METODIKA NÁVRHU SYSTÉMŮ ODOLNÝCH PROTI PORUCHÁM DO OMEZENÉHO IMPLEMENTAČNÍHO PROSTORU NA BÁZI FPGA

Lukáš Mičulka
Výpočetní technika a informatika, 2. ročník, prezenční studium
Vedoucí: doc. Ing. Zdeněk Kotásek, CSc.

Vysoké učení technické v Brně
Fakulta informačních technologií
Božetěchova 2, Brno, 612 66, Česká republika

imiculka@fit.vutbr.cz

Abstrakt. V článku je popisována navrhovaná metodika, která se zabývá možností rekonfigurace obvodu FPGA po vzniku přechodné i trvalé poruchy. Metodika se zabývá postupy pro rozlišení typu poruchy, lokalizaci postižené oblasti FPGA a pro výběr nové konfigurace v případě výskytu trvalé poruchy. Obsahem jsou i možné způsoby řešení problému synchronizace jednotek po částečné dynamické rekonfiguraci FPGA.

Klíčová slova. Systémy odolné proti poruchám, rekonfigurace, synchronizace, FPGA.

1 Úvod

Tato práce se zabývá návrhem systémů odolných proti poruchám (Fault Tolerant - FT) na bázi programovatelných hradlových polí (Field-Programmable Gate Array - FPGA). Jejich výhodou je rychlost blížící se klasickým aplikačně specifickým integrovaným obvodům (ASIC) ale zároveň možnost změny vykonávané funkce. Využití je proto vhodné například při vytváření prototypů obvodů či pro použití v oblastech, kde je nutný rychlý a jednoduchý vývoj. V současné době jsou FPGA stále více využívána i díky schopnosti dynamické rekonfigurace. Existence konfigurační paměti u FPGA má ovšem i své nevýhody, kterými jsou především nové typy poruch, které se u ASIC nevyskytují.

Zabývat se možnými poruchami v FPGA je v dnešní době nutné, protože ty se vyskytují stále častěji s tím, jak dochází k rychlému rozvoji číslicových obvodů. Je totiž používána stále jemnější technologie výroby, využívá se menšího napájecího příkonu a číslicové obvody se vystavují stále těžším podmínkám provozu, kdy jsou na ně zároveň kladeny stále vyšší nároky na spolehlivost. Pro řešení výskytu poruch se používají tři metodiky: předcházení chybám (Fault Avoidance), maskování chyb (Fault Masking) a eliminování vlivu chyby (FT - Fault Tolerance), které se věnuji ve své práci.

Existuje několik metod jako replikace a ztrojení funkčních jednotek (Triple Modular Redundancy - TMR), které se využívají k zajištění bezchybného běhu systému i v případě poruchy v obvodu. Při využití FPGA je možné je rozšířit o nové přístupy. Zejména lze využít vlastnosti dynamické rekonfigurace, kterou lze danou FT architekturu za běhu modifikovat. Systém tedy může při detekované poruše produkovat správné hodnoty na výstupech a zároveň může být porouchaná jednotka rekonfigurována.

Cílem práce je navržení metodiky, která bude řešit výskyt jak přechodné poruchy (opravitelná rekonfigurací postiženého bloku FPGA), tak i trvalé poruchy, která způsobí omezení implementačního prostoru v FPGA. Pro řešení problému nedostatečného prostoru pro novou konfiguraci bude využito principu, kdy nová konfigurace vkládaná do FPGA bude obsahovat méně prostorově náročnou (jednodušší) implementací FT zabezpečení obvodu. Funkčnost obvodu ale zůstane plně zachovaná.

2 Systémy odolné proti poruchám využívající dynamickou rekonfiguraci

FPGA poskytuje pro architektury odolné proti poruchám vhodné implementační prostředí, jelikož kromě tradičních FT metod nabízí i možnost rekonfigurace implementované architektury.

2.1 Detekce a lokalizace poruch

Při použití architektury odolné proti poruchám při konstrukci obvodu se celý systém nejenže dokáže vypořádat s poruchou v obvodu, ale je možné jej vybavit prostředky pro lokalizaci chybně pracující jednotky. Tato schopnost je nutná pro opravu postižené jednotky.

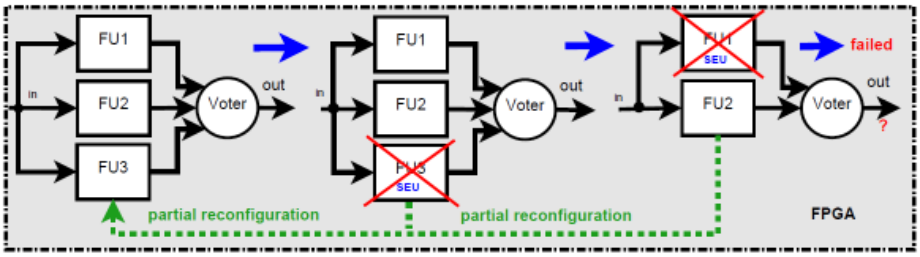
Pro lokalizaci poruchy u replikované jednotky architektury odolné proti poruchám lze využít hlídací obvod, který v případě produkce chybných hodnot na výstupech označí jednotku jako porouchanou. Výhodou použití (více) hlídacích obvodů je schopnost odhalení více poruch v obvodu najednou.

Lokalizace chybně pracující jednotky může být prováděna i přímo v hlasovacím obvodu. Například u architektury TMR lze na základě majority určit, které jednotky pracují správně. Jednotka, jejíž výstupy se liší od většinového výsledku, je označena jako nefunkční.

Kromě zmíněných postupů existují i celá řad dalších metod, jako například postupné offline testy malých částí obvodu (Self-Testing Areas - STAR) [1].

2.2 Částečná dynamická rekonfigurace

Moderní FPGA obsahují rozhraní, které umožňuje označené rekonfigurovatelné bloky (PRM – Partial Reconfiguration Modules) FPGA rekonfigurovat jednotlivě a to bez přerušení činnosti ostatních bloků. Tato činnost je označovaná jako částečná dynamická rekonfigurace (PDR – Partial Dynamic Reconfiguration). PRM jsou prostorově disjunktní, mohou spolu komunikovat prostřednictvím proxy logiky umístěné na jejich rozhraních. Konfigurace pro PRM je uložena jako bitová posloupnost (bitstream) a její velikost je dána počtem rekonfigurovaných bloků. Mnoho prací [2][3][4] se věnuje využití PDR u FPGA pro zlepšení jeho odolnosti proti poruchám. Princip použití rekonfigurace k obnově původního TMR schématu k zajištění odolnosti proti poruchám je znázorněn na obrázku 2.1.



Obr. 2.1 - Využití částečné rekonfigurace v architektuře odolné proti poruchám

2.3 Řešení výskytu trvalé poruchy

Pokud se v obvodu vyskytne trvalá porucha, původní počet využitelných zdrojů a tím i implementační prostor v poli FPGA se zmenší. Počet zdrojů, které nelze po výskytu poruchy použít, je dán především tím, s jakou přesností lze poruchu lokalizovat. Uvažujeme-li architekturu odolnou proti poruchám využívající replikované jednotky a hlídací obvody, je zpravidla nejmenším lokalizovatelným místem výskytu poruchy právě jedna replikovaná jednotka. Přesnost lokalizace poruchy je tedy dána zvolenou granularitou při návrhu odolné architektury.

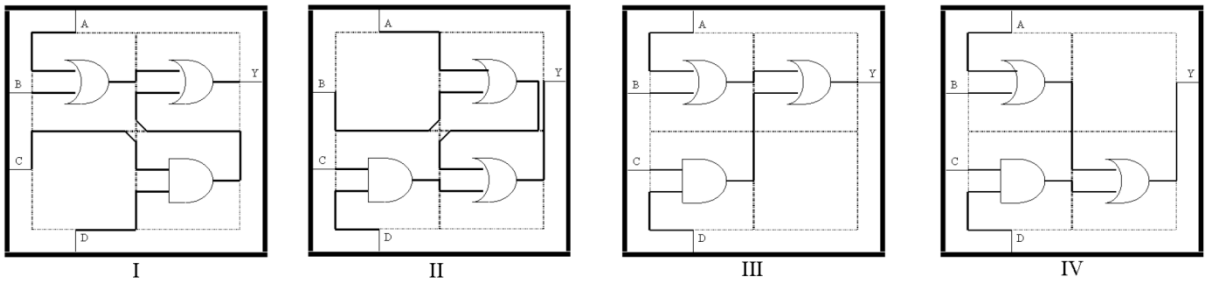
2.3.1 Inkrementální změna návrhu

Tato metoda [5] vychází z částečné změny návrhu a využívá procesor i s operačním systémem, kde probíhá proces umísťování hradel a jejich propojování v rámci FPGA. To může představovat velkou

náročnost na zdroje. Existuje však i varianta s využitím procesoru, který je používán v rámci aplikace běžící v FPGA, a to v době, kdy by jinak nebyl využit. Problémem je zde především zabezpečení odolnosti procesoru proti poruchám a také nutnost řešit problém možné dlouhé latence při modifikaci konfigurace, jelikož proces umísťování a propojovaní hradel může trvat dlouhou dobu a nemusí řešení najít. Dále je nutné, aby procesor dokázal zjistit z konfigurační bitové posloupnosti strukturu obvodu. Význam jednotlivých bitů posloupnosti z hlediska struktury obvodu bývá u komerční FPGA často neznámý. Výhodou je naopak efektivní využití zbývající části FPGA nepostižené poruchou.

2.3.2 Využití předkompilovaných konfigurací

Další metodou, která se zabývá opravou obvodu při výskytu trvalé poruchy, je využití tzv. generací konfigurací pro daný PRM, které jsou předem zkompilovány [6][7]. PRM je rozdělen na díly, které jsou navzájem disjunktní. Základem metody je vytvoření různě modifikovaných konfigurací pro daný PRM v FPGA, v němž je vždy jistý díl nevyužit. V případě, že je detekována trvalá porucha a je lokalizována, dochází k rekonfiguraci postiženého PRM tou konfigurací, která díl s poruchou nevyužívá. Příklad jedné generace konfigurací pro PRM nevyužívající vždy jeden díl kvůli trvalé poruše je zobrazen na obrázku 2.2. Jako vhodné se jeví použít tento princip u architektur odolných proti poruchám. Zatímco po počáteční konfiguraci je v PRM umístěn jeden typ architektury, po vzniku trvalé poruchy a vlivem omezení implementačního prostoru může být jako další konfigurace vybrána jiná s jednodušší architekturou odolnou proti poruchám, která postižený díl PRM nevyužívá.



Obr. 2.2 – Generace konfigurací pro PRM s jednou trvalou poruchou [6]

Metoda výběru nové konfigurace tedy vypadá následovně. Na počátku je detekována trvalá porucha a lokalizován díl PRM, v kterém se nachází. V rámci lokalizace poruchy je nutné zjistit postižený díl PRM. To lze určit na základě znalosti rozmístění replikovaných jednotek do dílů nebo pomocí lokalizování chyby přímo v bitové posloupnosti konfigurace při znalosti její struktury. Po lokalizování se řídicí jednotka rekonfigurace pokusí vybrat konfiguraci nevyužívající díl s poruchou z aktuální generace konfigurací, která používá stejnou FT architekturu. Pokud taková konfigurace neexistuje, pokračuje se s volbou v generaci, která má architekturu jednodušší a zabírá tak menší počet dílů. Následně je celé PRM touto konfigurací zrekonfigurováno. Takto lze vybírat i při dalších trvalých poruchách v PRM. V poslední generaci bude již pouze obvod bez redundance, nebude tedy odolný proti poruchám.

Nevýhodou popsaného řešení je nutnost uchovávání množství dat jednotlivých konfigurací v nevolatilní paměti. Částečně lze tento problém řešit efektivnějším výběrem konfigurací do generací a také použití komprese pro konfigurační data. Přínosem oproti výše uvedené metodě je pak menší náročnost na zdroje FPGA a latence obvodu, jelikož není nutné generovat konfigurace za běhu.

2.4 Synchronizace replikovaných modulů ve FT systému po dynamické rekonfiguraci

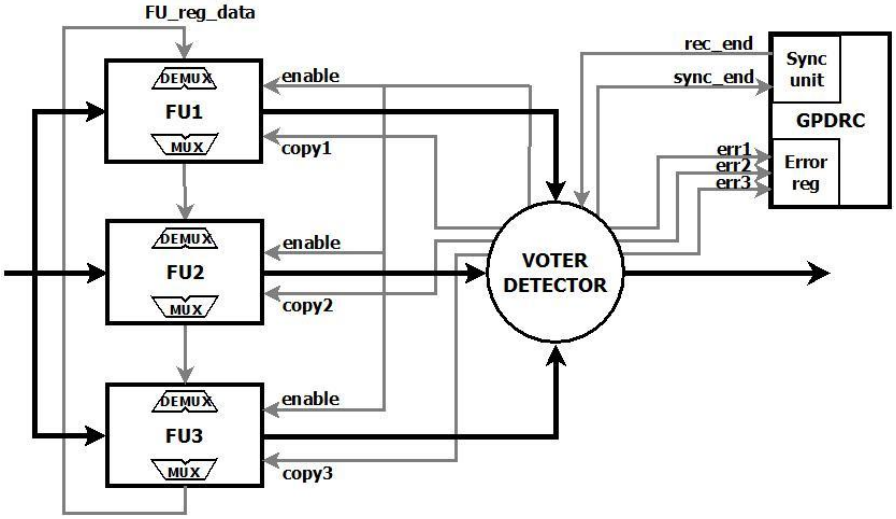
V rámci řešení opravy FT systému v FPGA využívajícího TMR pomocí částečné dynamické rekonfigurace vyvstává problém se synchronizací replikovaných obvodů. Toto se týká systémů implementujících sekvenční obvody, které obsahují stav systému, který je využíván pro další výpočet. Zatímco replikované funkční jednotky, které nebyly postižené poruchou, neustále provádí výpočet dále, jednotka původně postižená poruchou, na které proběhla rekonfigurace, se bude obecně nacházet v

nedefinovaném stavu, a tudíž je stále nefunkční. Samotná rekonfigurace tedy nestačí, jednotky je nutné synchronizovat, aby pokračovaly ve výpočtu od stejného vnitřního stavu. Další problém může nastat v situaci, kdy hlídací obvod (případně jiný obvod sloužící k lokalizaci jednotky s poruchou) ohlásí jednotku ihned po dokončení rekonfigurace jakou nefunkční a řadič rekonfigurace se jí pokouší rekonfigurovat znovu. Řešení tohoto problému záleží na typu obvodu a jeho složitosti.

V případě sekvenčních obvodů určených pro zpracování paketů, jsou jednotky samy synchronizovány generováním jejich lokálního resetu. Smyslem synchronizace je v jejich případě zabránit další rekonfiguraci již jednou rekonfigurované jednotky před příchodem lokálního resetu. Řešení pomocí vyčkávání jisté doby do další rekonfigurace stejné jednotky bylo prezentováno v [8].

V případě synchronizace sekvenčních obvodů implentujících konečný automat vhodného typu lze použít tzv. checkpointů, kdy stav zrekonfigurované jednotky je nastaven do určitého (většinou nejčastěji dosahovaného) stavu. Činnost jednotky je zastavena, dokud zbývající replikované jednotky nedosáhnou stejného stavu. Toto řešení z [9] je však použitelné jen u specifických automatů.

Komplexnější řešení pro všechny typy sekvenčních obvodů představuje metoda zkopírování stavu, který je uložen ve vnitřních registrech, ze správně fungující jednotky do jednotky po rekonfiguraci. Na základě prezentované metody [10] byla vyvinuta architektura publikovaná v [11] využívající pokročilý hlasovací obvod se schopností lokalizovat porouchanou jednotku TMR (viz obr. 2.3). O její rekonfiguraci se stará řadič částečné dynamické rekonfigurace, který byl prezentován v [8] a [12]. Po dokončení následuje kopírování stavu do zrekonfigurované jednotky řízené z hlasovací jednotky.



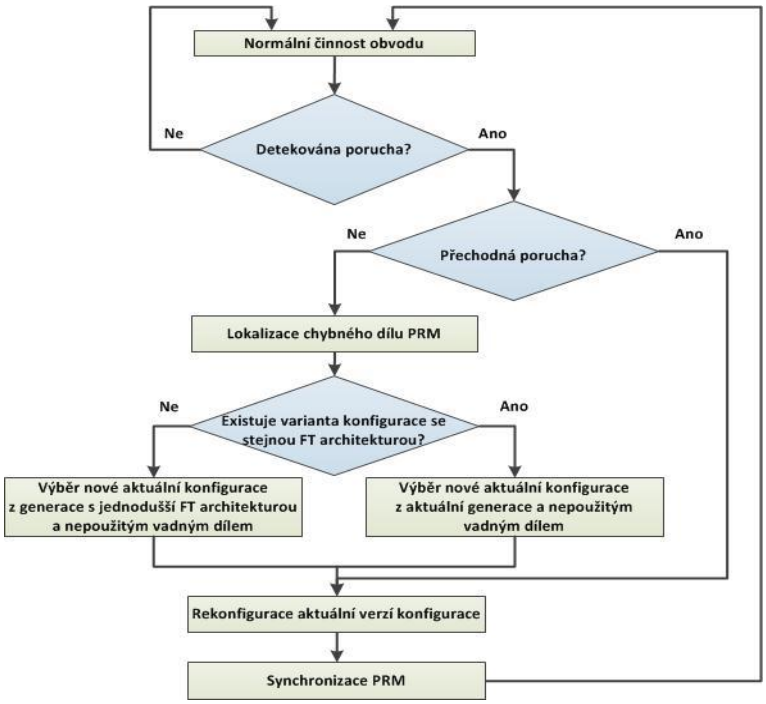
Obr. 2.3 – Generace konfigurací pro PRM s jednou trvalou poruchou [10]

3 Metodika pro návrh systému odolných proti poruchám

Základním principem zvoleného řešení je rozdělení návrhu aplikace v FPGA na části, u kterých bude zvlášť navržena architektura odolná proti poruchám umístěná v jednom PRM a schopná lokalizovat postiženou jednotku. Tato informace bude sloužit řadiči částečné dynamické rekonfigurace, který bude mimo jiné rozhodovat, o jaký typ poruchy se jedná. Pokud i po několika opakování rekonfigurace jednotky bude porucha stále přítomná, bude tato označena jako trvalá. Podle jejího typu bude poté provedena odpovídající oprava. V případě přechodné poruchy bude následovat rekonfigurace aktuálně používanou verzí konfigurace. U trvalé bude nejprve lokalizován díl, v kterém se nachází funkční jednotka s poruchou. V tomto okamžiku se řadič pokusí najít konfiguraci, která tento díl nevyužívá, a bude ze stejné generace konfigurací. Pokud taková nebude nalezena, bude prohledávána generace další. Vybraná konfigurace bude nyní aktuální a bude použita pro rekonfiguraci PRM. Po rekonfiguraci musí následovat synchronizace. Fáze činnosti při výskytu poruchy jsou na obrázku 3.1.

Pro jednotlivé PRM v FPGA, které mají být odolné vůči jistému počtu trvalých poruch, budou k dispozici v externí paměti předkompilované konfigurace. PRM budou rozděleny na díly s jistým počtem CLB. V jednom dílu smí být umístěna maximálně jedna funkční jednotka či voter z architektury odolné proti poruchám s hlídacím obvodem. Tímto způsobem je možné v případě poruchy, kterou ohlásí hlídací obvod, lokalizovat díl, jehož fyzická poloha v PRM je známa díky systematickému umísťování jednotek do dílů v průběhu mapovaní architektury do FPGA.

Na počátku je funkční jednotka navržena v architektuře s nejlepší odolností proti poruchám, která ještě splňuje zadané požadavky na počet obsazených zdrojů, spotřebu obvodu, odpadní teplo apod. Různé rozdělení funkčních jednotek do jednotlivých dílů PRM tohoto typu architektury odolné proti poruchám tvoří jednu generaci konfigurací. Další generace se od předchozí liší tím, že počet obsazených dílů je vždy minimálně o jeden menší. Zmenšení nároku na počet obsazených dílů PRM lze dosáhnout zejména přechodem na jiný jednodušší typ architektury odolné proti poruchám, jako například přechodem od TMR architektury s hlídacími obvody na duplexní architekturu.



Obr. 3.1 – Vývojový diagram činnosti při výskytu poruchy

4 Cíle disertační práce a plán další práce

4.1 Cíle disertační práce

V předchozích kapitolách byla uvedena motivace k výzkumu a současný stav poznání, na základě čehož byly stanoveny tyto cíle disertační práce:

- vypracování metodiky pro návrh systémů odolných proti přechodným poruchám a schopné pracovat i při výskytu určitého počtu trvalých poruch
- návrh metodiky generování konfigurací pro FPGA pro použití v rámci metody předkompilovaných konfigurací pro vytváření nové implementace obvodu do prostoru omezeného trvalou poruchou
- využití řadiče částečné dynamické rekonfigurace pro odstranění přechodné poruchy nebo lokalizaci trvalé poruchy a změnu konfigurace v postižené PRM
- návrh synchronizačního mechanizmu pro rekonfigurované jednotky
- pokrytí výskytu více trvalých poruch v jednom díle PRM

4.2 Plán další práce

Další práce bude zaměřena na zapojení řadiče rekonfigurace prezentovaného v [12] do procesu rozlišení typu poruchy, opravy obvodu v FPGA po výskytu trvalé poruchy a synchronizace jednotek po rekonfiguraci. Následovat by mělo navržení metodiky pro generování generací konfigurací a pro výběr jedné z nich podle jistého klíče (architektura mající nejlepší vlastnosti z pohledu odolnosti proti

poruchám), která nevyužívá bloky s trvalou poruchou. Tento výběr může vypadat tak, že se nejprve hledá konfigurace se stejnou architekturou odolnou proti poruchám, a pokud to není z důvodu nedostatku zdrojů možné, je vybrána jedna z jednodušších architektur vyžadujících méně zdrojů. Situaci výběru budou také komplikovat poruchy, které se objeví v již dříve rekonfigurované PRM. S počtem trvalých poruch v PRM bude klesat počet různých konfigurací pro ně určených. Dojde tak k významnému snížení počtu předkompilovaných konfigurací a tím snížení potřebné velikosti paměti pro jejich uložení.

Poděkování

Tato práce je podporována projektem MŠMT COST LD12036 –„Metodiky pro návrh systémů odolných proti poruchám do rekonfigurovatelných architektur - vývoj, implementace a verifikace“ a výzkumným projektem MSM 0021630528 – „Výzkum informačních technologií z hlediska bezpečnosti“ a grantem „BUT FIT-S-11-1“.

Reference

[1] Emmert, M.J., Stroud C.E., Cheatham, J., Taylor, M.A., Kataria, P., Abramovici, M.: Performance Penalty for Fault Tolerance in Roving STARs, *Proceedings of 10th International Workshop on Field-Programmable Logic and Application*, Springer, 2000, s. 545-554.

[2] Bolchini, C., Miele, A., Santambrogio, M. D.: TMR and Partial Dynamic Reconfiguration to mitigate SEU faults in FPGAs, *Procedings of 22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, IEEE Computer Society, 2007, s. 87-95.

[3] F. L. Kastensmidt, G. Neuberger, L. Carro, and R. Reis, Designing and testing fault-tolerant techniques for sram-based fpgas, *Proceedings of the 1st conference on Computing frontiers*. New York, NY, USA: ACM, 2004, s. 419–432.

[4] Straka, M., Kaštil, J., Kotásek, Z., Mičulka, L.: Fault Tolerant System Design and SEU Injection Based Testing, *Microprocessors and Microsystems*, roč. 2012, č. 01, Amsterdam, NL, s. 16, ISSN 0141-9331

[5] Emmert, J.M., Bhatia, D.: Incremental Routing in FPGAs, *Proceedings of 11th Annual IEEE International Conference of ASIC, IEEE Computer Society*, 1998, s. 217-221.

[6] Lach, J., Mangione W.H., Potkonjak, M.: Algorithms for Efficient Runtime Faulty Recovery on Diverse FPGA Architectures, *Proceedings on Defect and Fault Tolerance, IEEE Computer Society*, 1999, s. 386-394.

[7] Huang, W.-J., McCluskey, E.J.: Column-Based Precompiled Configuration Techniques for FPGA Fault Tolerance, *Proceedings of 9th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, IEEE Computer Society, 2001, s. 137-146.

[8] M. Straka, J. Kastil, and Z. Kotasek, Generic partial dynamic reconfiguration controller for fault tolerant designs based on FPGA, *NORCHIP, 2010*, 2010, s. 1–4.

[9] C. Pilotto, J. R. Azambuja, and F. L. Kastensmidt, Synchronizing triple modular redundant designs in dynamic partial reconfiguration applications, *Proceedings of the 21st annual symposium on Integrated circuits and system design*. New York, NY, USA: ACM, 2008, s. 199–204.

[10] Y. Ichinomiya, S. Tanoue, M. Amagasaki, M. Iida, M. Kuga, Improving the robustness of a softcore processor against seus by using tmr and partial reconfiguration, *Proceedings of the 2010 18th IEEE Annual International Symposium on Field-Programmable Custom Computing Machines*, IEEE Computer Society, 2010, s. 47–54.

[11] Mičulka, L., Kotásek, Z.: Design Sychronization after Partial DynamicReconfiguration of Fault Tolerant System, *15th Euromicro Conference on Digital System Design: Architectures, Methods and Tools*, Cesme-Izmir, TR, IEEE CS, 2012, s. 1-8

[12] Straka, M., Mičulka, L., Kaštil, J., Kotásek, Z.: Test Platform for Fault Tolerant Systems Design Qualities Verification, *15th IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems*, Tallin, EE, IEEE CS, 2012, s. 336-341, ISBN 978-1-4673-1185-4