APLIKACE HLÍDACÍCH OBVODŮ V ARCHITEKTURÁCH ODOLNÝCH PROTI PORUCHÁM

Martin Straka
Informační technologie, 2. ročník, prezenční studium
Školitel: Doc. Ing. Zdeněk Kotásek, CSc.

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

strakam@fit.vutbr.cz

Abstrakt. Příspěvek charakterizuje motivaci, cíle a dosavadní výsledky studenta při řešení jeho disertační práce. Je představena metodologie pro automatizované generování hlídacích obvodů pro jednoduché číslicové obvody nebo komunikační protokoly na bázi FPGA. Dále je ukázané, jak lze využít takto vygenerované hlídací obvody při návrhu architektur odolných proti poruchám. Je prezentován postup, jak vytvářet obvody se zvýšenou spolehlivostí s ohledem na dobu životnosti systému v obvodech FPGA s využitím architektur TMR a duplex doplněných o hlídací obvody. Experimentální výsledky rekapitulují náročnosti na zdroje FPGA samotných obvodů, jejich hlídačů a architektur odolných proti poruchám. V závěru je diskutován další směr výzkumu, jsou charakterizovány cíle disertační práce.

Klíčová slova. systém odolný proti poruchám, číslicový obvod, komunikační protokol, hlídací obvod, architektura, spolehlivost, TMR, duplex, FPGA, VHDL.

1 Úvod

Spolehlivost a odolnost proti poruchám (FT) jsou významnou metrikou pro návrh číslicových obvodů. Běžnou technikou pro zlepšování spolehlivosti systému je replikace funkčních jednotek a vyhodnocování paralelně zpracovávaných dat. Tento přístup však přináší vysoké nároky na zdroje číslicového obvodu, zvýšenou spotřebu energie a složitější testování obvodu. Proto je vhodné hledat nové postupy pro zabezpečení číslicových systémů a zvyšování spolehlivosti během jejich doby provozu [1].

Nové možnosti v oblasti systémů odolných proti poruchám nabízejí programovatelná hradlová pole FPGA. Funkci těchto obvodů lze měnit nahráním nové konfigurace vnitřní struktury a u vyspělých FPGA technologií je možné provést tuto změnu pouze nad částí obvodu (parciální rekonfigurace) [2]. S využitím těchto technik lze implementovat pokročilé diagnostické postupy [3]. Několik článků prezentuje mechanismy a principy pro návrh bezpečných systémů a jejich testování [4]. Techniky pro tvorbu systémů odolných proti poruchám často využívají replikaci funkčních jednotek – typicky tří modulovou redundanci [5]. V [6] popisují autoři techniku založenou na duplexním režimu využívající dvě hradlová pole FPGA. První část metodiky srovnává primární výstupy obou FPGA, druhá detekuje a označí vadný FPGA. Oba systémy využívají technik samočinné kontroly a zabezpečení s využitím parity. Pro každý systém autoři sestrojili Markovský spolehlivostní

model reflektující jednotlivé spolehlivostní parametry metody. Další články se zabývají zabezpečením propojovací sítě v FPGA a detekcí poruch funkčních částí číslicového obvodu v FPGA [7].

2 Motivace a definice problému

Cílem mého výzkumu je navrhnout kompletní metodologii pro tvorbu systémů se zvýšenou spolehlivostí využívající techniky odolnosti proti poruchám na bázi obvodů FPGA. Existuje několik přístupů, jak zvyšovat odolnost systému proti poruchám. Nejčastěji se využívá zabezpečení pomocí technik TMR a duplex nebo bezpečnostních kódů [8]. Zastoupení v této oblasti sehrávají také hlídací obvody, jež zatím nenašli v těchto technikách příliž velkého uplatnění. Jako první fáze výzkumu je ověření možnosti využití hlídacích obvodů s různou úrovní kontroly v architekturách odolných proti poruchám s ohledem na co možná nejdelší dobu životnosti systému v FPGA. Pro vytváření hlídacích obvodů by bylo vhodné navrhnout metodiku, která by umožňovala popsat důležité stavy hlídaného obvodu či systému a následné vygenerování hlídacího obvodu v jazyce VHDL. Na základě takto vygenerovaných hlídačů modifikovat různé architektury na bázi TMR a duplex a doplnit je o tyto hlídače. Mohou tedy vznikat jisté posloupnosti architektur odolných proti poruchám s různou úrovní spolehlivosti a různými nároky na zdroje FPGA. Uveďme možný případ:

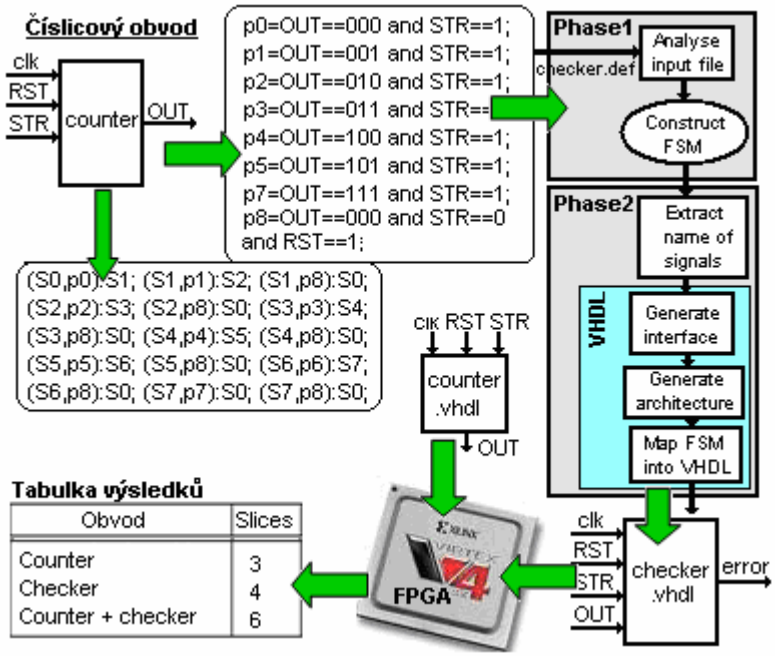
Uživatel přijde s požadavkem na vytvoření číslicového systému, který musí splňovat kritéria spolehlivosti a odolnosti proti poruchám a určí míru zabezpečení, kterou je nutno dodržet. Požaduje, aby systém, který bude implementován v obvodu FPGA měl požadovanou dobu životnosti, plnil svoji funkci i při výskytu poruchy a jeho spolehlivost neklesla pod stanovený práh. Výsledkem je vygenerování příslušné posloupnosti architektur respektující tyto požadavky.

Pojmem životnost systému rozumíme dobu, po kterou systém dokáže pracovat a produkuje správné výsledky i při výskytu poruchy. Snahou je, aby tato doba byla co nejdelší i s ohledem na omezenost prostoru v FPGA. Lze toho dosáhnou vhodnou volbou konfigurací systému, při kterém z počáteční architektury postupně ubíráme diagnostické prvky a přecházíme do jiné architektury, která zajišťuje plnou funkčnost systému, disponuje diagnostikou na nižší úrovni, neporušuje podmínky pro odolnost proti poruchám, nespadá svou spolehlivostí pod práh stanovený uživatelem a vyžaduje nižší nároky na zdroje FPGA. Je patrné, že konfigurací, kterých lze dosáhnout, může být více – viz obrázek 2 v sekci 4. Pro ohodnocení architektur v konfiguraci je nutno mít k dispozici spolehlivostní model. Tento model by zahrnoval spolehlivostní ukazatele a míru zabezpečení jak pro každou architekturu, která je součástí jakékoliv konfigurace, tak i pro celý systém. Dalším velmi důležitým problémem, kterým je třeba se zabývat, je vytipovaní nejdůležitějších funkcí systému a kvantifikace objemu kontrolovaných funkcí pro jednotlivé architektury konfigurace.

3 Metodika pro generování hlídacích obvodů

Chyby, které se mohou vyskytovat v číslicových obvodech nebo na sběrnicích komunikačních protokolů v FPGA je možné popsat různými způsoby. Typicky se používají formální modely popsané pomocí gramatik, stavových automatů nebo formálních jazyků [9]. Byla zavedena metodika, která dovoluje generovat hlídací obvody v jazyce VHDL či Verilog pro vybrané typy číslicových obvodů a komunikačních protokolů. Princip metodiky pro jednoduchý číslicový obvod je ukázán na obrázku 1. Pro popis chování obvodu byl zaveden jednoduchý definiční jazyk, kterým je možné popsat správné kombinace vstupně/výstupních (I/O) signálů a definovat správné nebo chybné stavy obvodu a umožnit tak automatické generování hlídacího obvodu. Formální popis definičního jazyka je prezentován v [10, 11]. Kombinace I/O signálů a stavy obvodu popsané definičním jazykem jsou následně analyzovány překladačem, který vytvoří popis hlídacího obvodu v jazyce VHDL. Takto získaný hlídač disponuje rozhraním hlídaného obvodu a architektura odpovídá konečnému automatu sestaveného z popisu v zavedeném definičním jazyce. Pak se hlídací obvod začlenění do systémů a provede se syntéza do FPGA. Hlavní výhodou tohoto přístupu proti přímé implementaci je možnost vygenerování

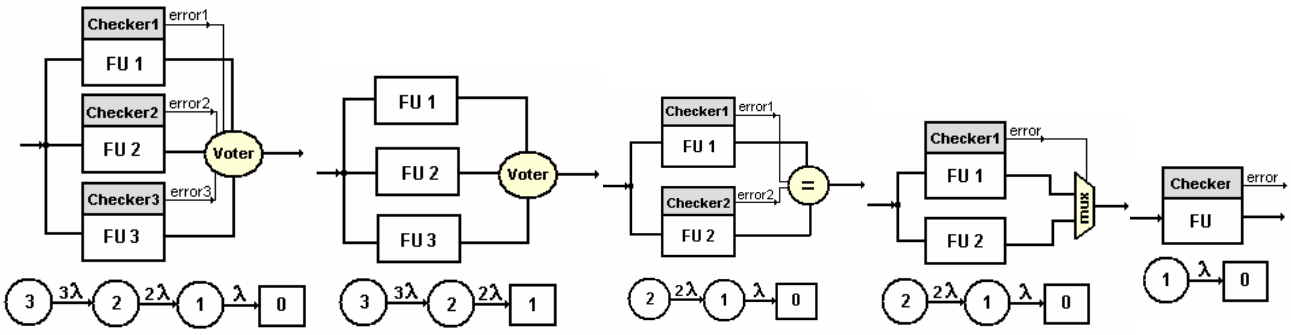
výsledného obvodu na základě jednoduchého popisu bez účasti zkušeného návrháře. V mém výzkumu představuje tato metodika první fázi plnění cílů disertační práce.



Obrázek 1: Princip metodiky pro jednoduchý číslicový obvod.

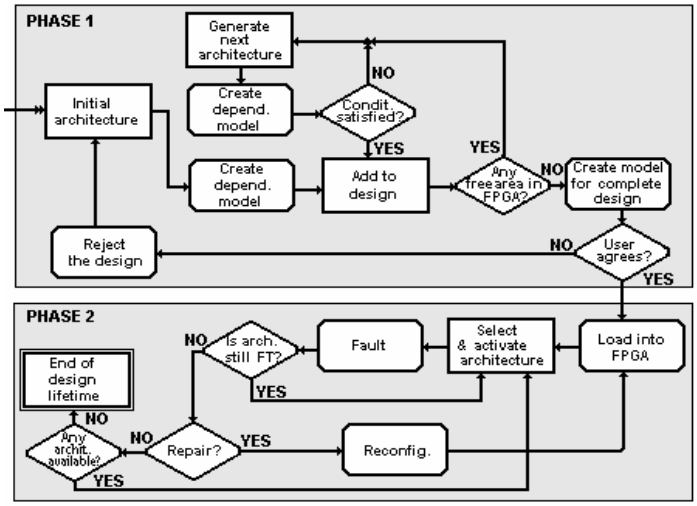
4 Aplikace hlídacích obvodů v FT architekturách

Uvažujme, že máme k dispozici FPGA, které má určitý počet CLB a systém, který je realizován jako TMR, kde každý funkční prvek (FU) má navíc zabezpečení pomocí hlídacího obvodu, který dokáže signalizovat veškeré poruchy, které mohou na hlídaném prvku nastat – viz první architektura na obrázku 2. Výstupy všech prvků jsou hlídány pomocí rozhodovacího členu (voteru). Zabezpečení rozhodovacích prvků lze realizovat pomocí dvou drátové logiky [12]. Jedná se tedy o maximální zabezpečení s ohledem na velikost obsazeného prostoru v FPGA. Pro tuto „startovací" architekturu je stanoven spolehlivostní model, jehož hodnoty budou nejvyšší ze všech možných dalších architektur. Tato architektura je odolná i vůči poruchám vzniklých na rozhodovacím členu, protože na chybu upozorní i příslušný hlídací obvod, jež nedovolí přenést chybné výsledky na výstup. Vznikne-li porucha na nějakém dalším funkčním prvku, projeví se chyba signalizací vyhodnocovacího členu a zároveň reakcí příslušného hlídacího obvodu.



Obrázek 2: Možná posloupnost architektur.

Je tedy přesně určena jednotka, která vykazuje poruchu a muže být ze systému odstraněna. V případě, že architektura už není FT, lze vyvolat reakci, kdy sytém přejde do jiné architektury, která vykazuje menší nároky na zdroje FPGA, disponuje jinou úrovní zabezpečení a zároveň nepadá pod spolehlivostní mez, kterou stanovil uživatel. Systém je tedy neustále plně funkční a vykonává nadále správnou funkci. Postupně tak systém přechází z architektur NMR na varianty TMR, dále na varianty duplex architektury až po samotný obvod s hlídačem, jež není už FT. Chyba na výstupu u architektury TMR se projeví už při poruše jedné funkční jednotky. Architektury osazené hlídači mají vlastnost fail-silent, buď vydají správný výstup nebo žádný. Každá architektura musí mít stanovený svůj spolehlivostní model, neuvažujeme-li obnovu poruchových modulů, tak je jedná o model pro neobnovitelný systém. V modelech nejsou zatím zahrnuty spolehlivosti hlídacích obvodů a rozhodovacích členů. Pro systém jako celek je pak z dílčích spolehlivostních ukazatelů stanoven celkový spolehlivostní model. Mezi důležité a sledované spolehlivostní ukazatelé patří intenzita poruch λ, střední doba bezporuchového provozu Ts a pravděpodobnost bezporuchového stavu R(t). Postup generování posloupností architektur a jejich aplikace do FPGA demonstruje obrázek 3.



Obrázek 3: Metodika generování posloupností architektur.

5 Experimenty a výsledky

Experimenty a ověření metodiky tvorby hlídačích obvodů byly provedeny na FPGA Virtex5 a Virtex2Pro firmy XILINX. Posuzovaly se náročnosti číslicových obvodů a jejích hlídačů na zdroje FPGA pro oba typy FPGA. Tabulky 1, 2 sumarizují dosažené výsledky. Hlídací obvody vyžadují větší nároky na zdroje FPGA, než samostatný obvod. To ovšem nemusí platit pro složitější funkční jednotky, kdy jeden hlídací obvod může pokrýt více komponent a tím svůj objem zmenšit. Tuto skutečnost v současné době ověřuji na vybraných obvodech testovací sady ISCAS89. Vliv na počet slices má nepochybně i architektura FPGA. Pro nové obvody Virtex5 jsou výsledky lepší, jelikož tyto typy FPGA disponují vícevstupou LUT tabulkou a mohou tedy realizovat lépe funkce více proměnných. Výsledky pro komunikační protokoly byly presentovány v [11]. Dalším ověřením bylo posouzení náročnosti představených FT architektur na zdroje FPGA. Zvolené architektury a jejich náročnost sumarizuje tabulka 3. Zde si velmi dobře vedly architektury TMR a duplex doplněný jedním hlídacím obvodem.

Virtex2Pro - XC2VP2	Circuit [slices]	Checker [slices]
Counter simple	2	14
Counter advance	2	12
Decoder	4	5
Counter+decoder	5	13
Serialiser	3	4
Shift register	4	6
Voter	8	-
Comparator	5	-

Tabulka 1: Nároky na zdroje FPGA Virtex2Pro.

Virtex5 - XCV50E	Circuit [slices]	Checker [slices]
Counter simple	2	7
Counter advance	2	4
Decoder	4	4
Counter+decoder	5	6
Serialiser	3	3
Shift register	3	4
Voter	7	-
Comparator	5	-

Tabulka 2: Nároky na zdroje FPGA Virtex5.

Virtex5 - XCV50E	TMR+3CH	TMR	Duplex+2CH	Duplex+1CH	Simple+1CH
Circuit	[slices]	[slices]	[slices]	[slices]	[slices]
Counter	21	9	15	11	6
Decoder	27	20	20	18	8
Counter+decoder	30	20	26	22	11
Serialiser	19	12	14	13	6
Shift register	24	13	18	15	7

Tabulka 3: Nároky na zdroje FPGA Virtex5 různých architektur.

6 Závěr a další výzkum

V tomto příspěvku jsou prezentovány cíle a dosavadní výsledky práce studenta na tématu. Je představena a ověřena navrhnutá metodika pro automatizované generování hlídacích obvodů a popsána aplikace těchto hlídačů v architekturách odolných proti poruchám. Další etapu plnění cílů disertační práce představuje ověření metodiky na vybraných obvodech testovací sady ISCAS89. Stěžejní částí ovšem bude rozšíření metodiky o vytipování a kvantifikovaní objemu kontrolovaných funkcí a rozvinutí navrhované metodiky pro generování posloupností architektur do širších detailů.

7 Cíle disertační práce

Cílem mého výzkumu je tedy navrhnout kompletní metodologii pro tvorbu systémů se zvýšenou spolehlivostí využívající techniky odolnosti proti poruchám na bázi obvodů FPGA. Práce by se měla opírat o možnosti využití hlídacích obvodů v různých architekturách odolných proti poruchám a tyto architektury vhodně modifikovat podle zadaných spolehlivostních parametrů. Dalším důležitým kritériem při tvorbě metodiky je zohlednění omezenosti zdrojů FPGA. Konkrétní výsledky by měly zahrnovat následující postupy a cíle:

1. Vytvořit formální prostředky pro popis vlastností, které musí kontrolovaný obvod splňovat a jejich transformace do obvodové realizace v jazyce VHDL.
2. Vytvořit obecný postup, který umožní reflektovat při návrhu kontrolního obvodu prioritní nebo vytipované funkce kontrolovaného obvodu a vytvářet hierarchicky funkční celky s různou (zadanou) úrovní kontroly správné funkce. Vytvořit postup, který umožní kvantifikovat objem kontrolovaných funkcí.
3. Pro účely popisu vlastností a hlídaných funkcí obvodu rozšířit již vytvořený definiční jazyk a metodiku pro generování hlídačů prezentovanou v kapitole 4. Porovnat obvody a jejich hlídače z hlediska objemů, které obvody představují v FPGA.
4. Pro účely implementace systému odolného proti poruchám do FPGA vytvářet metodiku pro generování sekvence dílčích architektur lišících se úrovní zabezpečení kontroly funkcí, zohlednit požadovanou dobu života systému. Výsledkem bude posloupnost architektur, každá z nich jinak diagnosticky vybavena tak, aby splňovala požadavky na spolehlivost. Tyto požadavky stanoví uživatel nebo návrhář. Pro každou architekturu a celý systém mít k dispozici spolehlivostní model. Celý postup a všechny kroky vhodně transformovat do ucelené metodiky.
5. Implementace a experimentální ověření navržených metodik.

Poděkování

Výzkum je podporován projektem financovaného Grantovou Agendou České Republiky pod číslem 102/05/H050 „Integrovaný přístup k výchově studentů DSP v oblasti paralelních a distribuovaných systémů“ a projektu č. MSM 0021630528 – „Výzkum informačních technologií z hlediska bezpečnosti“.

Literatura

[1] Galke, C., Grabow, M., Vierhaus, H. T.: Perspectives of combining on-line and off-line test technology for dependable systems on a chip. In: Proceedings of the 9th IEEE International Symposium on On-Line Testing, 2003, Paris, France, ss. 183-188

[2] Sedcole, P., Blodget, B., Becker, T., Anderson, J., Lysaght, P.: Modular dynamic reconfiguration in Virtex FPGAs. In: IEEE Proceedings Computers and Digital Techniques, 2006, IEEE Computer Society, New York, USA, ss. 157–164

[3] D'Angelo, S., Sechi, G. R., Metra, C.: Transient and Permanent Fault Diagnosis for FPGA-Based TMR Systems. In: Proceedings of the 14th international Symposium on Defect and Fault-Tolerance in VLSI Systems, 1999, IEEE Computer Society, Washington, DC, ss. 330-338

[4] D'Angelo, S., Metra, C., Pastore, S., Pogutz, A., Sechi, G. R.: Fault-Tolerant Voting Mechanism and Recovery Scheme for TMR FPGA-Based Systems. In: Proceedings of the 13th international Symposium on Defect and Fault-Tolerance in VLSI Systems, 1998, IEEE Computer Society, Cannes, France, ss. 233-240

[5] Bolchini, C., Miele, A., Santambrogio, M. D.: TMR and Partial Dynamic Reconfiguration to mitigate SEU faults in FPGAs. In: Proceedings of the 22nd IEEE international Symposium on Defect and Fault-Tolerance in VLSI, 2007, IEEE Computer Society, Rome, Italy, ss. 87-95

[6] Kubalik, P., Dobias, R., Kubatova, H.: Dependable Design for FPGA Based on Duplex System and Reconfiguration. In: Proceedings of the 9th EUROMICRO Conference on Digital System Design, 2006, IEEE Computer Society, Dubrovnik, ss. 139-145

[7] Oliveira, R., Jagirdar, A., Chakraborty, T. J.: A TMR Scheme for SEU Mitigation in Scan Flip-Flops. In: Proceedings of the 8th international Symposium on Quality Electronic Design. 2007, IEEE Computer Society, San Jose, Canada, ss. 905-910

[8] Elnozahy, E., Melhem, R., and Mossé, D.: Energy-Efficient Duplex and TMR Real-Time Systems. In: Proceedings of the 23rd IEEE Real-Time Systems Symposium, 2002, IEEE Computer Society, Austin, Texas, USA, ss. 256-262

[9] Frigerio, L., Salice, F.: RAM-based fault tolerant state machines for FPGAs. In: Proceedings of the 22nd IEEE international Symposium on Defect and Fault-Tolerance in VLSI Systems. 2007, IEEE Computer Society, Rome, Italy, ss. 312-320

[10] Straka, M., Kotasek, Z. and Winter J.: Digital systems architectures based on on-line checkers. In DSD ’08: Proceedings of the 11th EUROMICRO Conference on Digital System Design. 2008, IEEE Computer Society, Parma, Italy

[11] Straka, M., Tobola, J., and Kotasek Z.: Checker design for on-line testing of XILINX FPGA communication protocols. In DFT ’07: Proceedings of the 22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems. 2007, IEEE Computer Society, Rome, Italy, ss. 152–160

[12] Ito, H.: A 2-rail logic combinational circuit with easy detection ofstuck-open and stuck-on faults in FETs. In: International Symposium on Fault-Tolerant Systems, 1997, IEEE Computer Society, Kawasaki, Japan ss. 252-257