Metodika návrhu řadiče rekonfigurace pro Systémy odolné proti poruchám

Richard Pánek
2. 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, 1223}
Email: {ipanek, kotasek}@fit.vutbr.cz

***Abstrakt*—Programovatelná hradlová pole (FPGA) jsou v dnešní době populární nejen pro vestavěné systémy. Jejich nevýhodou je náchylnost na sluneční aktivitu, která díky radioaktivnímu záření způsobuje poruchy konfigurační paměti známé jako SEU. Ty mohou způsobit selhání celého systému. Proto je vyvíjena řada metod pro zvýšení odolnosti proti poruchám. Pro FPGA je typické využití prostorové redundance např. TMR, která ale poruchy pouze maskuje. Proto je velmi vhodné využít klíčové schopnosti FPGA – rekonfigurace a tudíž moci poruchy opravit. Vše potřebné k opravě pomocí rekonfigurace musí zajistit její řadič. Ovšem existuje mnoho přístupů jak jej implementovat a proto se v rámci disertační práce zabývám jeho návrhem. Dále je představen nástroj pro odhad spolehlivosti systému založeného na TMR a rekonfiguraci. Nástroj je založený na simulaci systému s parametry MTTF a dobou rekonfigurace.**

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

I. Úvod

Nejen pro implementaci vestavěných systému jsou velmi populární programovatelná hradlová pole (*angl. Field Programable Gate Arrays*, FPGAs). Důvodem je cenová dostupnost při výrobě malých sérií oproti aplikačně specifickým integrovaným odvodům (*angl. Application-Specific Integrated Circuits*, ASICs) a vyšší rychlost výpočtu v porovnání s procesorovou implementací. Využití FPGA přináší i další výhody, těmi jsou flexibilita, možnost přeprogramování a tudíž změna funkcionality, nebo jednoduché prototypování apod. Klíčovou vlastností je možnost změnit konfiguraci i za běhu aplikace a tím docílit buď přizpůsobení se měnícím se podmínkám nebo možnost odstranění za běhu objevených poruch. Současná konfigurace daného FPGA je dána bitstreamem uloženým v jeho konfigurační paměti. Bitstream tedy určuje využití a propojení zdrojů FPGA, jako jsou vyhledávací tabulky LUT, flip-flops registry, paměti BRAM, atd. Ty jsou organizovány do programovatelných logických bloků (*angl. Configurable Logic Blocks*, CLBs) a propojeny pomocí programovatelné propojovací sítě. Nejpoužívanější jsou tzv. SRAM FPGA, jejichž konfigurační paměť je založena na paměťových buňkách SRAM. Ovšem díky tomu jsou náchylná na radioaktivní záření např. v podobě nabitých částic, které způsobuje překlopení bitů konfigurační paměti a tudíž poškození implementovaného obvodu. Tyto poruchy jsou známé pod pojmem *Single Event Upset* (SEU) a je potřeba s nimi počítat obzvláště při návrhu vesmírných aplikací, protože ty budou pod vlivem slunečního záření [11].

Existuje mnoho metod na zajištění zvýšení odolnosti proti poruchám a tedy dopadům SEU. Značná část z nich je založena na prostorové redundanci, ovšem je možné využít i časovou nebo datovou redundanci. Patrně nejznámější metodou je tří-modulová redundance (*angl. Triple Modular Redundancy*, TMR), která je základem pro značnou část dalších metod jako např. [1], kde byl navržen spolehlivější prvek určující majoritu. Autoři článku [4] kombinují prostorovou a časovou redundanci, tudíž redukovali prostorovou náročnost na úkor potřebného času na maskování poruchy. Článek [15] dělí využité LUT na SEU-senzitivní a SEU-nesenzitivní. Pak aplikuje TMR pouze na SEU-senzitivní LUT a tím zajistí snížení prostorové náročnosti na úkor nepatrného zhoršení spolehlivosti.

Samotná TMR je schopná poruchy pouze maskovat, tudíž při nashromáždění více poruch časem dojde k selhání celého systému. Proto je vhodné využít rekonfiguraci, která je schopná chybu opravit [14]. V takovém případě mluvíme o systému řízení odolnosti proti poruchám (*angl. Fault-tolerant Control System*, FTCS). Ten je složen ze tří základních částí:

- rekonfigurovatelného řízení – v našem případě FPGA,
- detekce a diagnostiky poruch,
- řadiče rekonfigurace (*angl. Reconfiguration Controller*, RC), který za základě diagnostických dat zajistí opravu poruchy.

Pro detekci je možné využít právě TMR s tím, že prvek určující majoritu musí být schopen informovat řadič částečné dynamické rekonfigurace o modulu s poruchou, který bude následně opraven pomocí rekonfigurace [3]. I tento model je předmětem dalšího zkoumání. Např. v článku [8] se věnovali plánování ověřování majority TMR a rekonfigurace detekovaných poruch s upřednostněním kritických prvků, aby zvýšili celkovou spolehlivost. Ovšem i samotný řadič rekon-

figurace lze implementovat různými způsoby. Autoři článku [5] využívají procesorovou implementaci řadiče rekonfigurace. Dále pro úsporu energie je detekce poruch zajištěna časovou redundancí. Článek [2] popisuje FTCS rozprostřený na více FPGA. Řadič rekonfigurace je v systému také několikrát. Jedná se buď o soft-core procesor na každém využitém FPGA, nebo externí komponentu. Tímto modelem je možné rekonfigurovat i samotné řadiče rekonfigurace v případě jejich poruchy. Další možností je implementace řadiče rekonfigurace přímo v hardware. Příkladem je řadič popsaný v článcích [6], [13]. Takový řadič může být buď na stejném FPGA jako zabezpečovaný obvod nebo na jiném.

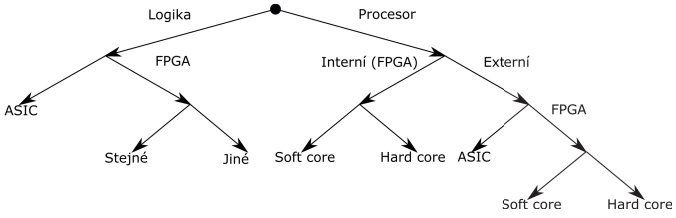
Tento článek je dále uspořádán následovně. Sekce II se věnuje definování řešené problematiky. V sekci III je popsáno vyhodnocení odhadu spolehlivosti systému odolného proti poruchám umístěného do FPGA a využívajícího částečnou dynamickou rekonfiguraci. Pro výpočet odhadované spolehlivosti je využita simulace. V sekci IV jsou rozpracovány cíle disertační práce. Závěrečné shrnutí je v sekci V.

II. Zaměření výzkumu

V rámci výzkumu se zaměřuji na návrh řadiče částečné dynamické rekonfigurace. Již z úvodu je patrné, že existuje mnoho způsobů jak jej implementovat:

- v logice nebo na procesoru,
- do FPGA (společně s obvodem nebo externí) nebo na ASIC,
- soft-core nebo hard-core procesor na FPGA.

Přehledné znázornění je na obrázku 1. Samozřejmě takových zobrazení může být více, záleží na volbě kořenového atributu. Jedná se o binární stromové uspořádání, kdy v kořenu je počáteční dělení a na listech konečné způsoby implementace. Ty dále mohou být na poruchy náchylné, nebo proti nim odolné.



Obrázek 1. Dělení způsobů implementace řadiče částečné dynamické rekonfigurace.

V rámci disertační práce budou diskutovány výhody a nevýhody jednotlivých přístupů. Z nich by mělo vyplynout, který přístup je vhodnější pro konkrétní navrhovaný systém odolný proti poruchám.

III. Simulační vyhodnocení odhadu spolehlivosti systému odolného proti poruchám na FPGA s využitím částečné dynamické rekonfigurace

Nástroj pro rychlé vyhodnocení využití rekonfigurace pro zajištění odolnosti proti poruchám byl představen v [10]. Zajímal nás dopad střední doby do výskytu poruchy (*angl. Mean Time To Failure*, MTTF) a doby potřebné pro opravu modulu pomocí rekonfigurace na celkovou spolehlivost celého systému. MTTF je dán prostředím, pro které je systém navrhován. Čas rekonfigurace lze ovlivnit velikostí rekonfigurovatelných modulů a také technologií (zvolením konkrétního FPGA).

Pro rychlé vyhodnocení byl vytvořen simulační nástroj postavený na knihovně *SimPy* [12], což je simulační framework založený na procesech a diskrétních událostech pro jazyk Python.

Pro experimentální systém jsme zvolili přístup TMR s rekonfigurací porouchaných jednotek. Každá jednotka může být v jednom ze dvou stavů: v poruchovém nebo bezporuchovém. Systém pracuje správně, pokud aspoň dvě jednotky TMR jsou v bezporuchovém stavu. V opačném případě dochází k selhání systému. Stav každé jednotky je možné změnit rekonfigurací nebo zásahem poruchy. V naší simulaci každá jednotka přejde do poruchového stavu v závislosti na MTTF. Konkrétní doba je určena normálním rozdělením, které je charakterizováno dvěma parametry [7]: *střední hodnota* (μ) a *rozptyl* (σ^2). Střední hodnotě odpovídá MTTF a rozptyl je dán na základě předchozích experimentů empiricky zjištěnou rovnicí 1.

$$\sigma^2 = \frac{\mu}{10} + 1\,[-] \tag{1}$$

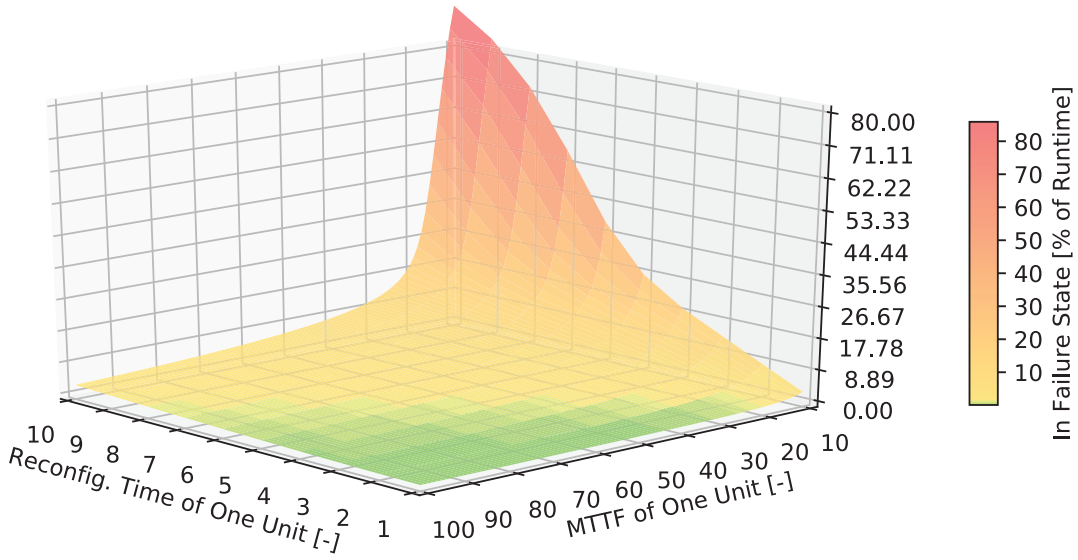
Výsledky experimentů s výše popsaným vyhodnocovacím prostředím jsou shrnuty v tabulce I. Doba potřebná k rekonfiguraci byla zvolena z intervalu $\langle 1, 10\rangle$. MTTF byla vybrána z intervalu $\langle 10, 100\rangle$. Tyto hodnoty byly zvoleny na základě monitorování skutečného experimentování s naším experimentálním elektromechanickým systémem (robotem v bludišti) [9]. Hodnoty jsou bezrozměrné, konkrétní rozměr záleží na výsledném systému, pro který budou využity. Čas běhu byl nastaven na 1000 jednotek a počet běhů jednoho scénáře byl 10 000. Jedním scénářem je myšlena jedna kombinace MTTF a doby rekonfigurace jednotky (jedna buňka tabulky).

Z výsledku v tabulce I je patrné, že největší pravděpodobnost selhání systému je při krátkém MTTF a dlouhé době rekonfigurace. Na opačné straně dlouhá MTTF a krátká doba rekonfigurace vede k nízké pravděpodobnosti selhání systému. Tyto výsledky byly očekávány, ovšem díky našemu simulačnímu nástroji mohou vývojáři snáze odhadnout, jak se jejich systém bude chovat i v jiném prostředí. Dále si mohou určit, jaká pravděpodobnost selhání je pro ně kritická a tudíž se rozhodnout, zda jimi navrhovaný systém se do nastavených hranic vejde. Výsledky mohou sloužit i jako základ pro rozhodnutí, zda snaha o zrychlení rekonfigurace bude mít dostatečný účinek na snížení pravděpodobnosti selhání.

Získané výsledky jsou také znázorněné v grafu na obrázku 2. Jedná se o znázornění totožných dat z tabulky I, ovšem z jiného pohledu. Z něj je patrný nelineární růst pravděpodobnosti selhání. Tudíž v určitých případech by bylo možné přijmout nepatrné zhoršení pravděpodobnosti selhání, ale zato využít méně náročný způsob rekonfigurace.

Tabulka I
PROCENTUÁLNĚ VYJÁDŘENÁ DOBA SELHÁNÍ SYSTÉMU BĚHEM PROVOZU ZÍSKANÁ POMOCÍ SIMULACE.

Failure State Representation [%]	Time To Reconfigure One Unit [-]									
MTTF of One Unit [-]	10.00	9.00	8.00	7.00	6.00	5.00	4.00	3.00	2.00	1.00
10.0	91.93	84.43	72.91	56.81	38.70	25.91	18.67	13.23	7.84	2.55
15.0	50.27	36.78	26.31	20.04	16.14	13.11	10.26	7.26	4.04	1.20
20.0	20.91	17.18	14.60	12.55	10.70	8.85	6.86	4.69	2.47	0.70
25.0	13.61	12.07	10.70	9.38	8.03	6.57	4.98	3.29	1.67	0.46
30.0	10.60	9.57	8.55	7.49	6.36	5.13	3.80	2.43	1.20	0.32
35.0	8.83	8.00	7.14	6.21	5.20	4.12	2.98	1.87	0.90	0.24
40.0	7.61	6.88	6.10	5.26	4.35	3.39	2.41	1.48	0.70	0.18
45.0	6.68	6.01	5.29	4.52	3.70	2.84	1.99	1.20	0.56	0.15
50.0	5.94	5.31	4.64	3.92	3.17	2.40	1.66	0.99	0.46	0.12
55.0	5.33	4.74	4.10	3.44	2.75	2.06	1.41	0.84	0.39	0.10
60.0	4.80	4.24	3.65	3.03	2.40	1.79	1.21	0.71	0.33	0.08
65.0	4.35	3.82	3.26	2.69	2.12	1.56	1.05	0.61	0.28	0.07
70.0	3.97	3.46	2.93	2.40	1.88	1.38	0.92	0.54	0.25	0.06
75.0	3.63	3.15	2.65	2.16	1.68	1.22	0.81	0.47	0.22	0.05
80.0	3.33	2.87	2.41	1.95	1.50	1.09	0.72	0.42	0.19	0.05
85.0	3.06	2.63	2.20	1.77	1.36	0.98	0.65	0.37	0.17	0.04
90.0	2.82	2.41	2.01	1.61	1.23	0.88	0.58	0.34	0.15	0.04
95.0	2.61	2.22	1.84	1.47	1.12	0.80	0.53	0.30	0.14	0.03
100.0	2.42	2.05	1.69	1.34	1.02	0.73	0.48	0.27	0.12	0.03



Obrázek 2. Graf doby selhání systému vyjádřené v procentech v závislosti na MTTF a době rekonfigurace každého jednotky.

IV. CÍLE DISERTAČNÍ PRÁCE

V rámci disertační práce se zaměřuji na vypracování metodiky návrhu a využití řadiče částečné dynamické rekonfigurace pro systémy odolné proti poruchám. Především se zaměřuji na dvě hlavní alternativy implementace řadiče v FPGA. První z nich je obvodová realizace a druhou pak program pro procesor, který je v FPGA. Dále budou vytvořena kritéria pro návrh, implementaci a samotné používání řadiče rekonfigurace. Ten bude následně implementován, aby s ním mohlo být experimentováno s cílem vyhodnotit míru splnění příslušných kritérií. Zatím známá kritéria pro posuzování jsou:

- Spolehlivost – odolnost proti poruchám, což je zásadní požadavek, protože o zvyšování odolnosti nám jde především.
- Rychlost a zpoždění, což souvisí s negativními dopady, které by mohlo přinést zvyšování odolnosti. Zajímá nás především, jestli využití rekonfigurace a potažmo jejího řadiče nebude mít za následek zvětšení zpoždění zabezpečované aplikace. I samotná rychlost rekonfigurace by mohla ovlivnit zabezpečovaný systém.
- Spotřeba, protože přidáním dalších komponent s velikou pravděpodobností naroste, ovšem záleží do jaké míry.

Obzvláště důležité je toto kritérium pro mobilní zařízení, které musí být napájené z baterií. V takovém případě úzce souvisí i s životností systému, protože i míra zabezpečení závisí na době, po kterou musí zařízení být plně funkční. Pokud by se měla energie, která je pro systém vyhrazená, vyčerpat dříve, než nastane jistá porucha, pak je zbytečné mít zabezpečení, které je na ni připravené a současně spotřebovává další energii.

- Zabraná plocha na FPGA se také zvětší, ale bude zkoumáno do jaké míry. Samozřejmě čím více FPGA zdrojů bude zapotřebí, tím větší a také dražší FPGA bude vyžadováno. Další možností může být využití více FPGA čipů. Vše povede na nárůst ceny. Zvětšení využité plochy FPGA také může zvýšit pravděpodobnost, že bude systém poruchou zasažen.
- Zabezpečení samotného řadiče, aby byl odolný proti poruchám. S tím souvisí vyhodnocení, jaké jsou možnosti pro zabezpečení řadiče a jaké budou dopady na zabezpečovaný systém. Bude zapotřebí vyhodnotit také všechna ostatní kritéria, protože i ta budou ovlivněna.

Další kritéria mohou být identifikována v průběhu výzkumu. Je zřejmé, že jsou vzájemně protichůdná a tak předpokládám vznik různých paretooptimálních řešení, která budou v rámci metodiky diskutována. Zejména jejich přínos pro různé požadavky aplikací.

V rámci výzkumné skupiny byly již mými předchůdci položeny základy pro využití rekonfigurace pro systémy odolné proti poruchám. Mým cílem je využití těchto základů pro moji práci a dále je rozvíjet. Pokračuji proto s vývojem řadiče částečné dynamické rekonfigurace GPDRC [6], [13]. Tento řadič budu dále zabezpečovat pomocí TMR a také budu zkoumat možnosti auto-rekonfigurace, tedy možnosti, že by se řadič dokázal po poruše sám opravit pomocí rekonfigurace. Dále pro porovnání počítám také s vytvořením implementace pro procesor a jejím zabezpečením stejnými postupy jako předchozí verzi. Všechny tyto přístupy budou podrobeny experimentům a budou diskutovány přínosy a úskalí, které budou potřeba pro vypracování metodiky.

Uvažovaná metodika má za cíl pomoci s výběrem ideálního řadiče rekonfigurace FPGA pro zajištění odolnosti proti poruchám výsledné aplikace tak, aby byly požadavky na ni kladené splněny co nejlépe.

V. Závěr

V rámci tohoto článku byla nastíněna problematika řešená v moji disertační práci. Jedná se metodiku návrhu řadiče rekonfigurace pro systémy odolné proti poruchám. Samotný řadič částečné dynamické rekonfigurace je klíčová komponenta pro zvýšení odolnosti proti poruchám. Ovšem může být implementována různými způsoby jako např. v FPGA (logika, procesor – sof-core, hard-core), nebo externí součástka: procesor, jiné FPGA atd. V rámci zamýšlené metodiky budou jednotlivé přístupy porovnány, aby bylo jednodušší rozhodnout, jaký typ řadiče zvolit při návrhu nového systému odolného proti poruchám.

Dále byl představen nástroj na vyhodnocení přínosu rekonfigurace založený na simulaci. Díky němu je možné odhadnout pravděpodobnost selhání konkrétního systému. Náš nástroj může být užitečný pro návrháře, protože budou moci odhadnout, zda jimi navrhované řešení je dostatečně spolehlivé pro dané prostředí.

Poděkování

Tato práce byla podporována projektem JU ECSEL SECREDAS (Product Security for Cross Domain Reliable Dependable Automated Systems), grantová dohoda č. 783119 a projektem řešeným na FIT VUT v Brně pod číslem FIT-S-17-3994.

Reference

[1] Afzaal, U.; Lee, J. A.: FPGA-based design of a self-checking TMR voter. *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, Sept 2017, s. 1–4, doi:10.23919/FPL.2017.8056811.

[2] Bolchini, C.; Fossati, L.; Codinachs, D. M.; aj.: A Reliable Reconfiguration Controller for Fault-Tolerant Embedded Systems on Multi-FPGA Platforms. *2010 IEEE 25th International Symposium on Defect and Fault Tolerance in VLSI Systems*, Oct 2010, ISSN 1550-5774, s. 191–199.

[3] 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)*, Sept 2007, ISSN 1550-5774, s. 87–95.

[4] Bělohoubek, J.; Fišer, P.; Schmidt, J.: Error masking method based on the short-duration offline test. *Microprocessors and Microsystems*, ročník 52, 2017: s. 236 – 250, ISSN 0141-9331, doi: https://doi.org/10.1016/j.micpro.2017.06.007.

[5] Frenkel, C.; Legat, J. D.; Bol, D.: A Partial Reconfiguration-based Scheme to Mitigate Multiple-Bit Upsets for FPGAs in Low-cost Space Applications. *2015 10th International Symposium on Reconfigurable Communication-centric Systems-on-Chip (ReCoSoC)*, June 2015, s. 1–7.

[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] Natrella, M.: *NIST/SEMATECH e-handbook of statistical methods*. NIST/SEMATECH, 2010.

[8] Nguyen, N. T. H.; Agiakatsikas, D.; Cetin, E.; aj.: Dynamic scheduling of voter checks in FPGA-based TMR systems. *2016 International Conference on Field-Programmable Technology (FPT)*, Dec 2016, s. 169–172, doi:10.1109/FPT.2016.7929525.

[9] Podivinsky, J.; Lojda, J.; Cekan, O.; aj.: Reliability Analysis and Improvement of FPGA-Based Robot Controller. *Digital System Design (DSD), 2017 Euromicro Conference on*, IEEE, 2017, s. 337–344.

[10] Pánek, R.; Lojda, J.; Podivínský, J.; aj.: Partial Dynamic Reconfiguration in an FPGA-based Fault-Tolerant System: Simulation-based Evaluation. *Submitted to: IEEE East-West Design & Test Symposium*, 2018.

[11] 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

[12] Team SimPy: SimPy: Discrete Event Simulation for Python. https://simpy.readthedocs.io/, 2017, accessed: 2018-06-10.

[13] 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.

[14] Zhang, Y.; Jiang, J.: Bibliographical Review on Reconfigurable Fault-tolerant Control Systems. *Annual Reviews in Control*, ročník 32, č. 2, 2008: s. 229 – 252, ISSN 1367-5788, doi: https://doi.org/10.1016/j.arcontrol.2008.03.008.

[15] Zheng, M. S.; Wang, Z. L.; Tu, J.; aj.: Reliability Oriented Selective Triple Modular Redundancy for SRAM-Based FPGAs. *Applied Mechanics and Materials*, ročník 713, Trans Tech Publ, 2015, s. 1127–1131.