verifikační prostředí je složeno ze dvou částí: 1) elektronického přípravku (*kit* s FPGA, do kterého jsou injektovány poruchy, zde *ML506* založený na FPGA řady *Virtex 5*); 2) mechanického prostředí, které je simulováno v osobním počítači (zde aplikace *Player/Stage* [2]), implementace UVM a injektoru poruch [18], rovněž běžících na PC. Schéma experimentálního prostředí ukazuje Obrázek 2.



Obrázek 2. Struktura experimentálního verifikačního prostředí.

Za pomoci injektoru poruch je možno cíleně vkládat poruchy do využitých bitů *Look-up* tabulek (LUT) a to zároveň pouze do těch, které jsou součástí specifikovaného bloku, tj. zde do verifikovaného obvodu.

B. Volba nastavení parametrů HLS

Protože nástroje HLS umožňují nastavit velké množství parametrů syntézy, bylo zapotřebí identifikovat vhodná nastavení, se kterými bude provedeno vyhodnocení. Jedním z významných parametrů u akceleračních technik v HLS je tzv. *iniciační interval*, jenž značí počet taktů mezi spouštěním jednotlivých cyklů smyčky (může dojít k překryvu běhů). *Úrovní paralelního výpočtu* rozumíme počet souběžně vykonávaných cyklů smyčky. Pro experimenty byla stanovena celkem čtyři různá nastavení HLS: **1) noopt-area** – HLS ve výchozím nastavení, cílem optimalizace byla plocha na čipu; **2) noopt-latency** – HLS ve výchozím nastavení, cílem optimalizace byla *latence*; **3) pipeline1-area** – celý obvod zřetězen při *iniciačním intervalu* jedna, cílem optimalizace plocha na čipu; **4) unroll2-area** – celý obvod rozbalen s *úrovní paralelního výpočtu* dvě, cílem optimalizace plocha na čipu.

C. Vliv nastavení HLS na náchylnost k SEU

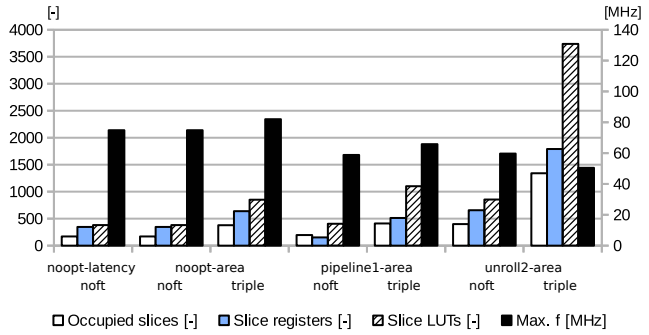
Algoritmus řadiče robota popsaného v jazyce C++ byl syntetizován pomocí HLS se čtyřmi popsanými nastaveními. Takto vznikly čtyři různé implementace řadiče robota, jejichž podrobnější popis z hlediska spotřebovaných zdrojů po syntéze nástrojem *Xilinx ISE* uvádí řádky označené *noft* v Tabulce I. Spotřebované zdroje shrnuje rovněž graf na Obrázku 3. Při bližším ohledání bylo zjištěno, že jednotky *noopt-area* a *noopt-latency* vedly na totožný VHDL kód, proto byla sada nastavení *noopt-latency* z dalších experimentů vyjmuta

Zbývající tři jednotky byly následně vyhodnoceny na náchylnost k poruchám. S každou *řídicí jednotkou* (ŘJ) bylo

Tabulka I
SPOTŘEBOVANÉ ZDROJE PRO KAŽDOU VERZI ŘJ ROBOTA.

Verze		Obsaz. *slices* [−]	*Slice reg.* [−]	*Slice LUTs* [−]	Max. frekv. [MHz]	bity LUT [*b*]
noopt-latency	*noft*	170	346	381	74.85	19392
noopt-area	*noft*	170	346	381	74.85	19392
	triple	378	638	851	82.01	48704
	TMR	546	1038	1143	74.85	58176
pipeline1-area	*noft*	196	152	405	58.82	21952
	triple	411	512	1101	65.81	67264
	TMR	540	456	1215	58.82	65856
unroll2-area	*noft*	399	656	854	59.70	48704
	triple	1341	1791	3738	50.48	224256
	TMR	1224	1968	2562	59.70	146112

provedeno 1000 verifikačních běhů, všechna vyhodnocení probíhala na totožné mapě bludiště. Vyhodnocení probíhalo následovně: 1) ŘJ robota byla uvedena do výchozího stavu; 2) do LUT tabulek obsazených ŘJ robota byla s *uniformním* rozložením pravděpodobnosti vložena jednonásobná porucha; 3) byla sledována schopnost ŘJ nalézt cílovou pozici.



Obrázek 3. Porovnání spotřebovaných zdrojů pro každou z verzí řadiče robota.

Výsledky získané při injekci poruch v tomto kroku popisuje ve sloupcích označených *noft* Tabulka II. Vrchní část tabulky informuje o celkovém stavu elektroniky, tj. zda došlo k odlišnostem mezi vzorovou implementací a implementací s injektovanou poruchou. Spodní část tabulky informuje o počtu případů, kdy robot nenalezl cíl. Tabulka II dále informuje o případech, kdy došlo ke kolizi robota se stěnou příp. kolikrát robot sice dorazil do cíle, ale cesta byla odlišná od vzorové.

Lze konstatovat, že jednonásobné poruchy se nejvíce projevovaly ve variantě *noopt-area*, která nezahrnuje bližší nastavení optimalizací v rámci HLS. Z výsledků však není jisté, zda rozdíly v citlivosti na jevy SEU nejsou způsobeny odlišnou plochou zabranou na čipu, protože počet verifikačních běhů byl shodný pro všechny implementace ŘJ. Z tohoto důvodu by bylo zajímavé jako součást budoucí práce provést rozsáhlejší testování výsledných designů.

Tabulka II
PŘEHLED DOPADU INJEKTOVANÝCH PORUCH PRO KAŽDOU ZE ZVOLENÝCH JEDNOTEK PŘI POČTU 1000 VERIFIKAČNÍCH BĚHŮ.

Dopad poruchy	*noopt-area*		*pipeline1-area*		*unroll2-area*	
	noft	*triple*	*noft*	*triple*	*noft*	*triple*
ŘJ OK [−]	949	982	967	996	979	995
ŘJ selhala [−]	51	18	33	4	21	5
Cíl nenalezen [−]	50	16	32	4	19	4
Kolize [−]	4	2	5	1	4	1
Cíl nalezen jinou cestou [−]	1	2	1	0	2	1

D. Vyhodnocení metody redundantních datových typů

Pro úvodní experiment byla zvolena strategie zabezpečit všechna *úložiště* a *operace* v algoritmu jednotky pomocí metody *ztrojení*. Využitý RDT, nazvaný *triple*, zahrnuje kombinaci *prostorové redundance* v případě replikace HW nástrojem HLS a *časové redundance* v případě optimalizací a mapování některých operací do shodné funkční jednotky. K vyhodnocení byla využita implementace zahrnující ztrojení paměťového elementu a všech operací nad daným RDT. Provedení jedné operace nad RDT zahrnuje spuštění operace nad třemi instancemi *orig*. DT a následné určení výsledku pomocí *majoritní funkce*. Jednotky, v nichž jsou všechna úložiště a operace replikovány na úrovni algoritmu pomocí ruční záměny DT na RDT, byly syntetizovány s vybranými množinami nastavení pro HLS a opět podrobeny popsané metodě vyhodnocení. Tabulka I uvádí rovněž spotřebované zdroje, řádky značené TMR ukazují zdroje jednotek, jež byly syntetizovány ztrojené na úrovni celých ŘJ. Vliv SEU ukazuje Tabulka II.

Závěrem této etapy experimentů lze konstatovat, že nejlepších výsledků dosahuje metoda v kombinaci se *zřetězením*, tj. nastavením HLS *pipeline1-area*, které proto bude využito v následující etapě. Na rozdíl od předchozí etapy nelze v tomto případě pozorovat přímou korelaci velikosti designu s výslednou náchylností k SEU, jako tomu je u jednotek *noft*.

E. Vliv zabezpečení jednotlivých operací

V poslední fázi experimentů byl zkoumán vliv aplikace *ztrojení* jednotlivých úložišť a operací v algoritmu na výslednou celkovou OPP. Úložiště a příslušející operace algoritmu ŘJ robota byly rozděleny do sedmi množin, jež byly označeny celými čísly $1-7$. Přehled zastoupení typů operací v jednotlivých množinách uvádí Tabulka III. Bylo sestaveno sedm korespondujících implementací ŘJ, v každé implementaci byla metoda *ztrojení* aplikována na danou množinu *operací* a *úložišť*. Syntéza byla provedena pomocí sady nastavení *pipeline1-area*, důvodem této volby je vysoká citlivost na zabezpečení prezentovanou metodou. Vyhodnocení je tak možné provést s vyšším rozlišením za konstantního počtu verifikačních běhů v případě, že je podstatný především procentuální rozdíl mezi jednotlivými implementacemi. Sloupec *Ref.* uvádí referenční parametry ŘJ *noft* syntetizované s nastavením *pipeline1-area*.

Dle výsledků 2000 verifikačních běhů lze konstatovat, že částečně zabezpečené jednotky, v nichž byly zabezpečeny množiny operací 2, 5 a 7, dosáhly nejvyšší úrovně OPP. V ostatních případech zabírají ŘJ dokonce menší plochu na čipu, přičemž parametry OPP ŘJ 1, 3 a 6 jsou stále lepší nežli v případě *referenční* jednotky. Věříme, že toto chování souvisí s velikostí řetězených bloků, jež je automaticky volena nástrojem HLS. Je však potřeba dalšího výzkumu pro ověření tohoto předpokladu, protože vzhledem ke snížení spotřebovaných zdrojů je získaná spolehlivost významná.

VI. CÍLE DISERTAČNÍ PRÁCE A ZÁVĚR

Hlavním cílem disertační práce je dospět k obecné metodice pro automatizaci návrhu systémů OPP, pro jeho dosažení byly stanoveny tři hlavní podcíle: 1) vybudovat prostředky pro automatické vkládání redundance do zvoleného jazyka

Tabulka III
EXPERIMENTÁLNÍ VYHODNOCENÍ SPOTŘEBY ZDROJŮ, ZABEZPEČENÝCH OPERACÍ A ÚLOŽIŠŤ A ZÍSKANÉ OPP.

Verze ŘJ	Ref.	1	2	3	4	5	6	7
Bity LUT [b]	21952	17408	55744	12800	15744	47552	15872	35840
Slices [–]	196	147	370	135	165	379	147	250
Selhání [‰]	33.0	27.0	13.5	30.5	37.5	15.5	29.5	17.0
RDT *unární*	0	0	7	22	4	4	2	2
ops. *binární*	0	6	7	32	7	9	5	2
[–] *ternární*	0	0	0	0	0	0	0	0
RDT *inter-DT*	0	0	0	0	0	0	0	0
ops. *intra-DT*	0	0	0	30	4	0	0	2
[–] *orig.-DT*	0	6	14	24	7	13	7	2
Zlepšení OPP [%]	–	18.2	59.1	7.6	-13.6	53.0	10.6	48.5
Prostorová režie [%]	–	-25.0	88.8	-31.1	-15.8	93.4	-25.0	27.6

(jazyků); 2) navrhnout metodu modifikace systému *neodolného* na systém *odolný* – hledat prostor pro zlepšení automatické volby zabezpečení ve vztahu k vlastnostem dané komponenty (příp. části algoritmu) při zohlednění optimalizačních parametrů; 3) zobecnit navržené postupy, aby byly použitelné pro různé jazyky využívané k popisu obvodové realizace.

V rámci podcíle 1) byla zvolena varianta HLS v kombinaci s popisem algoritmu v jazyce C++. Snahou tohoto přístupu je pochopit principy automatizace návrhu systémů OPP na jednodušších případech a poté snaha tyto principy zobecnit a definovat dostatečně obecný aparát, který by umožnil aplikaci těchto metodik automatizovat. K tomuto účelu byla navržena metoda vkládání informační redundance a záložních technických prostředků s využitím speciálně navržených RDT. Podúkol 2) má za cíl využít prostředky pro automatické vkládání redundance za účelem automatizace návrhu. Tyto prostředky umožní částečně abstrahovat metody OPP od jejich konkrétní aplikace. Metoda tedy bude schopna pracovat se systémem již na vyšší úrovni abstrakce, tj. bude rozhodovat na který blok (funkci) v systému aplikovat kterou metodu OPP. Výstupem analýzy bude rovněž granularita aplikace.

Z vyhodnocení metody lze učinit následující závěry: 1) použití popisované metody vkládání OPP se nejvíce projevilo u nastavení *pipeline1-area*; 2) odlišné důležitosti jednotlivých operací v algoritmu lze identifikovat aplikací OPP na tyto operace a následným vyhodnocením celkové spolehlivosti. Jako část budoucí práce by mohlo být zajímavé ověřit důvod fenoménu, jenž se ve zřetězeném designu projevuje u některých operací snížením spotřebovaných zdrojů za současného zvýšení spolehlivosti.

PODĚKOVÁNÍ

Tato práce byla podporována Ministerstvem školství, mládeže a tělovýchovy z Národního programu udržitelnosti (NPU II); projektu IT4Innovations excellence in science – LQ1602. Tato činnost byla rovněž podporována projekty řešenými na VUT v Brně pod číslem FIT-S-14-2297 a ARTEMIS JU pod grantem číslo 641439 (ALMARVI).

REFERENCE

[1] Entrena, L.; Lopez, C.; Olias, E.: Automatic insertion of fault-tolerant structures at the RT level. *Proceedings Seventh International On-Line Testing Workshop*, 2001, s. 48–50, doi:10.1109/OLT.2001.937817.

[2] Gerkey, B.; Vaughan, R.; Howard, A.; aj.: The Player/Stage Project. *přístupné z http://playerstage.sourceforge.net*, 2003.

[3] Inoue, T.; Henmi, H.; Yoshikawa, Y.; aj.: High-level synthesis for multi-cycle transient fault tolerant datapaths. *2011 IEEE 17th International On-Line Testing Symposium*, červenec 2011, ISSN 1942-9398, s. 13–18, doi:10.1109/IOLTS.2011.5993804.

[4] Iwagaki, T.; Ishimori, Y.; Ichihara, H.; aj.: Designing area-efficient controllers for multi-cycle transient fault tolerant systems. *2015 20th IEEE European Test Symposium (ETS)*, květen 2015, ISSN 1530-1877, s. 1–2, doi:10.1109/ETS.2015.7138742.

[5] Iwagaki, T.; Nakaso, T.; Ohkubo, R.; aj.: Scheduling algorithm in datapath synthesis for long duration transient fault tolerance. *2014 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, říjen 2014, ISSN 1550-5774, s. 128–133, doi:10.1109/DFT.2014.6962062.

[6] Leveugle, R.: Automatic modifications of high level VHDL descriptions for fault detection or tolerance. *Proceedings 2002 Design, Automation and Test in Europe Conference and Exhibition*, 2002, ISSN 1530-1591, s. 837–841, doi:10.1109/DATE.2002.998396.

[7] Leveugle, R.; Cercueil, R.: High level modifications of VHDL descriptions for on-line test or fault tolerance. *Proceedings 2001 IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2001, ISSN 1550-5774, s. 84–91, doi:10.1109/DFTVS.2001.966756.

[8] Lojda, J.; Podivínský, J.; Krčma, M.; aj.: HLS-based Fault Tolerance Approach for SRAM-based FPGAs. *Proceedings of the 2016 International Conference on Field Programmable Technology*, IEEE Computer Society, 2016, ISBN 978-1-5090-5602-6, s. 297–298.

[9] Mentor Graphics: Catapult C/C++/SystemC HLS - Mentor Graphics. `https://www.mentor.com/hls-lp/catapult-high-level-synthesis/c-systemc-hls`, navštíveno: 2016-12-22.

[10] Metra, C.; Francescantonio, S. D.; Omana, M.: Automatic modification of sequential circuits for self-checking implementation. *Proceedings 18th IEEE Symposium on Defect and Fault Tolerance in VLSI Systems*, listopad 2003, ISSN 1550-5774, s. 417–424, doi:10.1109/DFTVS.2003.1250139.

[11] Miller, G.; Carmichael, C.; Swift, G.: Single-event upset mitigation for xilinx FPGA block memories. *XILINX Application Note, Virtex-II FPGAs*, 2007.

[12] Nazar, G. L.; Santos, L. P.; Carro, L.: Scrubbing unit repositioning for fast error repair in FPGAs. *2013 International Conference on Compilers, Architecture and Synthesis for Embedded Systems (CASES)*, září 2013, s. 1–10, doi:10.1109/CASES.2013.6662506.

[13] Nunes, J. L.; Cunha, J. C.; Zenha-Rela, M.: Using Failure Prediction to Improve FPGA Scrubbing. *2016 Seventh Latin-American Symposium on Dependable Computing (LADC)*, říjen 2016, s. 135–138, doi:10.1109/LADC.2016.29.

[14] Padovani, R.: Reconfigurable FPGAs in Space – Present and Future. 2005.

[15] Podivinsky, J.; Cekan, O.; Simkova, M.; aj.: The Evaluation Platform for Testing Fault-tolerance Methodologies in Electro-mechanical Applications. *Microprocess. Microsyst.*, ročník 39, č. 8, listopad 2015: s. 1215–1230, ISSN 0141-9331, doi:10.1016/j.micpro.2015.05.011.

[16] Scharoba, S.; Vierhaus, H. T.: An Interactive Design Space Exploration Tool for Dependable Integrated Circuits. *2016 Euromicro Conference on Digital System Design (DSD)*, srpen 2016, s. 714–717, doi:10.1109/DSD.2016.83.

[17] Shastri, A.; Stitt, G.; Riccio, E.: A scheduling and binding heuristic for high-level synthesis of fault-tolerant FPGA applications. *2015 IEEE 26th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, červenec 2015, ISSN 1063-6862, s. 202–209, doi:10.1109/ASAP.2015.7245735.

[18] Straka, M.; Kastil, J.; Kotasek, Z.: SEU Simulation Framework for Xilinx FPGA: First Step Towards Testing Fault Tolerant Systems. *14th EUROMICRO Conference on Digital System Design*, IEEE Computer Society, 2011, ISBN 978-0-7695-4494-6, s. 223–230.

[19] Tonfat, J.; Kastensmidt, F. L.; Rech, P.; aj.: Analyzing the Effectiveness of a Frame-Level Redundancy Scrubbing Technique for SRAM-based FPGAs. *IEEE Transactions on Nuclear Science*, ročník 62, č. 6, prosinec 2015: s. 3080–3087, ISSN 0018-9499, doi:10.1109/TNS.2015.2489601.

[20] Vargas, F.; Amory, A.: Transient-fault tolerant VHDL descriptions: a case-study for area overhead analysis. *Proceedings of the Ninth Asian Test Symposium*, prosinec 2000, ISSN 1081-7735, s. 417–422, doi:10.1109/ATS.2000.893659.

Logická syntéza založená na obecných operátorech

Ivo Háleček
2. ročník, prezenční studium
školitel: Petr Fišer, školitel specialista: Jan Schmidt

Fakulta informačních technologií, ČVUT v Praze
Thákurova 9
halecivo@fit.cvut.cz

***Abstrakt*—V článku je představena nová reprezentace logických obvodů pomocí Xor-And-Invertor grafů (XAIG) v syntézních algoritmech. XAIG jsou založeny na And-Invertor grafech, orientovaných acyklických grafech, kde uzly představují dvouvstupá hradla AND či XOR a hrany mohou být negované. Nad reprezentací XAIG byl reimplementován syntézní algoritmus rewrite pro nástroj ABC. Reprezentace s více typy uzlů přinesla nová rozhodnutí, která algoritmus musí činit, proto je možné algoritmus ovlivňovat několika parametry, jejichž vliv jsme porovnali na sadě benchmarků. Výsledky dosavadní práce také ukazují, že nový algoritmus je schopen detekovat více XORů než původní algoritmus, a přestože celkově lepší výsledky přináší jen pro podmnožinu testovaných obvodů, možnosti k vylepšení algoritmu jsou známy a shrnuty v závěru článku.**

***Keywords*—AIG, XAIG, rewriting, logická syntéza, ABC**

I. Úvod

Zlepšování logické syntézy bylo v minulých dekádách považováno za dobře řešené téma, v posledních letech však výzkum ukázal, že pro některé struktury není současná logická syntéza schopná produkovat optimální výsledky.

Syntézní nástroje původně reprezentovaly obvody pomocí sumy logických součinů literálů (Sum-of-products - SOP) [1], [2], případně sítí tvořenou z uzlů představujících SOP.

Velký průlom přinesla reprezentace funkcí pomocí binárních rozhodovacích diagramů (Binary decision diagrams - BDD) [3], [4]. Úprava syntézních nástrojů pro tyto struktury vedla k zlepšení jejich výkonu [5], [6], [7], [8].

Binární rozhodovací diagramy trpí špatnou škálovatelností - struktura může růst exponenciálně s počtem vstupů, v závislosti na pořadí vstupních proměnných. Snaha o řešení tohoto problému vedla k fixnímu pořadí proměnných, avšak tím se jen problém posunul k hledání optimálního fixního pořadí vstupních proměnných.

Z těchto důvodů vznikla nová velmi efektivní reprezentace obvodů - And-Inverter-Grafy (AIG) [9], [10], [11]. V AIG je obvod reprezentován pomocí dvouvstupých hradel AND a propojení, která mohou být negovaná. Mnoho algoritmů založených na AIG bylo implementováno do moderního akademického nástroje pro logickou syntézu a verifikaci, ABC [12]. Reprezentace pomocí AIG pravděpodobně je, či brzo bude integrována do komerčních nástrojů [13].

II. Teoretické podklady

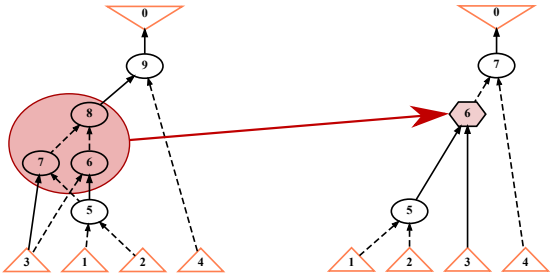
AIG je orientovaný acyklický graf, kde vnitřní uzly představují dvou-vstupá hradla AND a hrany mohou být negované. Kořeny představují výstupy obvodu, listy grafu představují vstupy.

Pro zajištění co nejmenší velikosti logické struktury je důležitá redukce ekvivalentních struktur. Ekvivalence může být buď strukturní nebo funkční. Strukturní ekvivalence znamená, že se v grafu vyskutují dvě stejné struktury reprezentující stejnou funkci primárních vstupů obvodu. Funkční ekvivalence znamená, že se v grafu vyskytují různé struktury, které ale stále reprezentují stejnou funkci primárních vstupů. V ABC je strukturní ekvivalence řešena automaticky při konstrukci AIG pomocí strukturního hashování. Je tedy zajištěno, že v grafu nejsou dvě ekvivalentní struktury. Stále však v grafu mohou být dvě různé struktury reprezentující stejnou funkci.

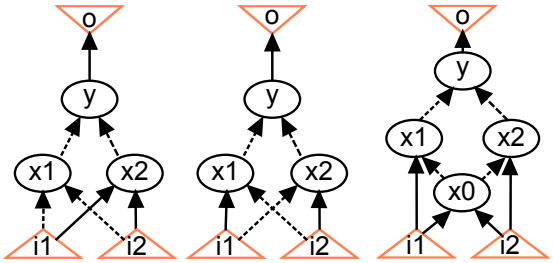
Rewriting [11] je algoritmus, který nahrazuje podgrafy s k listy (tzv. k-fesible CUTs [14]) funkčně ekvivalentními strukturami, s cílem zmenšit celkový počet uzlů grafu. V ABC je rewriting dostupný pro reprezentaci AIG a pracuje s podgrafy o 4 listech. Pro každý uzel jsou enumerovány všechny podgrafy s k listy. Každý takový podgraf je vyjádřen jako funkce jeho listů, reprezentována pravdivostní tabulkou. Tato funkce je následně převedena do kanonického tvaru na funkci, která je NPN-ekvivalentní (převoditelná na původní funkci pomocí permutací a negací vstupů a případné negace výstupu). Pro každou NPN třídu funkcí je k předpočítána jedna nebo více nahrazovacích struktur (replacement structure). Pro $k = 4$ existuje 2^{16} různých funkcí, avšak jen 222 NPN tříd, proto je $k = 4$ vhodný kompromis mezi paměťovou náročností a sílou rewritingu. Aplikováno je nahrazení takovou strukturou, která ušetří co nejvíce uzlů (pokud nějaká vůbec uzly ušetří). Toho se může dosáhnout tak, že je buď nahrazovací struktura přímo menší než nahrazovaná, nebo nový podgraf obsahuje strukturně ekvivalentní části obsažené již ve zbytku grafu a uzly se tak ušetří strukturním hashováním.

III. Popis problému

Přesto, že AIG jsou logickou reprezentací s dobrou škálovatelností, výzkum zaměřený na konstrukci benchmarků pro syntézu FPGA [15] odhalil, že pro některé obvody funguje současná logická syntéza neefektivně, produkuje řádově větší obvody, než optimální.



Obrázek 1: Funkce $Y = \neg(\neg(x_1 \vee x_2) \oplus x3) \wedge \neg x_4$ reprezentovaná v AIG (vlevo) a XAIG (vpravo). Oválné uzly představují funkci AND, šestiúhelníky uzly typu XOR. Čárkované hrany jsou negovány.



Obrázek 2: Funkce XOR reprezentována v AIG.

Pozdější výzkum také ukázal, že exisují dokonce i velmi malé obvody, pro které logická syntéza takto selhává [16], také byl představen způsob, jak vytvořit takové obvody z reálných praktických (průmyslových) obvodů [17], [18].

Jednou ze společných charakteristik těchto obvodů je vysoká intenzita XORů, především stromů XORů, které se staly běžnými s rozvojem aritmetických jednotek a systémů odolných proti poruchám. Hlavním problémem pro současné syntézní nástroje se zdá být, pokud tyto struktury nejsou dostatečně dobře popsané - například pokud jsou rozpuštěny do dvouúrovňové struktury - obvykle SOP.

V nedávné době se objevily nové reprezentace obvodů snažící se neefektivitu algoritmů postavených nad dosavadními reprezentacemi řešit. Jako alternativa k BDD vznikly Bikondicionální binární rozhodovací diagramy (BBDDs) [19], kde uzly jsou multiplexory řízené rovností dvou řídicích signálů. Nad AIG vznikla reprezentace Majority-Inverter-Grafů (MIG) [20], kde uzly představující dvouvstupý AND byly nahrazeny třívstupými majoritami. Uzly typu AND, resp. OR lze jednoduše pomocí uzlu majorita (MAJ) vyjádřit přivedením konstanty *0*, resp. *1* na jeho třetí vstup.

Velmi novou reprezentací jsou také XOR-Majority-Grafy (XMG) [21]. Zde je graf tvořen uzly dvou typů - MAJ a XOR.

IV. Navrhované řešení

Jedním ze způsobů, jak se pokusit zlepšit výsledky syntézy, je přepsat současné syntézní algoritmy tak, aby pracovaly s obecnější reprezentací obvodů. K řešení konkrétně problému s XOR intenzivními strukturami, navrhujeme syntézu postavenou nad reprezentací XAIG.

Xor-And-Inverter-Grafy (XAIG) je námi představená reprezentace, která rozšiřuje původní AIG o dvou-vstupé uzly typu XOR. Ukázka funkce $Y = \neg(\neg(x_1 \vee x_2) \oplus x3) \wedge \neg x_4$ v AIG a XAIG je vidět na Obrázku 1.

V původním AIG bylo nutné funkci XOR reprezentovat pomocí uzlů AND, jak je zobrazeno na Obrázku 2. Takto popsané XORy lze v AIG dobře strukturně identifikovat. V XAIG je XOR reprezentován pomocí speciálního typu uzlu.

Jako základ syntézního procesu nad XAIG jsem reimplementoval rewriting tak, aby pracoval nad XAIG. Algoritmus jsem implementoval stejným způsobem jako originální rewriting s tím rozdílem, že nahrazovací struktury byly reprezentovány pomocí XAIG (obsahovaly tedy nativní XORy) a generoval jsem pouze jednu nahrazovací strukturu pro každou NPN třídu.

Nahrazovací struktury XAIG jsem předpočítal načtením pravděpodobnostní tabulky pomocí příkazu *read_truth* do ABC, následně byly optimalizovány příkazem *dch* a namapovány příkazem *map* do standardních buněk s knihovnou obsahující invertory s cenou 1, a hradla AND a XOR s cenou 2. Namapované netlisty byly následně převedeny do XAIG. Totožná cena ANDů a XORů umožní vytvořit každý identifikovaný XOR místo toho, aby v některém případě byl realizován pomocí ANDů.

Různé typy uzlů přinesly algoritmu nová nutná rozhodnutí. Pokud víme, že pro cílovou technologii má pro nás XOR jinou cenu než AND (mapujeme-li do standardních buněk místo do FPGA), může pro nás být výhodnější vytvořit více ANDů místo méně XORů. Proto jsem do XAIG rewritingu implementoval konfigurovatelnou cenu ANDů a XORů a rewriting nerozhoduje o výhodnosti náhrady podgrafu na základě ušetřených uzlů, ale na základě změny celkové ceny uzlů grafu. To umožňuje řídit preferenci XORů nad ANDy.

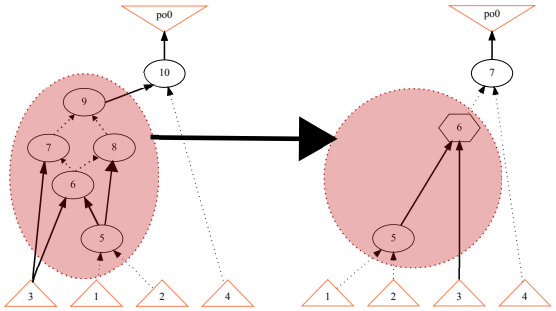
Pokud jsou v grafu ANDy tvořící XOR, ale zároveň jeden z vnitřních ANDů je vstup pro jinou část obvodu, musí se při nahrazení této struktury nativním XORem vnitřní uzel AND zduplikovat, aby nebyl ztracen onen výstup. Takovým XORům říkáme “non fanout-free". Někdy může být proto výhodné nativní XOR nevytvářet. V XAIG rewritingu je toto možné řídit povolením rozpouštění XORů, XOR je nahrazen ANDy, pokud to díky strukturnímu sdílení s ostatní částí grafu vede k menší celkové ceně uzlů.

Příklad jednoho nahrazení v rewritingu nad XAIG je zobrazen na obrázku 3.

V. Předběžné výsledky

Rewriting implementovaný nad XAIG jsem porovnal s původním AIG rewritingem. Zajímal nás především počet identifikovaných XORů a optimalizační výkon algoritmu.

Srovnání algoritmů z hlediska identifikace XORů je zobrazeno v Tabulce I. Sloupce obsahují počet XORů nalezených strukturní identifikací v původním obvodu, po XAIG rewritingu pro různý poměr cen uzlu AND a XOR, a po původním AIG rewritingu následovaným strukturní identifikací. Řádek *celkem* obsahuje sumu dané charakteristiky přes



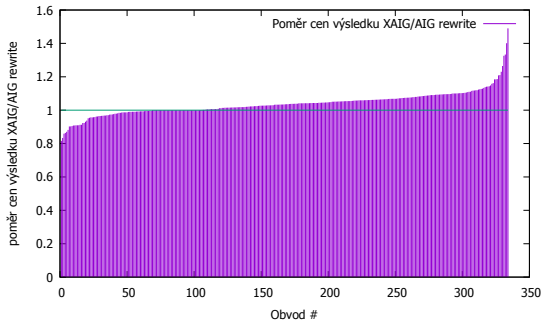
Obrázek 3: Příklad jednoho kroku rewritingu nad XAIG.

Tabulka I: Srovnání identifikace XORů původním AIG rewritingem s naším XAIG rewritingem, a současně se strukturní identifikací v původních obvodech.

obvod	původní	XAIG (cena AND:XOR) 1:1	1:3	2:5	AIG
c6288	0	476	0	433	28
bigkey	5	109	4	4	5
mm30a	60	116	2	30	58
c1355	0	107	82	104	56
prom2	27	67	34	46	32
s635	0	31	0	2	0
g25	50	50	27	27	30
ex1010	43	62	41	51	46
mm9a	18	32	2	10	16
Altera_oc_hdlc	132	149	117	141	134
Mentor_1_11	3	62	50	56	47
Mentor_1_12	3	62	50	56	47
mm9b	19	32	3	11	17
celkem	**5289**	**6632**	**4588**	**5629**	**5345**
více než v AIG		123	25	56	
méně než v AIG		19	146	102	

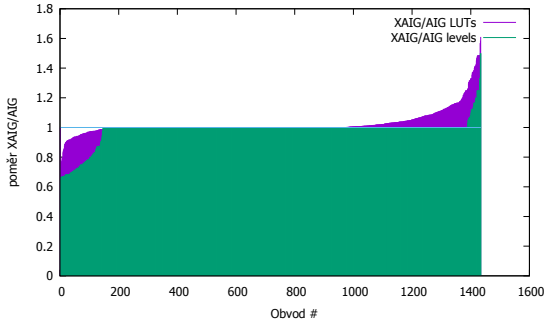
všechny měřené obvody. Poslední dva řádky uvádějí, pro kolik obvodů byla daná charakteristika vyšší resp. nižší po rewritingu nad XAIG proti původnímu rewritingu nad AIG. Obě verze algoritmu rewriting dokáží u většiny obvodů detekovat více XORů, než strukturní identifikace v původním obvodu. Ve většině případů je XAIG rewriting z hlediska identifikace silnější než AIG rewriting, s klesající nastavenou cenou XORů pochopitelně počet nalezených XORů roste. Měření bylo provedeno nad více než 1 400 obvody ze sad benchmarků LGSynth‘91 [22], IWLS‘93 [23], ISCAS‘85 [24], ISCAS‘89 [25], Advanced Synthesis Cookbook [26], IWLS 2005 [27] a dalšími [28], [29], a [30] - dostupnými z [31].

Pro základní srovnání z hlediska redukce velikosti výsledného grafu jsem u obou algoritmů spočítal výslednou cenu uzlů jako součet počtu uzlů typu AND a trojnásobku počtu uzlů typu XOR. Toto ocenění vychází ze skutečnosti, že každý XOR lze vyjádřit pomocí maximálně tří uzlů typu AND. Srovnání je zobrazeno v grafu 4, kde jsou ve vzestupném pořadí vyneseny poměry výsledné ceny grafu pro obě verze algoritmu rewriting. Hodnoty nižší než 1 znamenají nižší cenu pro XAIG rewriting. Je však nutno zmínit, že vyšší cena grafu neznamená automaticky větší plochu po mapování na technologii, mapovací nástroj může lépe zpracovat větší, zato



Obrázek 4: Porovnání rewritingu nad AIG vůči rewritingu nad XAIG z hlediska celkové ceny uzlů obvodu.

vhodněji strukturovaný graf.



Obrázek 5: Porovnání rewritingu nad AIG vůči rewritingu nad XAIG z hlediska zpoždění a počtu LUTů po mapování na FPGA.

Pro srovnání rewritingu nad AIG a XAIG z hlediska optimalizační síly jsem výsledky obou algoritmů namapoval na 6-LUTy a porovnal zpoždění a počet LUTů. Cena uzlů v XAIG rewritingu byla nastavena 1:1, jelikož v FPGA má pro nás XOR stejnou cenu jako AND, poměr mezi výsledkem XAIG rewritingu a AIG rewritingu je pro obě charakteristiky zobrazen v grafu na Obrázku 5. Z výsledků je vidět, že XAIG rewriting je z celkového pohledu lepší v optimalizaci zpoždění, pro některé obvody je lepší i z hlediska minimalizace počtu LUTů.

VI. Závěr a práce do budoucna

Rozšířil jsem původní reprezentaci logických funkcí AIG o XOR uzly a vytvořil tak reprezentaci XAIG. Nad toutou reprezentací jsem reimplementoval jednoduchý syntézní algoritmus rewriting. Jelikož nový typ uzlů přinesl do algoritmu nová rozhodnutí, umožnil jsem kontrolu průběhu algoritmu parametrizovatelnou preferencí XORů a možností XORy rozpouštět do ANDů, pokud je to výhodné. XAIG rewriting jsem srovnal s původním AIG rewritingem z hlediska identifikace XORů a z hlediska výsledného zpoždění a počtu LUTů po mapování na FPGA. XAIG rewriting je silnější v identifikaci XORů, celkově je silnější z hlediska zpoždění po mapování a pro některé obvody je silnější i z hlediska minimalizace počtu LUTů.

XAIG rewriting ale v současné předběžné verzi trpí nahrazovacími strukturami. Odhalili jsme, že ne všechny naším způsobem generované struktury jsou optimální, navíc používáním pouze jedné struktury pro každou z NPN tříd vede k menší redukci uzlů díky strukturnímu sdílení. Proto jsme pro budoucí experimenty připravili struktury generované exaktní syntézou a algoritmus jsem upravil tak, aby pro každou z NPN tříd zkoušel nahrazení každou možnou optimální nahrazovací strukturou. Nová verze algoritmu však byla vytvořena až po odeslání tohoto článku.

Z dlouhodobějšího zaměření je možnost zobecnit syntézní proces na rozdělení do nezávislých vrstev tak, že bude možné kombinovat různé reprezentace (více typů uzlů), různé způsoby generování podgrafů (například vícevýstupové funkce) a různé nahrazovací struktury (předpočítané, případně generované online).

Cílem dizertační práce bude syntézní proces, který bude schopný efektivně pracovat se strukturami, kde je současná syntéza neoptimální. Výsledkem tak nebude jen nová verze algoritmu rewriting, ale zaměřím se také na identifikaci struktur, pro které samotný rewriting není schopný produkovat dostatečně dobrá řešení a hledání dalších způsobů, jak s těmito strukturami efektivně pracovat. Pravděpodobně bude také nutné upravit další syntézní algoritmy. Jak už ale ukázal rewriting postavený nad XAIG, obvykle nestačí pouze nahradit strukturu, nad kterou algoritmus pracuje, ale řešit i nová rozhodnutí, která s novými typy uzlů nastanou.

Poděkování

Práce byla částečně podpořena z grantů:

- GAČR: GA16-05179S Výzkum vztahů a společných vlastností spolehlivých a bezpečných architektur založených na programovatelných obvodech (03/2016 - 12/2018).
- SGS17/213/OHK3/3T/18: Bezpečné a spolehlivé architektury pro programovatelné obvody.

Výpočetní prostředky byly poskytnuty CESNET LM2015042 a CERIT Scientific Cloud LM2015085, pod programy "Projects of Large Research, Development, and Innovations Infrastructures"

Reference

[1] E. McCluskey, "Minimization of Boolean functions," *The Bell System Technical Journal*, vol. 35, no. 6, pp. 1417–1444, Nov 1956.

[2] R. K. Brayton, A. L. Sangiovanni-Vincentelli, C. T. McMullen, and G. D. Hachtel, *Logic Minimization Algorithms for VLSI Synthesis*. Norwell, MA, USA: Kluwer Academic Publishers, 1984.

[3] S. B. Akers, "Binary Decision Diagrams," *IEEE Transactions on Computers*, vol. 27, no. 6, pp. 509–516, Jun. 1978.

[4] R. E. Bryant, "Graph-Based Algorithms for Boolean Function Manipulation," *IEEE Transactions on Computers*, vol. 35, no. 8, pp. 677–691, Aug. 1986.

[5] K. Karplus, "Using if-then-else DAGs for Multi-Level Logic Minimization," in *Proc. of Advance Research in VLSI, C. Seitz Ed.* MIT Press, 1989, pp. 101–118.

[6] Y.-T. Lai, M. Pedram, and S. B. K. Vrudhula, "BDD Based Decomposition of Logic Functions with Application to FPGA Synthesis," in *Proceedings of the 30th International Design Automation (DAC'93)*. New York, NY, USA: ACM, 1993, pp. 642–647.

[7] C. Yang and M. Ciesielski, "BDS: A BDD-Based Logic Optimization System," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 7, pp. 866–876, August 2002.

[8] N. Vemuri, P. Kalla, and R. Tessier, "BDD-based logic synthesis for LUT-based FPGAs," *ACM Transactions on Design Automation of Electronic Systems*, vol. 7, no. 4, pp. 501–525, 12 2001.

[9] A. Kuehlmann, V. Paruthi, F. Krohm, and M. Ganai, "Robust Boolean reasoning for equivalence checking and functional property verification," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 12, pp. 1377–1394, 2001.

[10] P. Bjesse and A. Borälv, "DAG-aware circuit compression for formal verification," in *IEEE/ACM International Conference on Computer-Aided Design*, 2004, pp. 42–49.

[11] K. Brayton, Robert, A. Mishchenko, and S. Chatterjee, "DAG-aware AIG rewriting: a fresh look at combinational logic synthesis," in *43rd ACM/IEEE Design Automation Conference*. ACM, 2006, pp. 532–535.

[12] A. Mishchenko *et al.*, "ABC: A system for sequential synthesis and verification," 2012. [Online]. Available: http://www.eecs.berkeley.edu/˜alanmi/abc

[13] R. Brayton and A. Mishchenko, "ABC: An Academic Industrial-strength Verification Tool," in *Proceedings of the 22nd International Conference on Computer Aided Verification (CAV'10)*. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 24–40.

[14] A. Mishchenko, S. Chatterjee, K. Brayton, Robert, X. Wang, and T. Kam, "Technology Mapping with Boolean Matching, Supergates and Choices," ERL Technical Report, EECS Dept., UC Berkeley, Tech. Rep., 03 2005.

[15] J. "Cong and K. Minkovich, ""Optimality Study of Logic Synthesis for LUT-Based FPGAs"," in *"14th International ACM Symposium on Field-Programmable Gate Arrays"*, "2006", pp. "33–40".

[16] P. Fiser and J. Schmidt, "Small but Nasty Logic Synthesis Examples," in *8th. Int. Workshop on Boolean Problems (IWSBP)*, 2008, pp. 183–189.

[17] J. Schmidt and P. Fiser, "The Case for a Balanced Decomposition Process," in *12th Euromicro Conference on Digital System Design, Architectures, Methods and Tools*, Aug 2009, pp. 601–604.

[18] P. Fiser and J. Schmidt, "New Ways of Generating Large Realistic Benchmarks for Testing Synthesis Tools," in *9th. Int. Workshop on Boolean Problems (IWSBP)*, 2010, pp. 157–164.

[19] L. Amaru, P.-E. Gaillardon, and G. De Micheli, "Biconditional Binary Decision Diagrams: A Novel Canonical Logic Representation Form," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 4, no. 4, pp. 487–500, Dec 2014.

[20] ——, "Boolean logic optimization in Majority-Inverter Graphs," in *52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, June 2015, pp. 1–6.

[21] W. Haaswijk, M. Soeken, L. Amaru, P.-E. Gaillardon, G. De Micheli, "A Novel Basis for Logic Rewriting," Integrated Systems Laboratory, EPFL, Lausanne, Switzerland, Tech. Rep., 2017.

[22] S. Yang, "Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0," MCNC Technical Report, Tech. Rep., Jan. 1991.

[23] K. McElvain, "IWLS'93 Benchmark Set: Version 4.0," Tech. Rep., May 1993.

[24] F. Brglez and H. Fujiwara, "A Neutral Netlist of 10 Combinational Benchmark Circuits and a Target Translator in Fortran," in *IEEE International Symposium Circuits and Systems (ISCAS'85)*. IEEE Press, Piscataway, N.J., 1985, pp. 677–692.

[25] F. Brglez, D. Bryan, and K. Kozminski, "Combinational profiles of sequential benchmark circuits," in *IEEE International Symposium on Circuits and Systems (ISCAS'89)*, May 1989, pp. 1929–1934 vol.3.

[26] Altera, "Advanced Synthesis Cookbook," Tech. Rep., Jul. 2011.

[27] C. Albrecht, "IWLS 2005 Benchmarks," Tech. Rep., Jun. 2005.

[28] ——, "Altera's Quartus University Interface Program (QUIP)," Tech. Rep., Jun. 2005.

[29] S. Yang, "Logic Synthesis and Optimization Benchmarks User Guide: Version 3.0," MCNC Technical Report, Tech. Rep., 1989.

[30] "Berkeley PLA Test Set Results," Tech. Rep., Jun. 1986.

[31] P. Fiser and J. Schmidt, "A Comprehensive Set of Logic Synthesis and Optimization Examples," in *12th. Int. Workshop on Boolean Problems (IWSBP)*, 2016, pp. 151–158. [Online]. Available: http://ddd.fit.cvut.cz/prj/Benchmarks/

Rozvoj techník návrhu nízko-napäťových integrovaných systémov

Matej Rakús
2. ročník, denná prezenčná forma štúdia
Školiteľ: Viera Stopjaková

Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave
Ilkovičova 3, 812 19 Bratislava, SR
matej.rakus@stuba.sk

***Abstrakt*—Tento príspevok sa zaoberá analýzou rôznych techník vhodných pre návrh nízko-napäťových a nízko-príkonových integrovaných obvodov (IO), so zameraním sa na analogové obvody. Z vykonanej analýzy vyplýva, že pre návrh IO s nízkym napájacím napätím v štandardnej CMOS technológii a bez potreby zásahu do štruktúry alebo technologického procesu, sú vhodné techniky využívajúce MOS tranzistory riadené substrátovou elektródou, MOS tranzistory s dynamickým prahovým napätím a MOS tranzistory pracujúce v slabej a strednej oblasti inverzie. Základné obvodové bloky analogových IO, napríklad prúdové zrkadlá na báze takto zapojených MOS tranzistorov sú schopné pracovať pri napájacích napätiach okolo 0,6 V, ba dokonca aj nižších. Výsledky získané zo simulácií a meraní experimentálnych obvodov potvrdzujú, že tieto techniky môžu byť kľúčovými pre návrh nízko-napäťových IO s nízkou spotrebou.**

***Kľúčové slová*—bulk-driven, dynamic-threshold, oblasť inverzie, nízke napájacie napätie, analógové integrované obvody**

I. Úvod

Zvýšený dopyt po všadeprítomných prenosných elektronických zariadeniach spôsobil narastajúcu potrebu rozvoja techník návrhu pre nízko-napäťové analógové integrované obvody (IO) v štandardnej CMOS technológii. Tieto techniky návrhu spĺňajú súčasné požiadavky spojené so znižovaním napájacieho napätia a zároveň sa javia ako kľúčové pre dosiahnutie vyššej výkonnosti navrhnutých obvodov. Návrh nízko-napäťových IO je náročnou výzvou pre návrhára nielen z pohľadu súčasných požiadaviek zákazníkov, ale hlavne vzhľadom na fluktuáciu technologických parametrov nanometrových CMOS technológií.

Zmenšovanie rozmerov tranzistorov ako aj neustále zmenšovanie hrúbky vrstvy izolačného oxidu (rádovo jednotky nm) má za následok nízke hodnoty prierazného napätia tranzistorov. Na zabezpečenie správnej činnosti a spoľahlivosti obvodu je teda potrebné znížiť aj napájacie napätie obvodu. Na druhej strane je však pokles hodnoty prahového napätia tranzistorov V_{TH} v porovnaní s trendom poklesu napájacieho napätia V_{DD} podstatne menej významný, čo má za následok zníženie rozsahu pracovných napätí a zhoršenie spínacích charakteristík MOS tranzistorov [1].

V CMOS analógových obvodoch bez využitia špeciálnych techník návrhu nízko-napäťových IO je minimálna hodnota napájacieho napätia obmedzená na hodnotu danú súčtom napätia medzi hradlom a emitorom V_{GS} CMOS štruktúry a požadovaného rozkmitu signálu. V 130 nm štandardnom CMOS procese je hodnota napätia V_{GS} potrebná na otvorenie MOS tranzistora a dosiahnutie silnej inverzie približne 300 mV. Následne tak nastáva problém pri nízkom napájacom napätí ($V_{DD} \approx 0,6\ V$) a štruktúrach so zapojením viacerých tranzistorov nad seba do tzv. *kaskódy*, a to kvôli nedostatočnému, prípadne žiadnemu rozkmitu napätia potrebného na spracovanie signálu.

II. Techniky návrhu nízko-napäťových IO

Existuje niekoľko techník pre návrh nízko-napäťových a nízko-príkonových analógových IO [2]. Najvhodnejšie z nich pre vybranú štandardnú 130 nm CMOS technológiu sú techniky využívajúce: a) MOS tranzistory pracujúce v podprahovej oblasti, b) MOS tranzistory riadené substrátovou elektródou (ďalej BD *z angl.* Bulk-Driven) a c) MOS tranzistory s dynamickým prahovým napätím (ďalej DT *z angl.* Dynamic-Threshold).

A. MOS tranzistory pracujúce v podprahovej oblasti

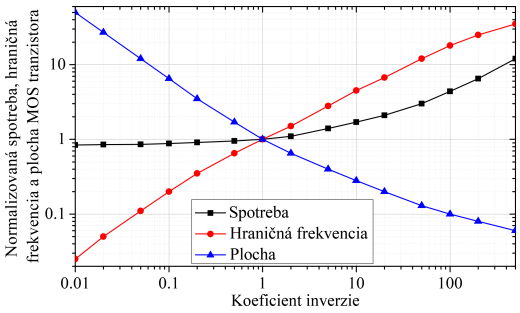
Základný a veľmi rozšírený model MOS tranzistora je použiteľný iba v oblastiach silnej inverzie a saturácie. Ak $V_{GS} < V_{TH}$, tranzistorom by v ideálnom prípade nemal pretekať prúd. V skutočnosti, pri vstupnom napätí menšom ako prahové napätie nie je kolektorový prúd nulový, ale klesá exponenciálne v závislosti od hodnoty napätia V_{GS}.

Oblasť inverzie MOS tranzistora je určená tzv. koeficientom inverzie i_c, ktorý je matematicky definovaný vzťahom 1, kde I_D je kolektorový prúd, I_0 je tzv. technologický prúd, ktorý je špecifický pre danú technológiu, μ je pohyblivosť voľných nosičov náboja, C_{ox} je kapacita hradlového (izolačného) oxidu, W je šírka hradla MOS tranzistora, L je jeho dĺžka a U_T je termálne napätie.

$$i_c = \frac{I_D}{I_0(W/L)} = \frac{I_D}{2\mu C_{oc} U_T^2 (W/L)} \tag{1}$$

Ak je technologický prúd rovnaký ako kolektorový prúd, tranzistor pracuje presne v strede strednej inverzie ($i_c = 1$). MOS tranzistor pracuje v slabej (silnej) inverzii, ak koeficient

inverzie $i_c < 0,1$ ($i_c > 10$), ako je možné pozorovať zo závislostí normalizovanej plochy tranzistora, hraničnej frekvencie a spotreby tranzistora od koeficientu inverzie, znázornených na obr. 1.



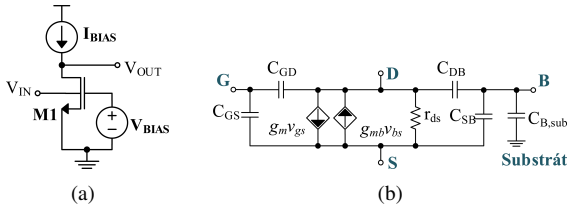
Obr. 1. Závislosť normalizovanej spotreby, hraničnej frekvencie a plochy MOS tranzistora od koeficientu inverzie

Parameter g_m/I_D vyjadruje efektivitu prenosovej vodivosti MOS tranzistora a hovorí o tom, ako efektívne sa prúd (resp. príkon) prenesie do prenosovej vodivosti tranzistora. Pre analógové obvody navrhnuté v CMOS technológii, MOS tranzistor dosahuje najvyššie hodnoty g_m/I_D práve vtedy, keď pracuje v slabej inverzii. Výhodou g_m/I_D metodiky je, že stanovuje jednoduché pravidlá pre návrh rozmerov MOS tranzistora, ktoré návrhár môže zvážiť pre ktorúkoľvek oblasť inverzie [3]. Táto metodika opisuje oblasť strednej inverzie s vysokou presnosťou, ktorá je pre väčšinu analógových obvodov najlepším kompromisom vzhľadom na spotrebu, rýchlosť a plochu. Závislosť parametra g_m/I_D od koeficientu inverzie možno opísať vzťahom 2.

$$\frac{g_m}{I_D} = \frac{1}{nU_T} \frac{1}{0,5 + \sqrt{0,25 + i_c}} \tag{2}$$

B. MOS tranzistory riadené substrátovou elektródou

Prúd I_D MOS tranzistora je zvyčajne riadený napätím V_{GS}. Napätie medzi substrátovou elektródou a emitorom V_{BS} však môže taktiež ovplyvniť tento prúd, čo je zvyčajne považované za parazitný jav vnášajúci do obvodu nechcenú substrátovú vodivosť g_{mb}. Pripojením konštantného napätia $V_{GS} = V_{BIAS}$ na hradlo MOS tranzistora a privedením vstupného signálu na substrátovú elektródu, vykazuje MOS tranzistor podobné vlastnosti ako JFET tranzistor. Takéto zapojenie voláme BD tranzistor a je znázornené na obr. 2(a).



(a) (b)

Obr. 2. NMOS tranzistor: (a) BD zapojenie; (b) náhradná schéma

Substrátová vodivosť g_{mb} je sekundárny jav MOS tranzistora, ktorý reprezentuje zisk napätím V_{BS} riadeného prúdového zdroja umiestneného medzi kolektorom a emitorom (vo všeobecne známom náhradnom modeli MOS tranzistora znázorneného na obr. 2(b)). Táto vodivosť je vyjadrená pomocou vzťahu 3, kde g_m je prenosová vodivosť bežného hradlom riadeného (GD *z angl.* Gate-Driven) MOS tranzistora, γ je substrátový činiteľ, ϕ_F je Fermiho potenciál and η udáva pomer medzi g_{mb} a g_m. η je zvyčajne v rozmedzí od 0,2 do 0,4 v závislosti od hodnoty V_{BS} a špecifických parametrov technológie [4].

$$g_{mb} = \frac{\gamma g_m}{2\sqrt{-\phi_F - V_{BS}}} = g_m \cdot \eta \tag{3}$$

Využívanie g_{mb} namiesto g_m znižuje celkovú vodivosť tranzistora a zvyšuje vstupnú kapacitu, ktorá je 3 až 5-krát väčšia ako vstupná kapacita bežného MOS tranzistora, čo vedie k nižšej šírke pásma zosilnenia (GBW *z angl.* Gain Band-Width). Navyše technika využívajúca BD tranzistory má za následok zvýšenie tepelného šumu a zvýšenie rizika zopnutia parazitných bipolárnych tranzistorov v substráte, čo môže viesť k tzv. „*Latch-up*“ javu [5]. Na druhej strane, výhodou tejto techniky je jej využiteľnosť v štandardnom CMOS procese bez potreby zmeny štruktúry alebo technológie. Ďalšou výhodou je, že použitie BD techniky výrazne znižuje vplyv prahového napätia V_{TH} na funkciu tranzistora (nie je potreba prekonávať V_{TH} v signálovej ceste) [6].

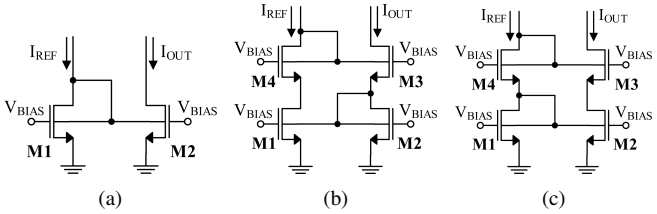
C. MOS tranzistory s dynamickým prahovým napätím

MOS tranzistory s dynamickým prahovým napätím DTMOS (ďalej DT) sú odvodené z BD techniky s tým rozdielom, že namiesto substrátového kontaktu využívajú ako vstup tranzistora združený substrátový kontakt spojený s hradlom [74]. Princíp tejto technika spočíva v tom, že vstupné napätie musí byť $V_{BS} = V_{GS} > 0$ pre NMOS (a $V_{BS} = V_{GS} < 0$ pre PMOS), čím je dynamicky redukované aj prahové napätie tranzistora V_{TH}. Vďaka dynamickému predpätiu na substrátovom kontakte je potenciál v oblasti kanála riadený pomocou hradla a substrátového kontaktu súčasne, čo vedie k vysokej celkovej vodivosti $g_m + g_{mb}$ a rýchlejšiemu prúdovému prenosu.

III. Experimentálne výsledky

A. Simulácie

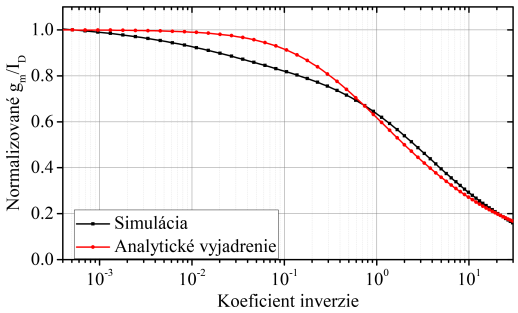
Na overenie týchto techník bolo navrhnutých niekoľko experimentálnych obvodov. Keďže prúdové zrkadlá (PZ) patria medzi základné stavebné bloky používané pri návrhu analógových IO, v experimente boli použité tri základné topológie prúdových zrkadiel navrhnutých pomocou GD a BD techniky v 130 nm štandardnej CMOS technológii. Výstupom tejto analýzy je porovnanie simulácií a meraní na vyhodnotenie presnosti modelov MOS tranzistorov pre návrh obvodov s napájacím napätím nižším ako 600 mV. Obr. 3 znázorňuje PZ navrhnuté použitím BD techniky. Substrátové elektródy tranzistorov v oboch vetvách zrkadla sú navzájom prepojené a na hradlá je pripojené predpätie $V_{BIAS} = 300\ mV$. Minimálne výstupné napätie $V_{MIN} = V_{DS,sat}$ jednoduchého PZ nezávisí od prahového napätia V_{TH} a ani od celkovej vodivosti g_{mbs}, a preto výstupné charakteristiky a výstupný odpor GD a BD ekvivalentov by mali byť takmer zhodné [7].



Obr. 3. Prúdové zrkadlá: (a) jednoduché BD; (b) vylepšené Wilsonovo BD; (c) kaskódové BD

Minimálne výstupné napätie $V_{MIN} = V_{TH} + 2V_{DS,sat}$ vylepšeného Wilsonovho a kaskódového PZ závisí od prahového napätia a ich výstupný odpor $r_{out} = g_m r_{ds}^2$ závisí od prenosovej vodivosti, ich výstupné charakteristiky sa budú navzájom líšiť [8]. Tento predpoklad potvrdzujú aj charakteristiky získané simuláciami a overené meraniami nachádzajúce sa v podkapitole *Výsledky merania čipov*. Oblasť charakteristiky v okolí V_{MIN} zároveň reprezentuje prechod zo slabej do silnej oblasti inverzie MOS tranzistora. Odchýlka výsledkov simulácií od reálnych meraní nám tiež napovie o presnosti použitých modelov v týchto oblastiach inverzie MOS tranzistora. Simulácie boli vykonané v prostredí Virtuoso® XL od spoločnosti Cadence® s použitím SPICE modelov tranzistorov.

Simulácia parametra g_m/I_D odhaľuje nepresnosť použitých simulačných modelov MOS tranzistorov v oblasti slabej a strednej inverzie. Porovnanie kriviek získaných analyticky (rovnica 2) a simuláciou je znázornené na obr. 4. Zaznamenaná nepresnosť modelov je najväčším problémom pri návrhu a simulácii nízko-napätových IO.

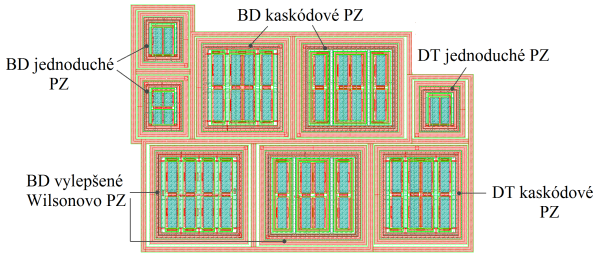


Obr. 4. Závislosť normalizovaného parametra g_m/I_D od koeficientu inverzie NMOS tranzistora

B. Návrh topografie testovacieho čipu

Navrhnuté prúdové zrkadlá sme následne implementovali do fyzického návrhu topografie, ktorý sa stal súčasťou testovacieho prototypového čipu. Na testovacom čipe sa nachádza celkovo 8 testovacích štruktúr prúdových zrkadiel zahŕňajúcich BD a DT PZ (obr. 5). Niektoré topológie sa na čipe nachádzajú dva krát z dôvodu rozdielnej topografie daného zrkadla. Týmto spôsobom je možné vyhodnotiť vplyv rôzneho rozloženia a topografie na rozptyl parametrov a charakteristík PZ.

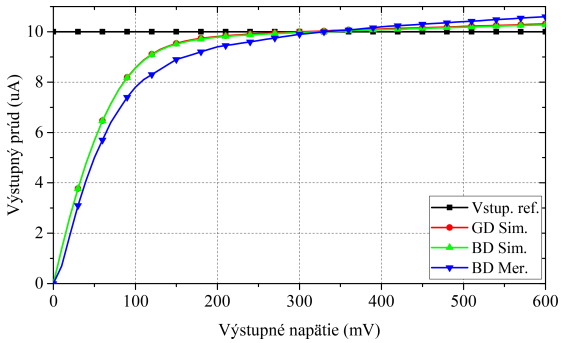
Na testovacom prototypovom čipe sa nachádza aj 12 testovacích tranzistorov s rozličným pomerom šírky a dĺžky hradla určených na ich charakterizáciu, najmä pre činnosť v oblasti strednej a slabej inverzie (a pre BD aplikácie). Očakávame, že charakterizácia a jej výsledky prinesú spresnenie súčasných simulačných modelov MOS tranzisotrov pre návrh IO s nízkym napájacím napätím.



Obr. 5. Návrh topografie prúdových zrkadiel

C. Výsledky merania čipov

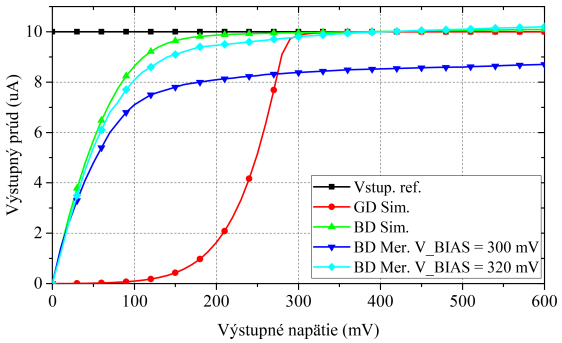
Simulované charakteristiky jednoduchých GD a BD zrkadiel sa prekrývajú, ako bolo spomenuté vyššie, s minimálnym výstupným napätím okolo 120 mV. Z toho vyplýva, že jednoduché PZ je vhodné pre návrh IO s nízkym napájacím napätím. Avšak jeho nevýhoda je v relatívne nízkom výstupnom odpore, ktorý sa pohybuje rádovo v stovkách $k\Omega$. Odsimulované a namerané výstupné charakteristiky sú znázornené na obr. 6. Merania odhalili, že reálny výstupný odpor jednoduchých PZ vyrobených na testovacom čipe je ešte nižší a to približne jedna tretina hodnoty v porovnaní so simuláciami.



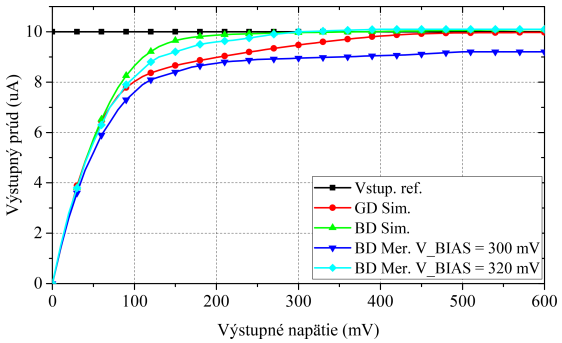
Obr. 6. Výstupné charakteristiky jednoduchého PZ

Vylepšené Wilsonovo PZ je schopné zvýšiť výstupný odpor zaradením zápornej sériovej spätnej väzby do obvodu. Z obr. 7 je možné pozorovať, že BD technika výrazne znižuje V_{MIN} z hodnoty okolo 300 mV na hodnotu približne 120 mV, ale s vyššou hodnotou výstupného odporu, ktorý je rádovo v jednotkách $M\Omega$. Podobná situácia je aj pri kaskódovom PZ (obr. 8). Minimálne výstupné napätie je opäť výrazne znížené vďaka použitiu BD techniky na hodnotu približne 150 mV. Merania oboch topológií potvrdili simulované výsledky s miernou nepresnosťou - pre správnu činnosť a na dosiahnutie požadovanej presnosti zrkadlenia musia byť upravené DC predpäťové podmienky na hradlách MOS tranzistorov z 300 mV na 320 mV [9], [10].

Porovnanie hodnôt minimálneho výstupného napätia V_{MIN}, ktoré boli získané simuláciami a meraniami sú zhrnuté v Tabuľke I.



Obr. 7. Výstupné charakteristiky vylepšeného Wilsonovho PZ



Obr. 8. Výstupné charakteristiky kaskódového PZ

Tabuľka I
POROVNANIE SIMULOVANÝCH A NAMERANÝCH HODNÔT V_{MIN} $[mV]$

Topológia PZ	Simulácia		Meranie	
	GD	BD	BD 300 mV	BD 320 mV
jednoduché	118	114	109	-
vylepšené Wil.	286	131	97	137
kaskódové	372	131	127	16

IV. CIELE DIZERTAČNEJ PRÁCE

- ✓ Analýza techník vhodných pre návrh nízko-napäťových IO a získanie najnovších poznatkov o vlastnostiach MOS tranzistorov riadených substrátovou elektródou.
- ✓ Využitie bulk-driven prístupu pri návrhu základných stavebných blokov pre IO s veľmi nízkym napájacím napätím využívaných v nanotechnológiách.
- ✓ Vyšetrenie a vyhodnotenie rizika vzniku a prípadného vplyvu nechceného latch-up javu na MOS tranzistory a navrhnuté obvodové bloky.
- ✓ Návrh testovacích štruktúr a vytvorenie návrhu topografie testovacieho čipu.
- Charakterizácia testovacích tranzistorov vyrobených na prototypovom čipe pomocou meraní pre návrh nízko-napäťových IO.
- Implementácia vylepšených simulačných modelov získaných z meraní do analógového návrhárskeho prostredia Cadence.
- Overenie presnosti a použiteľnosti spresneného modelu pre návrh nízko-napäťových IO pomocou meraní navrhnutých a vyrobených štruktúr a obvodov.

V. ZÁVER

V tomto príspevku boli analyzované metódy návrhu nízkonapätových IO prostredníctvom návrhu a implementácie základných topológií prúdových zrkadiel na báze MOS riadených substrátovou elektródou. Získané parametre boli porovnané s ich bežnými GD ekvivalentami. Simulácie a merania dokazujú, že použitím BD techniky je možné znížiť minimálne výstupné napätie vylepšeného Wilsonovho a kaskódového PZ na približne 30% oproti bežným PZ riadených hradlom. Merania odhalili horšiu presnosť simulačných modelov pri použití tranzistorov riadených substrátovou elektródou, a preto museli byť pri meraniach upravené napájacie DC podmienky. Vykonaný výskum a prezentované výsledky preukázali, že použitie techniky návrhu využívajúcej MOS tranzistory riadené substrátovou elektródou môže byť prínosom pre návrh nízkonapäťových analógových IO.

V rámci mojej doterajšej práce a výskumu vzniklo 9 publikácií (2 príspevky na medzinárodnom sympóziu DDECS, 3 príspevky na domácej konferencii ADEPT, 1 príspevok na medzinárodnej konferencii ICETA, 2 príspevky na doktorandskom seminári PAD a 1 článok v impaktovanom časopise).

POĎAKOVANIE

Táto práca bola podporená projektami APVV-15-0254 a VEGA 1/0905/17. Autor zároveň ďakuje STU za finančnú podporu v rámci Grantovej schémy na podporu mladých výskumníkov.

LITERATÚRA

[1] K. Bult, "Analog design in deep sub-micron CMOS," in *26th European Solid-State Circuits Conference*, Sept 2000, pp. 126–132.

[2] Y. Shouli and E. Sanchez-Sinencio, "Low voltage analog circuit design techniques: A tutorial," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 83, no. 2, pp. 179–196, 2000.

[3] P. Jespers, *The Gm/ID Methodology, a Sizing Tool for Low-voltage Analog CMOS Circuits: The Semi-empirical and Compact Model Approaches*. Springer Publishing Company, Incorporated, 2012.

[4] B. Razavi, *Design of Analog CMOS Integrated Circuits*, 1st ed. New York, NY, USA: McGraw-Hill, Inc., 2001.

[5] B. Blalock and P. Allen, "A low-voltage, bulk-driven mosfet current mirror for cmos technology," in *Circuits and Systems, 1995. ISCAS '95., 1995 IEEE International Symposium on*, vol. 3, Apr 1995, pp. 1972–1975 vol.3.

[6] S. Rajput and S. Jamuar, "Low voltage analog circuit design techniques," *Circuits and Systems Magazine, IEEE*, vol. 2, no. 1, pp. 24–42, First 2002.

[7] F. Khateb, D. Biolek, N. Khatib, and J. Vavra, "Utilizing the bulk-driven technique in analog circuit design," in *Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2010 IEEE 13th International Symposium on*, April 2010, pp. 16–19.

[8] B. Tupti and P. Pratik, "Simulation and analysis of bulk driven circuits for low power applications," *International Journal of Engineering and Technical Research*, vol. 2, February 2014.

[9] M. Rakús, V. Stopjaková, and D. Arbet, "Analysis of bulk-driven and dynamic-threshold current mirrors for low-voltage applications," in *Design and Diagnostics of Electronic Circuits and Systems (DDECS), 2017 IEEE 20th International Symposium*, April 2017, pp. 16–19.

[10] M. Rakús, V. Stopjakova, and D. Arbet, "Analysis of bulk-driven current mirrors in 130 nm CMOS technology," in *Advances in Electronic and Photonic Technologies (ADEPT)*, June 2017.

Zabezpečenie vnorených systémov s kritickou dobou odozvy proti poruchám

Vladimír Kunštár
1.ročník, denná forma
Školiteľ: Tibor Krajčovič

Slovenská technická univerzita v Bratislave, Fakulta informatiky a informačných technológií
Ilkovičova 2, 842 16 Bratislava 4 Slovenská Republika
vladimir.kunstar@stuba.sk

***Abstrakt*— Tento príspevok sa zaoberá analýzou kolízií pri RFID systémoch a detailnejšie rozoberá rôzne algoritmy používané na riešenie týchto kolízií. Ďalej sú v príspevku spomenuté aj mechanizmy zabezpečenia systému proti poruchám. Cieľom dizertačnej práce je návrh novej antikolíznej metódy a spojenie tejto metódy s mechanizmami zabezpečenia odolnosti proti poruchám a zabezpečenia z hľadiska bezpečnosti prenášania dát. V závere je uvedené ďalšie smerovanie práce a jej možné prínosy.**

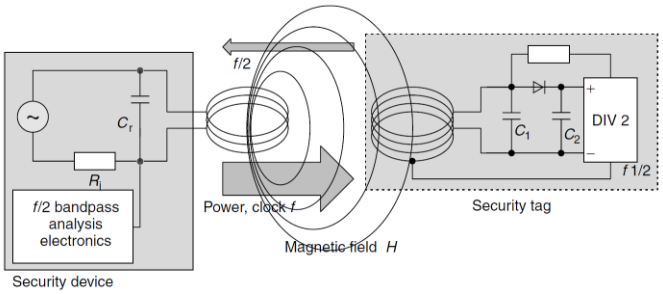
Kľúčové slová—RFID, kolízia, zabezpečenie, odolnosť, bezpečnosť

I. Úvod

Stále častejším používaním bezkontaktných technológií narastá potreba lepšieho a dokonalejšie zabezpečenia prenášaných dát. Ďalším nevyhnutným aspektom je aj zabezpečenie spoľahlivého prenosu týchto dát medzi jednotlivými zariadeniami. RFID (Radio Frequency Identification) je technológia pri ktorej je realizovaná komunikácia bezkontaktne a na krátku vzdialenosť. V RFID systéme existujú dva základné prvky. Prvým sú rádiofrekvenčné značky „tagy“, druhým sú rádiofrekvenčné čítačky. Tagy sú zväčša tvorené len integrovaným obvodom pripojeným k anténe a majú za úlohu uchovávanie statických informácií ako je napríklad identifikačné číslo. Okrem takéhoto typu existujú aj rozsiahlejšie typy, ktoré môžu vykonávať funkciu rôznych senzorov. Ďalšie delenia „tagov“ je podľa napájania. Ak je „tag“ napájaný vlastným zdrojom označuje sa ako aktívny. Pokiaľ je na napájanie použitá energia z čítačky označuje sa ako pasívny. K „tagom“ sa pristupuje bezkontaktne a na komunikáciu sa využíva elektromagnetická indukcia v spojení s amplitúdovou moduláciou. Čítačka vysiela prenosovú vlnu na frekvencií napríklad 13,56 MHz. Čítačka následne zachytí odrazenú vlnu so zmenenou intenzitou (Obr. 1) [12]. Frekvencia závisí od použitej RFID technológie. Prehľad niektorých RFID technológií uvádza tabuľka (Tab. 1). Pri takejto komunikácií nie sú však dáta pri prenose nijako zabezpečené a taktiež vznikajú rôzne rušenia a kolízie. Aby bola možná komunikácia je potrebné sa týmto kolíziám vyhnúť pomocou nato určených algoritmov.

Tab. 1.: Prehľad frekvencií k RFID

RFID technológia	Frekvencia	Komunikačná vzdialenosť
Proxy	125 kHz – 135 kHz	20 – 100 cm
Legic	13,56 MHz	10 – 50 cm
NFC	13,56 MHz	10 – 50 cm
Mifare	13,56 MHz	10 – 50 cm



Obr. 1.: Princíp komunikácie [3]

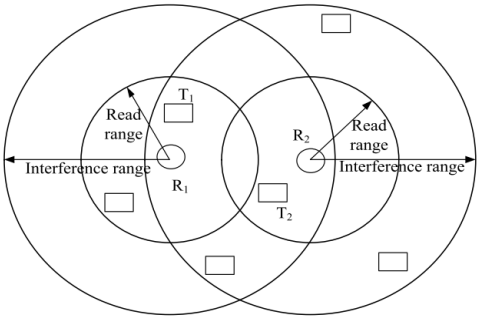
II. Súčasný stav

Podľa [2] pri aplikáciách s väčším počtom RFID zariadení dochádza k takzvanému problému kolízie čítačky RCP (Reader Collision Problem). Jedná sa o problém keď je vedľa seba pripojených viac čítačiek a signál z jednej čítačky interferuje so signálom s inými čítačkami. Tento problém možno klasifikovať do dvoch kategórií, prístup založený na plánovaní [1,4,9] a prístup založený na pokrytí [5]. Pri kategórií plánovania sa jedná o plánovanie dostupných zdrojov ako je frekvencia a čas. Tieto zdroje sú nastavené tak aby sa zamedzilo súčasnému vysielaniu z viacerých čítačiek. Tento prístup môže znížiť kolízie za cenu rýchlosti a spotreby. Pri prístupe založenom na pokrytí je rozsah susedných prekrývajúcich sa oblastí dynamicky prispôsobený na minimum. Táto realizácia si zväčša vyžaduje centrálny uzol,

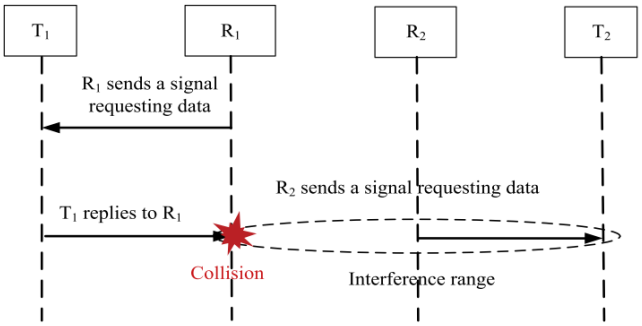
ktorý zabezpečuje výpočet vzdialenosti medzi susednými čítačkami. Takéto riešenie je zároveň zložitejšie na realizáciu a cenovo je nákladnejšie.

Pri niektorých aplikáciách môže byť vedľa seba umiestnených viacero čítačiek, ktoré majú byť schopné prečítať väčšie množstvo „tagov“. Umiestnenie týchto RFID prvkov vedľa seba môže mať za následok niekoľko typov kolízií RCP. Existujú tri typy kolízií RFID.

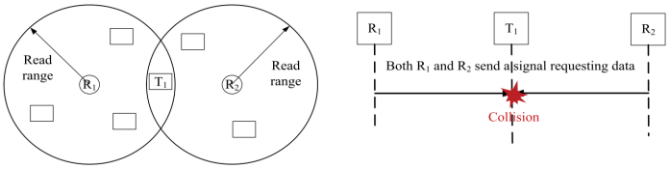
- **Kolízia „tag“, „tag“** (Tag to tag collision) – nastáva vtedy, keď čítačka odošle príkaz na načítanie ID do všetkých „tagov“ a viaceré „tagy“ naraz odošlú svoje ID čítačke [10]. Protokoly, ktoré riešia tento typ kolízie sú opísane v nasledujúcich publikáciách [8]. Tieto protokoly sú zvyčajne založené na protokole ALOHA alebo na báze stromov.
- **Interferencia čítačka, čítačka RRI** (Reader to reader interference) – pri tomto probléme je nevyhnutné rozlišovať medzi prenosovým rozsahom čítačky a interferenčným rozsahom. Na obrázku (Obr. 2) sú znázornené dve čítačky R1, R2 a dva „tagy“ T1, T2. Pri výkone 2W môže byť oblasť prenosového rozsahu v okruhu 10 metrov [11] a interferenčný rozsah môže dosahovať až 1000 metrov [8]. RRI sa vyskytuje práve vtedy, keď čítačky umiestnené v interferenčnom rozsahu pracujú na rovnakej frekvencií f. Keď sa R1 pokúša prečítať „tag“ T1 frekvenciou f a v tom istom čase s tou istou frekvenciou číta čítačka R2 dáta z „tagu“ T2, jej signál zasahuje do signálu odoslaného z T1 do R1 (Obr. 3). Kolízií sa dá vyhnúť tak, že R1 a R2 budú pracovať na inej frekvencií alebo v inom čase [8].
- **Interferencia čítačka, „tag“ RTI** (Reader to tag interference) – táto kolízia sa delí na ďalšie dva typy. Prvým typom je, keď sa viaceré čítačky pracujúce na rôznych frekvenciách pokúšajú prečítať „tag“ v spoločnom prenosovom rozsahu tak ako je to zobrazené na obrázku (Obr. 4). V takomto prípade T1 nedokáže rozpoznať príkazy z obidvoch čítačiek. Tejto kolízií sa dá vyhnúť tak, že R1 a R2 budú pracovať v rôznych časoch. Druhým typom je, keď viaceré čítačky pracujú na rovnakej frekvencií a „tag“ je umiestnený v prenosovom rozsahu jednej čítačky a interferenčnom rozsahu druhej čítačky. Na obrázku (Obr. 5) je znázornený príklad, kde sa čítačky R1 a R2 pokúšajú čítať T1 a T2 pomocou rovnakej frekvencie. Keďže sa T1 nachádza v interferenčnom rozsahu R2 a prenosovom rozsahu R1 oba signáli dosiahnu na T1 a kolízia sa prejaví na „tagu“. Tomuto typu kolízie sa dá predísť rôznymi frekvenciami čítačiek alebo rôznym pracovným časom čítačiek [8].



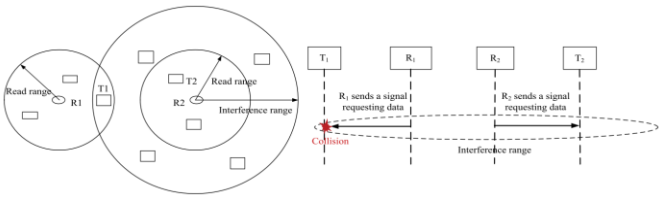
Obr. 2.: RRI kolízia



Obr. 3.: Znázornenie kolízie pri RRI



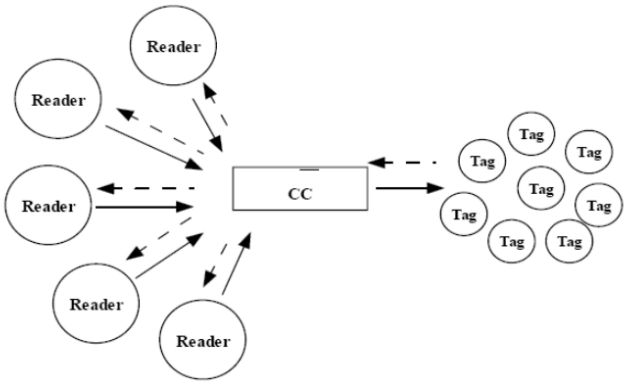
Obr. 4.: Prvý typ RTI



Obr. 5.: Druhý typ RTI

Pri čítaní toho istého RFID „tagu“ susednými čítačkami sú načítané dáta rovnaké. Žiaden z aktuálnych protokolov RCP neberie do úvahy túto duplicitu dát. Pokiaľ by rovnaké „tagy“ boli čítané jednou čítačkou a ďalej by boli zdieľané medzi ostatné čítačky, efektivita čítania v RCP by bola zvýšená. Na základe tohto zistenia bola navrhnutá nasledujúca CC-RFID (Central Cooperator RFID) architektúra (Obr. 6). Zavedením systému s centrálnym operátorom CC je súčasný problém komunikácie „viacerých bodov s viacerými bodmi“ MP2MP (Multiple Point to Multiple Point) zmenený na problém „viacerých bodov do jedného bodu“ MP2P (Multiple points to one Point), čo je bežný kolízny problém. Čítanie z viacerých

čítačiek by mohlo byť pomocou CC multiplexovaný a rovnaké informácie o „tagoch“ by mohli byť zdieľané susediacim čítačkám.



Obr. 6.: Architektúra CC-RFID systému [2]

A. Problém pridelenia frekvencie

Problém výberu frekvencie nastáva vo všetkých systémoch, ktoré používajú na komunikáciu rádiovú frekvenciu. Výber frekvencie závisí od účelu použitia zariadenia a technológie. Existujú dva hlavné typy, ktoré sa používajú na vyriešenie tohto problému. Prvým typom je fixná frekvencia alebo priradenie kanála (FCA). Pri tomto type je frekvencia pridelená fixne na statický súbor kanálov. Druhým typom je dynamické pridelenie kanálov (DCA). Na rozdiel od predchádzajúceho typu sa alokovaný súbor kanálov môže časom meniť. Kanály sú priradené v prípade ich potreby.

B. Riešenie problému pridelenia frekvencie

Existuje veľký počet algoritmov, ktoré sa zaoberajú problémom pridelenia frekvencie. Jednotlivé algoritmy sa od seba výrazne líšia v závislosti od typu problému, ktorý sa pokúšajú vyriešiť. Niektoré z algoritmov sa spoliehajú na centralizované riadenie pri prideľovaní kanála, iné sa zase spoliehajú na distribuované riadenie. Príkladom dynamického prideľovania kanálov sú neurónové siete alebo genetické algoritmy [6,7]. V neurónovej sieti sa vytvára matematický systém umelých neurónov. V tomto systéme predstavujú neuróny jednobunkovú základňovú stanicu a jeden kanál., tento kanál môže potom základňová stanica využívať. Genetické algoritmy sú formou slepého vyhľadávania. Jedným z riešení v genetickom algoritme je schéma priradenia pre všetky základňové stanice. Genetický algoritmus zaberá množinu riešení.

III. Zámer dizertačnej práce a jej ciele

Jedným zo zámerov dizertačnej práce je podrobnejšia analýza algoritmov využívaných pri kolíziách v RFID komunikácií. Analýza týchto algoritmov by mala pomôcť pri návrhu novej antikolíznej metódy. Nová metóda by mala byť zabezpečená aj z pohľadu bezpečnosti komunikácie a mala by byť odolná proti poruchám. Tézy dizertačnej práce sú:

- Špecifikácia požiadaviek potrebných pri návrhu jednotlivých metód,
- Návrh novej bezkolíznej metódy medzi RFID zariadeniami, prípadný návrh špecializovaného hardvéru potrebného pri tejto metóde,
- Návrh metódy na samočinnú opravu porúch vzniknutých pri komunikácií s RFID zariadeniami,
- Návrh metódy na bezpečný prenos dát medzi zariadeniami používanými pri komunikácií s ostatnými súčasťami systému, ktoré sú potrebné pri novo navrhnutých metódach,
- Implementácia a overenie jednotlivých metód na špecializovanom hardvéri,
- Overenie spoľahlivosti a účinnosti navrhnutých metód v praxi,
- Experimentálne testovanie vytvorených zariadení s implementovanými metódami.

IV. Záver

V tejto dobe prebieha podrobné štúdium a dôkladný výskum ohľadom fungovania jednotlivých antikolíznych metód, ktoré tvoria základ vývinu novej lepšej antikolíznej metódy. Tento článok sa zaoberá niektorými z nich a zároveň kolíziami, ktoré treba eliminovať. Do budúcna budú prebiehať testy, ktoré buď potvrdia alebo vyvrátia funkčnosť novej metódy podľa stanovených kritérií. Počas mojej doterajšej práce bol uverejnený článok v IIT.SRC 2017.

Vlastné publikácie

[1] KUNŠTÁR, Vladimír - Proximity based access control system with encryption. In IIT.SRC 2017, Student Research Conference [elektronický zdroj] : 13th Student Research Conference in Informatics and Information Technologies, Bratislava, Apríl 27, 2017. 1. vyd. Bratislava : Nakladateľstvo STU, 2017, ISBN 978-80-227-4342-6. Vnútrofakultná kategória: D

Poďakovanie

Tento článok bol vytvorený s podporou Ministerstva školstva, vedy, výskumu a športu Slovenskej republiky v rámci Operačného programu výskumu a vývoja pre projekt Univerzitný vedecký park STU Bratislava, ITMS 26240220084, spolufinancovaný Európskou komisiou. Fond regionálneho rozvoja. Túto prácu podporila aj Slovenská národná výskumná grantová agentúra v rámci projektu č. VG 1/0836/16.

Referencie

[1] BIRARI, S.M. - IYER, S. Mitigating the reader collision problem in RFID networks with mobile readers. In *2005 13th IEEE International Conference on Networks jointly held with the 2005 7th IEEE Malaysia International Conference on Communications, Proceedings* . 2005. Vol. 1, s. 463–468. .

[2] DONG, W. et al. A novel solution to the reader collision problem in RFID system. In *2006 International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM 2006* . 2007. s. 1–4. .

[3] FINKENZELLER, K. - WADDINGTON, R. *RFID Handbook*. . 2003. ISBN 0470844027.

[4] HO, J. et al. HiQ: A hierarchical Q-learning algorithm to solve the reader collision problem. In *Proceedings - 2006 Symposium on Applications and the Internet Workshops, SAINT 2006 Workshops* . 2006. Vol. 2006, s. 88–91. .

[5] KIM, J. et al. Effect of localized optimal clustering for reader anti-collision in RFID networks: Fairness aspects to the readers. In *Proceedings - International Conference on Computer Communications and Networks, ICCCN* . 2005. Vol. 2005, s. 497–502. .

[6] KUNZ, D. Channel Assignment for Cellular Radio Using Neural Networks. In *IEEE Transactions on Vehicular Technology* . 1991. Vol. 40, no. 1, s. 188–193. .

[7] LAI, W.K. - COGHILL, G.G. Channel Assignment for a Homogeneous Cellular Network with Genetic Algorithms. In *IEEE Transactions on Vehicular Technology* [online]. 1996. Vol. 45, no. 1, s. 91–96. Dostupné na internete: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=481825>.

[8] SAFA, H. et al. A distributed multi-channel reader anti-collision algorithm for RFID environments. In *Computer Communications* [online]. 2015. Vol. 64, s. 44–56. [cit. 2017-05-31]. . Dostupné na internete: <http://www.sciencedirect.com/science/article/pii/S0140366415000286>.

[9] WALDROP, J. et al. Colorwave: an anticollision algorithm for the reader collision problem. In *IEEE International Conference on Communications, 2003. ICC '03.* . 2003. Vol. 2, s. 1206–1210. .

[10] EPC ™ Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface. In . 2013. s. 1–152. .

[11] ETSI - European Telecommunications Standards Institute. In [online]. [cit. 2017-05-31]. Dostupné na internete: <http://www.etsi.org/>.

[12] RFID-Handbook - Active Load Modulation. In [online]. [cit. 2017-05-30]. Dostupné na internete: <http://rfid-handbook.de/about-rfid/active-load-modulation.html>.

Swatch Group creates the world's smallest Bluetooth chip

more than 5 million transistors
on a surface of about 5 mm2

Three world records in the "Swiss Silicon Valley" :
- the smallest Bluetooth chip on the market
- the lowest energy consumption compared to its competitors
- high-speed start-up capability is unparalleled

EM9304 has already been officially qualified to meet the latest Bluetooth standard, version 5.0.

ASICentrum®
A COMPANY OF THE SWATCH GROUP

EM MICROELECTRONIC®
A COMPANY OF THE SWATCH GROUP

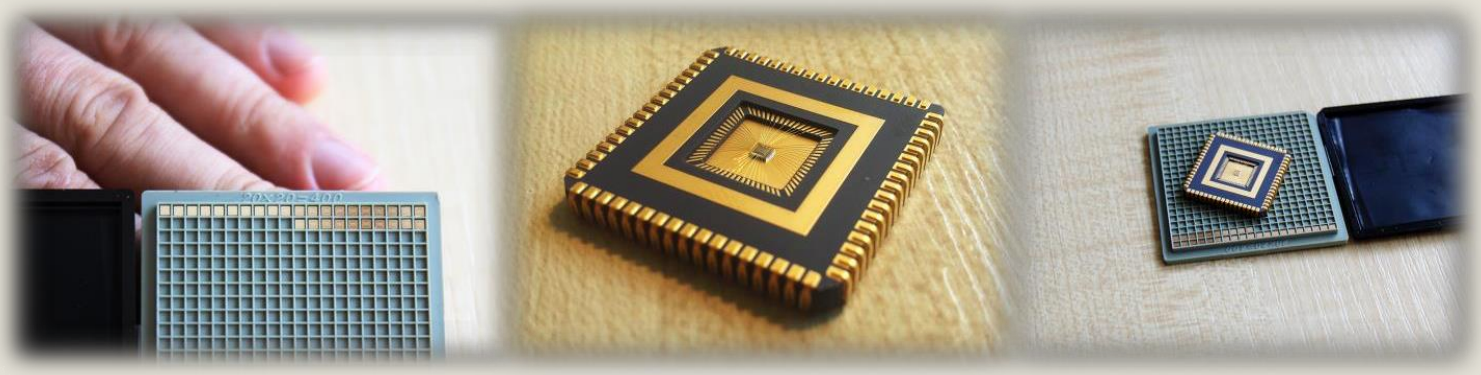
AGENTÚRA
NA PODPORU
VÝSKUMU A VÝVOJA

Projekt INSiDE

APVV-15-0254

http://deimos.elf.stuba.sk/projekty/inside/

Rozvoj a implementácia analógových integrovaných systémov pre ultra-nízkonapäťové aplikácie



Slovenská technická univerzita v Bratislave

Fakulta elektrotechniky a informatiky
Ústav elektroniky a fotoniky
Oddelenie návrhu a testovania integrovaných obvodov

STU FEI

INSiDE by ONTIO

ISBN 978-80-972784-0-3