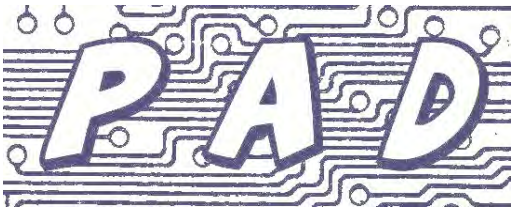
TECHNICKÁ UNIVERZITA V LIBERCI
www.tul.cz



Sborník příspěvků PAD 2014

Počítačové architektury & diagnostika

Česko-slovenský seminář pro studenty doktorského studia

Elektronická verze

TU v Liberci

FMIMS – ústav ITE

Malá Skála, 4. – 6. 9.2014

TECHNICKÁ UNIVERZITA V LIBERCI
Fakulta mechatroniky, informatiky
a mezioborových studií

Počítačové architektury a diagnostika 2014

Česko-slovenský seminář pro studenty doktorského studia

Malá Skála 4. – 6. 9. 2014

Elektronický sborník příspěvků

Technická univerzita v Liberci
Fakulta mechatroniky a mezioborových inženýrských studií
Ústav informačních technologií a elektroniky

Editor publikace: prof. Ing. Zdeněk Plíva, Ph.D., Ing. Martin Rozkovec, Ph.D.

ISBN 978-80-7494-027-9

Pár slov na úvod...

Je září, dozrává ovoce, vítr se prohání po strništích, dým z bramborových natí vhání slzy do očí a tažní ptáci se chystají na dalekou cestu. A studenti doktorského studia se zaměřením na počítačové architektury a jejich diagnostiku se sjíždějí na pravidelné setkání, na PAD. Ano, tímto textem začínal úvodník sborníku PADu 2008 a již je to tu opět. Jaká je celá historie těchto setkání? Tedy chronologicky navazujeme na setkání ve Zvíkovském Podhradí (2003), v Moravanech nad Váhom (2004), Lázních Sedmihorky (2005), Papradnu v Javorníkách (2006) v Srní (2007), v Hejnicích (2008), v Soláni (2009), v Češkovicích (2010), ve Staré Lesné (2011), v Milovech ve Žďárských vrších (2012) a konečně v Teplé (2013). Údolí Jizery, hrady Vranov, Frydštejn, Suché skály, či vrch Sokol budou bdít nad pokračováním tohoto setkání studentů se školiteli v Malé Skále, hotelu Kavka; PAD 2014 pořádá Ústav informačních technologií a elektroniky na fakultě mechatroniky, informatiky a mezioborových studií Technické university v Liberci. Seminář PAD je pořádán s cílem umožnit studentům konfrontovat výsledky práce mimo rámec mateřské univerzity, vyzkoušet si prezentaci před kolegy i před pedagogy ze všech koutů bývalého Československa. Studenti tak mohou získat nejen zkušenosti s vystupováním před odbornou veřejností ale i názor na své výsledky od širšího okruhu posluchačů. Není totiž běžné, aby se nějaké akce zúčastnilo více „školitelů“ než přednášejících studentů; je to způsobeno i zájmem externích firem, který upřímně vítáme.

Na tomto místě je zvykem pogratulovat studentům, jejichž práce byla na loňském PADu oceněna cenou Prof. Jana Hlavičky. Byli to tito studenti:

1. ročník – Ing. Gabriel Nagy, FEI STU v Bratislavě
2. ročník – Ing. Jiří Matoušek, FIT VUT v Brně
3. ročník – cena nebyla udělena

Je také mou milou povinností poděkovat sponzorům akce. V abecedním pořádku to jsou firmy ASICentrum spol. s r.o., Jablotron Alarms a.s. a Presiosa a.s., jejichž inzeráty jsou uvedeny v tomto sborníku; mediálním partnerem je DPS – Elektronika od A do Z.

V Liberci 11. srpna 2014

Za celý organizační výbor PAD 2014
Zdeněk Plíva

Programový výbor PAD 2014

Michal Bidlo, VUT v Brně
Roland Dobai, VUT v Brně
Jan Dohnal, ON-semi
Vladimír Drábek, VUT v Brně
Karel Dudáček, ZČU v Plzni
Petr Fišer, ČVUT v Praze
Elena Gramatová, STU v Bratislavě
Jiří Jaroš, VUT v Brně
Katarína Jelemenská, STU v Bratislavě
Jiří Jeníček, TU v Liberci
Jan Kořenek, VUT v Brně
Zdeněk Kotásek, VUT v Brně
Tomáš Koutný, ZČU v Plzni
Hana Kubátová, ČVUT v Praze
Róbert Lórencz, ČVUT v Praze
Ondřej Novák, TU v Liberci
Antonín Pleštil, ASICentrum
Zdeněk Plíva, TU v Liberci
Stanislav Racek, ZČU v Plzni
Martin Rozkovec, TU v Liberci
Richard Růžička, VUT v Brně
Jan Schmidt, ČVUT v Praze
Miroslav Skrbek, ČVUT v Praze
Vladimír Smotlacha, ČVUT v Praze
Viera Stopjaková, ČVUT v Praze
Josef Strnadel, STU v Bratislavě
Vlastimil Vavřička, ZČU v Plzni
Karel Vlček, UTB ve Zlíně
Tomáš Zahradnický, ČVUT v Praze

Organizační výbor PAD 2014

Tomáš Drahoňovský
Jiří Jeníček
Ondřej Novák
Zdeněk Plíva
Petr Pfeifer
Martin Rozkovec

Obsah sborníku:

VÝUKOVÁ LABORATOŘ IP TELEFONIE

Tomáš Knot
Inženýrská informatika, 1. ročník, prezenční studium
Školitel: Karel Vlček

Fakulta aplikované informatiky, Univerzita Tomáše Bati ve Zlíně
Nad Stráněmi 4511, 760 05 Zlín

knot@fai.utb.cz

Abstrakt. Tato práce se zabývá popisem tvorby laboratoře a pobočkové ústředny pro výuku IP telefonie na Univerzitě Tomáše Bati ve Zlíně. Laboratoř se skládá z pobočkové ústředny, která je založena na GNU/Linux Debian a softwaru Asterisk a je využívána v předmětu Telekomunikační systémy. Ústředna je připojena do veřejné telefonní sítě přes poskytovatele hlasových služeb. Studenti mají možnost tvorby vlastních ústředen na svých stanicích (využívá se virtualizace GNU/Linuxu a Asterisku), které mohou mezi sebou propojovat včetně výukové ústředny, která má nejvyšší úroveň zapojení.

Klíčová slova. IP telefonie, Asterisk, VoIP (Voice over IP), PBX (Private Branch Exchange), Linux.

1 Úvod

S rozvojem telekomunikací a Internetu dostávají hlasové služby nový rozměr. V současné době je možné sledovat odklon od klasické telefonie k IP (Internet Protocol) telefonii, jenž má také označení jako VoIP (Voice over IP). Společně s IP telefonní se také rozvíjejí sítě NGN (Next Generation Network), které ukazují nový pohled na telekomunikační sítě. NGN je vysokorychlostní digitální síť, jejíž úkolem je integrovat technologii přepojování okruhů a paketový přenos do jedné služby. Tímto způsobem dochází ke snížení nákladů na nákup zařízení, které jsou nutné pro konverzi mezi jednotlivými protokoly a rozhraními s využitím směrovačů IP.

V sítích, u nichž probíhá datové a hlasové služby, je vhodné použít řešení kvality služeb QoS (Quality of Service). Toto řešení umožňuje provádět upřednostňování daných služeb před ostatními. Každá služba má přiřazenu prioritu. Je-li využívána služba IP telefonie, pak by měla mít nastavenu nejvyšší prioritu. Tím pádem je upřednostňována před veškerým dalším provozem na sítí. Je to dáno z důvodu zvýšení kvality hlasových služeb.

1.1 Výhody a nevýhody IP telefonie

Jednou z významných výhod IP telefonie je sdílení síťové infrastruktury, protože není nutné budovat novou infrastrukturu. Na datové sítí se provozují společně hlasové a datové služby. Toto řešení snižuje náklady na další rozvoj telefonie v rámci firem, domácností a má za následek snadnější a rychlejší implementaci IP telefonie. Vybudování IP ústředny založené na VoIP je díky nízkým nákladů na provoz, rozvoj a propojovací poplatky vhodnou alternativou ke klasické telefonii.

Použití IP telefonie přináší také nevýhody, mezi které patří nižší spolehlivost a dostupnost oproti klasické telefonii o zhruba 0,5 %. S rozvojem voláním přes Internet se začínají objevovat nové hrozby. Jedná se například o zneužití VoIP systémů, podvržení cizí identity, spam, zajištění utajení identity

(hovorů i signalizace) a integrity hovoru. Přestože IP telefonie má své nevýhody, převažují výhody pro její zavádění do praxe.

1.2 Důvod vytvoření výukové laboratoře

Před vybudováním laboratoře se na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně nenacházela žádná laboratoř, která by se věnovala IP telefonii a hlasovým službám.

Dalším důvodem vytvoření laboratoře bylo zařazení výuky IP telefonie do předmětu Telekomunikační systémy. Společně s budováním laboratoře se vytvořily výukové materiály pro studenty, které obsahují základní seznámení s tvorbou, nastavením a zabezpečením softwarové pobočkové ústředny Asterisk včetně správy a nastavením operačního systému GNU/Linux Debian.

1.3 Analýza navrhovaného řešení

Zvolená varianta vychází z počtu nasazení obdobného řešení u nás a ve světě. Softwarová pobočková ústředna Asterisk je v kombinaci s webovým rozhraním FreePBX velmi populární díky svému širokému technickému řešení, které se vyrovná placeným pobočkovým ústřednám (nízké naklady na budování, její následný provoz a rozvoj).

Při řešení projektu bylo kladeno za cíl, aby výsledné řešení odpovídalo reálnému nasazení v praxi. Zvolený software je pod licencí GNU GPL (General Public Licence) a může být nasazen v libovolném množství instalací bez nutnosti platit licenční poplatky za SW. Dále licence také nabízí možnost modifikace zdrojových kódu, které je vhodné pro další zkoumání funkčnosti a možnost tvorby nových funkcí ústředny.

Realizaovaná laboratoř umožňuje propojení IP telefonie s klasickou digitální telefonií ISDN (Integrated Services Digital Network). Toto rešení je velmi flexibilní a dokáže reagovat na jakýkoliv nový prvek či případnou změnu topologie. Proto je v budoucnu plánováno propojení IP a digitální telefonie v rámci laboratoře, za cílem poskytnout obě možnosti spojení do veřejné telefonní sítě.

2 Návrh laboratoře

Laboratoř je sestavena ze serveru, který představuje softwarovou pobočkovou ústřednu. Její chod zajišťuje operační systém (OS) GNU/Linux Debian Wheezy. OS je zabezpečen a nastaven jako běžný linuxový server. Dále je nainstalován software (SW) Asterisk ve 11.10.2 a webové rozhraní FreePBX ve verzi 2.11. Tento SW je srdcem celé ústředny a má na starost samotné směrování IP hovorů. Ústředna je zabezpečena a nastavena pro provoz ve veřejné telefonní síti.

2.1 Asterisk

Asterisk je softwarová pobočková ústředna pracující s IP telefonií, digitální ISDN, analogovou telefonní a má označení jako source hybrid TDM (Time Division Multiplex) a také jako packet voice PBX (Private Branch Exchange). Jejím úkolem je spojovat a směrovat jednotlivé hovory na příslušné telefony, případně předávat hovory pomoci trunkového spojení na další ústředny, které mohou být propojeny do veřejné telefonní sítě.

Součástí ústředny je i IVR (Interactive Voice Responce). Jedná se o automatický odpovídač ovládaný přes DTFM (Dual-Tone Multi-Frequency) nebo pomocí hlasu. IVR obsahuje také hlasovou schránku pro zanechání vzkazu volajícímu nebo systém pro obsluhu zákaznického účtu (oznámení o zůstatku na účtu, informace o účtu apod.). Další funkcionalitou je ACD (Automatic Call Distribution), jehož úkolem je rozdělování hovorů podle zadaných kritérií (podle určených schémat, podle čísla volajícího, časových podmínek apod.).

SW Asterisk je vyvíjen pod GNU GPL licencí. Jeho výhodou je to, že nabízené funkce se vyrovnají komerčním PBX systémům.

Účel	Počet kanálů	Minimální konfugurace
Domácí systém	Ne více než 5	400 MHz x86, 256 MB RAM
SOHO (Small Office Home Office) systém	5 – 10	1 GHZ x86, 512 MB RAM
Malý podnikový systém	Až 25	3 GHz x86, 1 GB RAM
Střední až velký podnikový systém	Více než 25	dual-core CPU

Tabulka 1: HW konfigurace Asterisku.

2.1.1 Rozhraní a protokoly Asterisku

Asterisk je možné použít v následujících aplikacích:

- PBX s rozhraním do PSTN (Public Switched Telephone Network)
- Gateway pro VoIP - protokoly MGCP (Media Gateway Control Protocol), SIP (Session Initiation Protocol), IAX (Inter-Asterisk eXchange), H.323
- Softwarová ústředna (softswitch) – softwarové řešení komunikačního serveru
- Šifrování telefonních a faxových spojení
- Překlad čísel
- Konferenční server – je obsaženo konferenční místo, funkce Meet me

Existuje také podpora pro řadu rozhraní, protokolů a kodeků. Podporovaná rozhraní jsou:

- DAHDI – Jedná se o HW pro zpracování TDM, které nabízí různá síťová rozhraní – PSTN, POTS (Plain Old Telephone Service), PRI (Primary Rate Interface), BRI (Basic Rate Interface) a další.
- Non-Zaptel HW – Je to rozhraní zajišťující připojení k tradičním telefonním službám jako je ISDN4Linux, OSS/Alsa, Linux Telephony Interface atd. Není podporováno pseudo-DTM.
- Packet voice – Nejedná se o HW, ale o skupinu protokolů, které zajišťují komunikaci přes IP síť. Do této skupiny patří protokoly SIP, MGCP, IAX/IAX2, H.323, VoFR (Voice over Frame Relay).

V Asterisku se kanálem rozumí logické spojení různých signalizačních a přenosových cest k vytvoření a následnému spojení telefonních hovorů. Přes kanál mohou do systému vstupovat různé druhy komunikace. Na vstupu může být fyzický telefonní okruh (PRI, BRI apod.), softwarová spojení, síťová spojení (SIP, AIX) nebo vnitřní kanály. Asterisk přistupuje ke všem spojením rovnocenně, i když se jedná o různé technologie, protože každý kanál je interpretován jako přípojný bod. Z toho důvodu je Asterisk velmi flexibilní řešení.

Využívá se H.323 jako gateway, dále IAX2 protokol pro přenos signalizace a hlasu. Tímto způsobem je zajištěno propojení Asterisk serverů a klientů. Protokol SIP má na starost komunikaci mezi ústřednou a telefony. Asterisk může vystupovat jako SIP server, SIP klient případně jako SIP gateway pro SIP, IAX, MGCP, H.323 a PSTN.

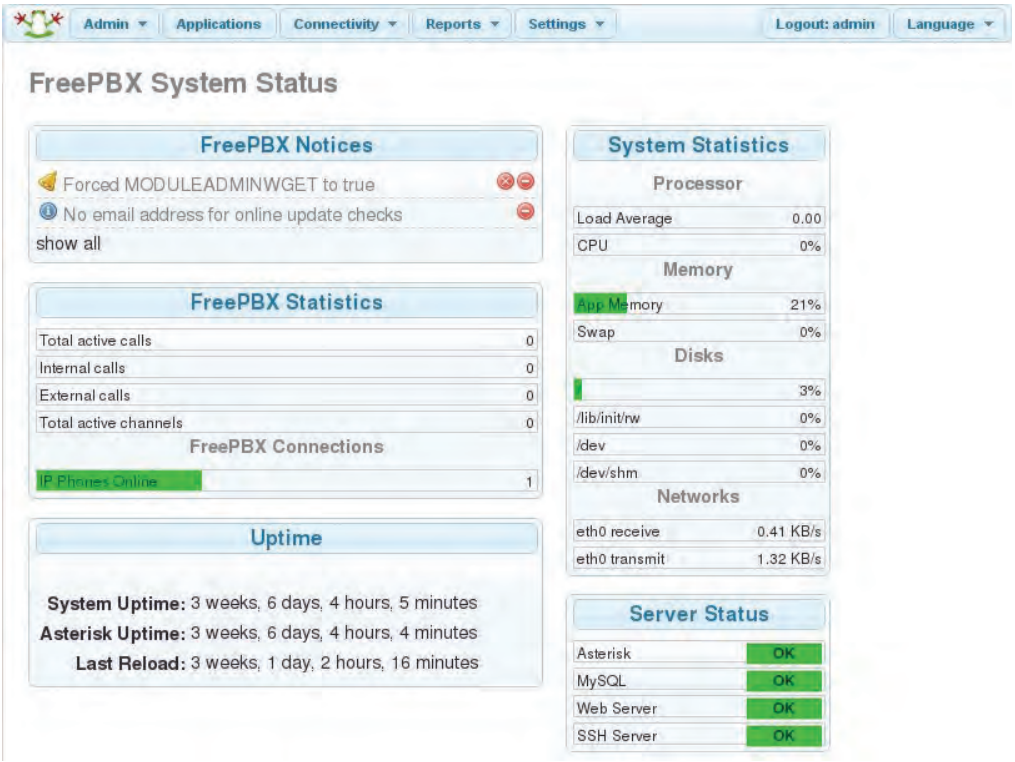
Kodeky v Asterisku jsou paketizovány na 20 ms u protokolů RTP (Real-time Transport Protocol). Podporované protokoly jsou ADPCM (Adaptive Differential Pulse Code Modulation), G.711 μlaw, G.711 alaw, G.723.1, G.729, GSM, iLBC (Internet Low Bitrate Code), LPC10, Speex.

2.2 FreePBX

FreePBX (Obrázek 1) je webové grafické rozhraní, které je nadřazeno nad celou PBX Asterisk a umožňuje zjednodušenou správu celé ústředny. Veškerá nastavení jsou prováděna přes toto rozhraní a správce již nemusí přistupovat ke konfiguračním souborům Asterisku skrze příkazovou řádku. Pro svůj běh FreePBX vyžaduje webový server Apache, databázový server MySQL a samozřejmě Asterisk.

FreePBX nabízí vysokou modularitu skrze repozitáře, které nabízejí nové rozšiřující moduly a přináší nové vlastnosti pro celou ústřednu. Po instalaci nabízí FreePBX následující funkce:

- Neomezený počet hlasových schránek
- Music on Hold – Přehrávání MP3 nebo stream z Internetu, kdy volající čeká na přijmutí hovoru
- Neomezený počet konferencí – Omezení je dáno výkonem procesoru.
- Fronta hovorů a další



Obrázek 1: Ukázka webové aplikace FreePBX.

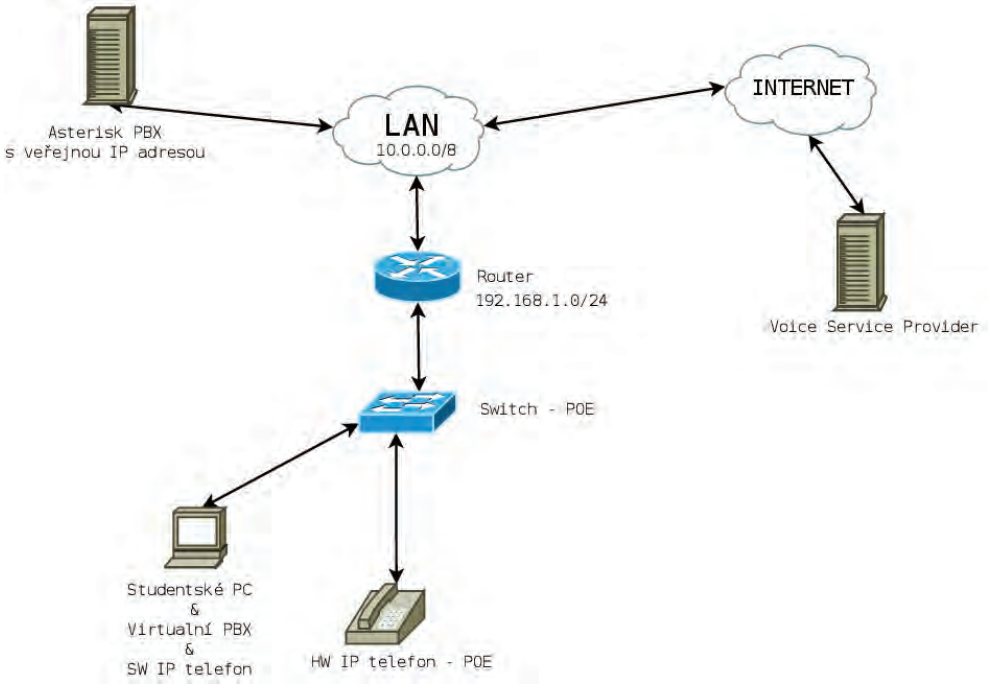
2.3 Topologie laboratoře

Topologie laboratoře se skládá ze serveru s nainstalovaným OS GNU/Linux Debian Wheezy s KVM (Kernel-based Virtual Machine) virtualizací. Tento způsob instalace je zvolen z důvodu provozu více různých verzí Asterisku nezávisle na sobě, kdy je možné testovat nové funkce, nastavení a případné nové změny lze přenášet do hlavní výukové ústředny. Server je zabezpečen proti případným útokům pomocí iptables.

Hlavní výuková ústředna Asterisk byla zkompilována ze zdrojových kódů a následně nainstalována. V následujícím kroku proběhla instalace grafického rozhraní FreePBX a dalších aplikací nutných pro její chod. Zabezpečení ústředny je provedeno tak, aby bylo umožněno připojení dalších ústředen. Ústředna má připojení do veřejné telefonní sítě přes trunk k poskytovali hlasových

služeb. Studenti mají k dispozici několik veřejných telefonních čísel, které využívají pro ověření svého nastavení virtuální ústředny a telefonu.

Obrázek 2 popisuje zapojení síťové infrastruktury. Každý ze studentů vytváří vlastní virtuální pobočkovou ústřednu, se kterou se připojuje k hlavní výukové ústředně. Studentské PC obsahuje program pro virtualizaci Virtualbox a aplikaci pro SW telefon Linphone s headsetem. K dispozici jsou také HW telefony, které jsou připojeny do samostatné VVLAN (Voice VLAN), aby byly odděleny od běžného provozu sítě. Na celou síť je aplikováno QoS. Laboratoř je vytvořena tak, aby svým zapojením připomínala malou firmu, která využívá IP telefonii.



Obrázek 2: Topologie zapojení výukové laboratoře.

3 Výukové materiály

Společně s budováním laboratoře vznikly také výukové materiály, které studentům slouží k seznámení se s principem a provozem IP telefonie. Materiály mají podobu laboratorních úloh, v nichž studenti řeší konkretní situace s provozem ústředny.

První část materiálů je zaměřena na instalaci OS GNU/Linux Debian, PBX Asterisk a FreePBX. V úloze jsou popsány postupy, jak korektně zprovoznit ústřednu jako celek.

Druhá část materiálů se věnuje zabezpečení. Obsahem je zabezpečení proti neoprávněnému přístupu. Student je seznámen s nástrojem iptables, ve kterém se vytváří hlavní firewallový skript pro zabezpečení OS. Dále je popsáno zabezpečení vzdáleného přihlášení SSH (Secure Shell) a jeho zabezpečení, který slouží pro přístup k příkazovému řádku OS a konzoli Asterisku. Jako další bezpečnostní opatření je studentům ukázána změna výchozích hesel, jenž jsou nastaveny v Asterisku a FreePBX.

Další část laboratorních úloh se zaměřuje na nastavení samotné ústředny. Studenti získávají přehled o způsobu propojování vlastních ústředen za pomocí trunku k nadřazené ústředně u

poskytovatele hlasových služeb. Všechny vytvořené studentské ústředny lze propojovat mezi sebou skrze trunk (nastavení příchozích a odchozích cest pro směrování hovorů na ústřednu a mimo ni).

Poslední část se věnuje troubleshootingu, v níž studenti řeší ukázkové chyby, jenž vznikají při realizaci ústředny. Příkladem může být propojení dvou ústředen mezi sebou a tvorby trunku včetně směrování hovorů mezi nimi. Řešeny jsou také chyby související s registrací telefonů na ústřednu. Ukázány jsou příklady odchytávání chyb v konzoli Asterisku přes vestavěný debugger.

4 Závěr

Tento příspěvek si kladl za cíl prezentovat nově vzniklou laboratoř na Fakultě aplikované informatiky Univerzity Tomáše Bati ve Zlíně. Představeno bylo řešení topologie, HW a SW vybavení.

Inovace tohoto řešení spočívá v tom, že je provozována ve virtuálním prostředí. Proto případné změny, tedy nové funkce a nastavení, je možné zkoušet nezávisle na hlavním produkčním řešení. Při návrhu bylo snahou přinést studentům možnost tvorby vlastní ústředny, kterou je možné propojit do veřejné telefonní sítě. Díky návrhu studenti získávají praktický náhled do tvorby vlastní ústředny.

Celá laboratoř se postavena na open source SW a při realizaci není nutné kupovat proprietární SW. Veškeré náklady na budování laboratoře proto mohou směřovat pouze do nákupu HW. V budoucnu bude snaha laboratoř doplnit o další zařízení, jenž souvisí s IP telefonií a jejím propojením do ISDN.

Poděkování

Tento článek byl podpořen projektem IGA/FAI/2014/008.

Reference

[1] MEGGELEN, Jim Van, Leif MADSEN a Jared SMITH. Asterisk: the future of telephony. 2nd ed. Beijing: O´Reilly, 2007, 574 s. ISBN 05-965-1048-9.

[2] KNOT, Tomáš. Softwarová pobočková ústředna Asterisk [online]. 2013 [cit. 2014-06-18]. Diplomová práce. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky. Vedoucí práce Karel Vlček. Dostupné z: <http://theses.cz/id/7z6cx3/>.

[3] VOZŇÁK, Miroslav. Voice over IP. 1. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2008, 176 s. ISBN 978-80-248-1828-3.

[4] VOZŇÁK, Miroslav. Telefonní ústředny Asterisk. In: Teorie a praxe IP telefonie [online]. 2008 [cit. 2014-06-19]. Dostupné z: <http://www.ip-telefon.cz/archiv/dok_osta/ipt-2008_Telefonni_ustredny_Asterisk.pdf>

[5] WIJA, Tomáš, David ZUKAL a Miroslav VOZŇÁK. Asterisk a jeho použití [on-line]. 2005 [cit. 2014-06-19]. Dostupné z: <http://archiv.cesnet.cz/doc/techzpravy/2005/voip/asterisk.pdf>

[6] FreePBX - voip-info.org. Voip-info.org [online]. 2012, Fri 23 of Mar, 2012 (18:31) [cit. 2014-06-20]. Dostupné z: <http://www.voip-info.org/wiki/view/freePBX>

[7] Welcome | FreePBX. FreePBX [online]. 2013 [cit. 2014-06-20]. Dostupné z: <http://www.freepbx.org/>

[8] Installation | FreePBX. FreePBX [online]. 2013 [cit. 2014-06-20]. Dostupné z: <http://www.freepbx.org/support/freepbx-terms/documentation/installation-0>

Testing Fault-Tolerance Properties in FPGA based Systems Controlling Electro-mechanical Applications

Jakub Podivínský
Computer Science and Engineering, 1st class, full-time study
Supervisor: Zdeněk Kotásek

Faculty of Information Technology, Brno University of Technology
Božetěchova 2, Brno 612 66

ipodivinsky@fit.vutbr.cz

Abstract. The aim of this paper is to present a new platform for estimating the fault-tolerance quality of electro-mechanical applications based on FPGAs. We demonstrate one working example of such EM application that was evaluated using our platform: the mechanical robot and its electronic controller in an FPGA. In the experiments, the mechanical robot is simulated in the simulation environment, where the effects of faults injected into its controller can be seen. In this way, it is possible to differentiate between the fault that causes the failure of the system and the fault that only decreases the performance.

Keywords. Fault Tolerance, Electro-mechanical Systems, Fault Injection, SEU.

1 Introduction

In several areas, such as aerospace and space applications or automotive safety-critical applications, fault tolerant electro-mechanical (EM) systems are highly desirable. In these systems, the mechanical part is controlled by its electronic controller. Currently, a trend is to add even more electronics into EM systems. For example, in aerospace, extending of the electronic part results in a lower weight that helps reduce the operating cost [1]. It is obvious that the fault-tolerance methodologies are targeted mainly to the electronic components because they perform the actual computation. However, as the electronics can be realized on different hardware platforms (processors, ASICs, FPGAs, etc.), specific fault-tolerance techniques dedicated for these platforms must be developed.

Our research is targeted to *Field Programmable Gate Array*s (FPGAs) as they present many advantages from the industrial point of view. They can compute many problems hundreds times faster than modern processors. Moreover, their reconfigurability allows almost the same flexibility as processors. FPGAs are composed of *Configurable Logic Blocks* (CLBs) that are interconnected by a programmable interconnection net. Every CLB consists of *Look-Up Tables*(LUTs) that realize the logic function, a multiplexer and a flip-flop. The configuration of CLBs and of the interconnection net is stored in the SRAM memory. The problem from the reliability point of view is that FPGAs are quite sensitive to faults caused by charged particles [2]. These particles can induce an inversion of a bit in the configuration SRAM memory of an FPGA and this may leads to a change in its behaviour. This event is called the *Single Event Upset* (SEU). Sensitivity to faults (SEUs) and the possibility of reconfiguration are the main reasons why so many fault-tolerance methodologies inclined to FPGAs have been developed and new ones are under investigation [3].

The paper is organized as follows. The goals of our research and the platform for estimating the quality of EM applications can be found in Section 2. The architecture of our experimental robot controller is

provided in Section 3. A description of the fault injection process are described in Section 4. Results of the experiments with the robot controller are available in Section 5. The future work that includes using *functional verification* for automated evaluation of impacts of faults is presented in Section 6. Finally, Section 8 concludes the paper.

2 The Goals of the Research

From the above facts, we have identified two areas that we would like to focus on in our research of fault-tolerant FPGA-based systems controlling electro-mechanical applications.

The first one is that methodologies are validated and demonstrated only on simple electronic circuits implemented in FPGAs. For instance, methodologies focused on the memory in [4] are validated on simple memories without the additional logic around. In [5], the fault-tolerance technique is presented only on a two-input multiplexer, one simple adder and one counter. However, in real systems different types of blocks must be protected against faults at the same time and must communicate with each other. Therefore, a general evaluation platform for testing, analysis and comparison of alone-working or cooperating fault-tolerance methodologies is needed.

As for the second area of the research and the main contribution of our work, we feel that it must be possible to check the reactions of the mechanical part of the system if the functionality of its electronic controller is corrupted by faults. It is either done in simulation or in a physical realization. In our opinion, it is important to find a relation between the level of functional corruption of the electronic controller and the corruption of the mechanical functionality in the EM applications (i.e. between the robot controller and the simulated mechanical robot).

According to the identified problems we have formulated our goal in the following way:

> *To develop an evaluation platform based on the FPGA technology for checking the resilience of EM applications against faults.*

Under the term EM application we understand a mechanical device and its electronic controller implemented in an FPGA. In our experiments, these components are represented by a robot device and its controller, which drives the movement of a robot in a maze. At this point, we wanted to target also the issue of complexity. We have implemented the evaluation platform that consists of three basic parts:

- the Virtex5 FPGA board into which the robot controller is configured,
- the simulation environment for simulating robot and its environment,
- the external fault injector (PC) which inserts faults into the robot controller [6].

3 The Robot Controller - Structure and Principles

In Figure 1, the block diagram of the implemented robot controller is available. The control unit is connected to the PC (where the simulation environment is located) via the Interface Block. Through this block, data from the simulation are received and in the opposite direction, instructions about the movement of the robot are sent back.

The robot controller is composed of various blocks, their function is described in [7]. Here, we only summarize the main characteristics of every component. The central block of the robot controller is a bus through which the communication between each block is accomplished. The Position Evaluation Unit (PEU) calculates position of the robot in the maze and provided them to other units as coordinates x and y. The Barrier Detection Unit (BDU) uses four sensors and provides information about the distance to the surrounding barriers as four-bit vector. Map updating provided by the Map Unit (MU) is based on

the information about the position of the robot and the four-bit barriers vector. The Map Memory Unit (MMU) stores the information about the up-to-date map. Path Finding Unit (PFU) implements simple iteration algorithm for finding a path through the maze according to the information about the current and the desired target position. The mechanical parts of the robot are driven by the setting of the speed in the required direction of the movement by the Engine Control Module (ECM).

The robot controller is designed as a complex system with specific components that will allow testing and validating various types of individual or cooperating fault-tolerance methodologies focused on FPGAs. There are combinational circuits, sequential circuits, finite state machines, memories or buses.

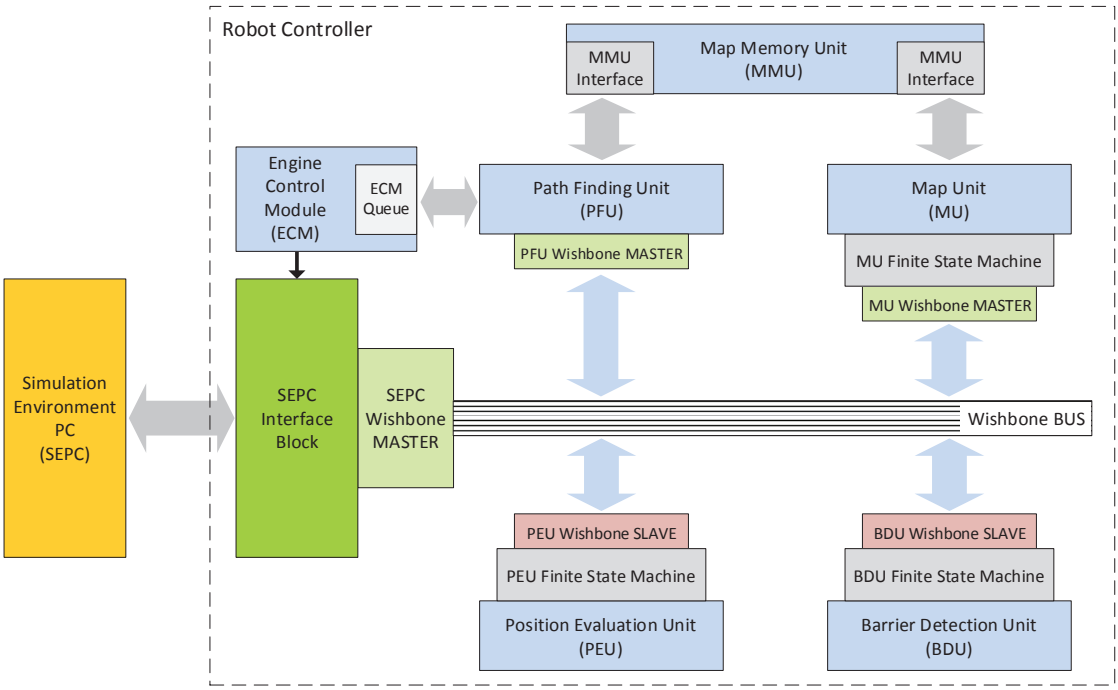


Figure 1: The block diagram of the robot controller.

4 Evaluation of Reliability by Fault Injection

During testing the resilience of systems against faults, waiting for their natural appearance is not feasible. A typical reason is the *Mean Time Between Failures* (MTBF) parameter that can be in the order of years. The most popular techniques to artificially accelerate fault occurrence is called *fault injection*.

Therefore, to simulate the effects of faults in the FPGA, it could be done by a direct change of the configuration bitstream which is loaded into the configuration memory. For this purpose, a fault injector [6] was implemented which allows to modify single or multiple specified bits of the bitstream in order to simulate single and multiple faults.

For effective testing of fault effects on a system composed of several blocks, we need to determine the block in which the fault will be injected. In the case of injecting faults into the whole FPGA we are not sure which block is affected, or if the useful part of the bitstream is hit. The list of bits representing each component can be obtained through several steps by using the PlanAhead [8] tool for the layout of the components on the FPGA. The knowledge about component layout allows us to use the RapidSmith [9] tool for analysing the design. This tool is able to generate a list of the bitstream bits that correspond to the identified areas of the FPGA, while we know what components are configured into particular area. The disadvantage of such approach is that this process provides only a list of bitstream bits that correspond to *Lookup Tables* (LUTs).

5 The Experiment with the Robot Controller

The aim of the experiment is to identify which parts of the robot controller are vulnerable to faults. The flow of the experiment is displayed in Figure 2. At first, we initiate the environment of the robot in simulation. As the first scenario, we chose a small maze with 8x8 fields. Subsequently, the robot controller is initiated. Then the robot starts to search a path to the end position. At this point, the fault injection takes place. We generate randomly an LUT of every unit of the robot controller into which the fault will be injected. Thanks to the Rapidsmith, just the corresponding bits of the bistream are inverted. Faults are injected one after another until the robot starts to behave incorrectly or has an accident. We were monitoring (1) the number of faults that led to the malfunction of the robot and (2) how the behaviour of the robot was changed.

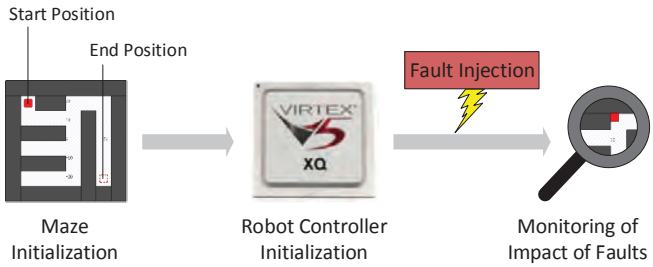


Figure 2: The flow of one experiment.

The results of the experiments are shown in Table 1. In the first column, the list of components of the robot controller is provided. In the second column, the total number of bits of the bitstream that belong to the LUTs of corresponding components is shown. The following three columns represent the number of injected faults into particular components which caused incorrect behaviour of the robot. Injecting faults into all bits of the bitstream would be very time-consuming, because evaluation of faults impact on robot behaviour was monitored manually. Therefore, we utilise the statistic evaluation. 20 experimental runs were performed for each component (320 experimental runs in total). The last column of the table contains the state of the robot that was evaluated as the wrong behaviour.

One interesting conclusion arises from the results. The incorrect behaviour did not appear immediately after the first injection of a fault. We can conclude that some bits of the bitstream, despite they are identified as related to the robot controller, are not used to store a useful information. This can be seen particularly in components PEU_FSM and PEU_WB. Nevertheless, we realised that some components contain more critical bits than others and thus they should be preferred while hardening against faults by some fault-tolerance methods.

The most common consequences of injected faults which are presented in table are Freezing on place, Deadlock, Crashing into a wall and something other. As can be seen from the table, the most common consequence of injected faults is *Freezing on place*. We can also conclude that stopping of the robot is not so critical as for example a collision with the wall. This conclusion can be very critical and useful for different kinds of EM applications.

6 Functional Verification for Automated Evaluation of Fault Impacts

For extensive testing of the behaviour of the robot or any other EM system placed into our evaluation platform, we need to examine various test scenarios. After application of proper test vectors, we can prove the correctness and accuracy of the behaviour of the system with respect to the specification. The manual check of these test vectors is difficult as it requires a full control from the user. The user is responsible for running the test environment, generating test vectors and also analysing the outputs of

Table 1: The experimental results.

Components	Bits of bitstream	Number of injected faults			Consequence
		Min	Median	Max	
PEU	21 632	2	6	12	freezing
PEU_FSM	2 112	>80	-	>80	-
PEU_WB	2 112	41	-	>80	freezing
BDU	320	2	6	21	freezing
BDU_FSM	2 752	3	6	34	freezing
BDU_WB	2 176	3	9	28	freezing
SEPC_INF	1 216	2	3	7	freezing
SEPC_WB	9 088	2	3	7	freezing
ECM	25 664	1	2	7	freezing
PFU	7 488	3	6	12	deadlock
PFU_WB	7 424	2	3	9	freezing
MU	11 840	1	2	3	crashing
MU_FSM	1 280	1	3	5	freezing
MU_WB	7 680	1	3	6	freezing
MMU	3 008	1	3	6	freezing
WB_BUS	5 056	1	3	6	freezing

the system. All these activities are time-demanding and therefore, it is not possible to test the system thoroughly within a reasonable time. It is necessary to apply some kind of automation. An extended technique for automated checking of the correctness of the system is called verification. There are several techniques used in the verification domain. We decided to use an approach called functional verification, as this type of verification fits best to our future experiments.

To be able to inject faults into the FPGA while performing functional verification, we must carry out verification directly in the FPGA (not in the simulation as usually). Advantageously we can use and modify hardware accelerated verification that uses an FPGA as the acceleration board. An example of such accelerator is the framework HAVEN [10]. The DUT (in our case the robot controller) will be placed on the FPGA. The outputs from the FPGA are compared to the outputs of the reference model and they represent also the inputs that are propagated to the simulation of the mechanical part. Thus, the output of the DUT stimulates the movement of the mechanical part of the robot in the simulated maze. The inputs for the FPGA and for the reference model are data from the sensors of the mechanical part of the robot.

7 Goals of the Ph.D. Thesis

In previous text, problems associated with faults in FPGA were presented , in particular those related to the evaluation of the quality of the fault tolerance methodologies. From mentioned findings the goals of the Ph.D. thesis titled *Use of verification for evaluation fault tolerance systems based on FPGAs* arise:

- Create an electromechanical application as an experimental system for testing and validating the fault tolerance methodologies.
- Create a platform for the evaluation quality of fault tolerance methodologies based on the interconnection of two techniques: verification of digital circuits and fault injection.
- The proposition of processes for effective ensuring fault tolerance with using implemented platform.

In this paper, the first version of the platform was presented, now without the use of verification techniques connected with fault injector.

8 Conclusion and Future Work

In this paper, we introduced the evaluation platform for estimating reliability of FPGA designs. As our research focuses on testing EM applications, we presented the experimental design which is composed of the mechanical robot and its electronic controller situated in the FPGA. The robot controller contains a variety of components. During the experiments, random faults were artificially injected into these components and we were monitoring impact of these faults on the behaviour of the robot in the simulation environment. These experiments showed that some faults have an impact on the behaviour of the robot, and others do not have. According to this result we were able to identify the parts/components of the robot controller that need to be hardened by some fault-tolerance techniques.

In addition, we have recognised from the experiments that some kind of automation is unavoidable in our future experiments, especially in the early phases of testing. The reason is that monitoring the behaviour of system in simulation is very time-demanding. Therefore, we have already prepared an innovative extension of our platform - interconnection of fault injection and functional verification environment with advanced test generation. Using this approach we will be able to automatically verify an EM system during the fault injection. The automation is achieved by comparing the outputs of the verified system to the reference model that is in our case represented by the same design but without injected faults.

Acknowledgment

This work was supported by the following projects: BUT project FIT-S-14-2297, National COST LD12036, project IT4Innovations Centre of Excellence (ED1. 1.00/02.0070) and COST Action project "Manufacturable and Dependable Multicore Architectures at Nanoscale".

References

[1] S. Cutts, "A collaborative approach to the more electric aircraft," in *Power Electronics, Machines and Drives, 2002. International Conference on (Conf. Publ. No. 487)*, June 2002, pp. 223–228.

[2] M. Ceschia, M. Violante, M. Reorda, A. Paccagnella, P. Bernardi, M. Rebaudengo, D. Bortolato, M. Bellato, P. Zambolin, and A. Candelori, "Identification and classification of single-event upsets in the configuration memory of SRAM-based FPGAs," *Nuclear Science, IEEE Transactions on*, vol. 50, no. 6, pp. 2088–2094, 2003.

[3] L. Sterpone, M. Aguirre, J. Tombs, and H. Guzmán-Miranda, "On the Design of Tunable Fault Tolerant Circuits on SRAM-based FPGAs for Safety Critical Applications," in *DATE '08: Proceedings of the conference on Design, automation and test in Europe*. New York, NY, USA: ACM, 2008, pp. 336–341.

[4] N. Rollins, M. Fuller, and M. Wirthlin, "A comparison of fault-tolerant memories in sram-based fpgas," in *Aerospace Conference, 2010 IEEE*, 2010, pp. 1–12.

[5] M. Naseer, P. Sharma, and R. Kshirsagar, "Fault tolerance in fpga architecture using hardware controller - a design approach," in *Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on*, 2009, pp. 906–908.

[6] M. Straka, J. Kastil, and Z. Kotasek, "Seu simulation framework for xilinx fpga: First step towards testing fault tolerant systems," in *14th EUROMICRO Conference on Digital System Design*. IEEE Computer Society, 2011, pp. 223–230.

[7] J. Podivinsky, M. Simkova, and Z. Kotasek, "Complex Control System for Testing Fault-Tolerance Methodologies," in *Proceedings of The Third Workshop on Manufacturable and Dependable Multicore Architectures at Nanoscale (MEDIAN 2014)*. COST, European Cooperation in Science and Technology, 2014, pp. 24–27.

[8] N. Dorairaj, E. Shiflet, and M. Goosman, "Planahead software as a platform for partial reconfiguration," *Xcell Journal*, vol. 55, no. 68-71, p. 84, 2005.

[9] C. Lavin, M. Padilla, P. Lundrigan, B. Nelson, and B. Hutchings, "Rapid prototyping tools for fpga designs: Rapidsmith," in *Field-Programmable Technology (FPT), 2010 International Conference on*, Dec 2010, pp. 353–356.

[10] M. Simkova, O. Lengal, and M. Kajan, "Haven: An open framework for fpga-accelerated functional verification of hardware," Tech. Rep., 2011. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php.en?id=9739

Faktorizace přirozených čísel metodou eliptických křivek využívající HPC systémy

Daniel Kobrle
Počítačová bezpečnost, 1. ročník, full-time studium
Supervisor: Róbert Lórencz

ČVUT FIT
Thákurova 9, 160 00 Praha 6

daniel.kobrle@fit.cvut.cz

Abstract. Výpočetní náročnost faktorizace velkých čísel stojí v cestě při útocích na většinu asymetrických šifer. *Metoda eliptických křivek* (ECM) je považována za jednu z nejlepších pokud jde o čísla řádově kolem 200b, což přímo neumožňuje útok na *RSA*, avšak lze tuto metodu využít například jako stavební prvek některé ze sofistikovanějších metod řešení, jakou je například *GNFS*.
V tomto článku představujeme novou metodu přístupu k řešení faktorizace velkých čísel s využitím ECM, zaměřenou na HPC systémy TIER. Naše metoda *SPHERE* (Scalable Parallel HPC Efficient Realization of ECM), kombinuje v současné době nový výpočetní postup využívající afinních souřadnic se známým algoritmem na inverzi Left-Shift. Navrhovaná metoda *SPHERE* je v současné době v pokročilém stádiu vývoje a probíhá optimalizace použitých algoritmů pro potřeby nasazení na HPC systémech. Naším cílem je rychlostně překonat ostatní používané souřadné systémy jako jsou *projektivní*, či *Jacobiho* a vyvrátit tak domněnku, že razantní zvýšení počtu všech matematických operací je výhodné.

Keywords. ECM, HPC, TIER, SPHERE, afinní souřadný systém, Left-Shift, double-and-add, modulární inverze, operační složitost, faktorizace

1 Úvod

Pro faktorizaci čísel máme na výběr z množství algoritmů. Nejpoužívanějším je v současné době General Number Field Sieve (GNFS), který je možno v omezené míře využít k útoku na RSA, obecně se jedná o metodu vhodnou pro čísla s vysokými prvočíselnými faktory. Elliptic Curve Method (ECM) je jednou z metod vhodných pro faktorizaci menších čísel, kde však zaujme svou rychlostí. Implementací ECM na specializovaném hardwaru se zabýval již například Franke a kolektiv [3] s realizací SHARK na FPGA, o 3 roky později Gaj a kolektiv [4] taktéž na FPGA, či Bernstein a kolektiv [5] s realizací na GPU. Naším cílem je realizovat ECM uzpůsobenou pro paralelní počítače typu TIER, která by nabízela vyšší výpočetní potenciál než předchozí realizace a potvrdila naše závěry o výpočetních složitostech souřadných systémů, shrnutých v sekci 3. Abychom dosáhli našeho cíle, snažíme se ECM dále urychlit pomocí speciálních algoritmů a vzorců, které představujeme v této práci.

V sekci 2 popisujeme metodu ECM, sekce 3 se zaměřujeme na výpočetní náročnosti operací nad eliptickou křivkou v různých souřadných systémech, sekce 4 obsahuje náš navrhovaný postup pro výpočet ECM - *SPHERE*.

2 Metoda eliptických křivek - ECM

ECM principiálně čerpá z Pollardovy (p-1) metody a odstraňuje nedostatek, kterým byla možnost generovat pouze jedinou multiplikativní grupu pro daný modulus. Nyní popíšeme ECM publikovanou H. W. Lenstrou [1].

2.1 Popis algoritmu

Nechť $E/\mathbb{Q}$ je eliptická křivka, N přirozené číslo s nejméně dvěma prvočíselnými děliteli z nichž jeden označíme q a bod $P \in E(\mathbb{Q})$. Redukci modulo q definujme jako $E(\mathbb{Q}) \to E(\mathbb{F}_q)$, tedy $Q \to \overline{Q}$. Hledáme takový bod $\overline{P} \in E(\mathbb{F}_q)$, jehož *faktorizační základ* splňuje stanovené podmínky a povede k faktorizaci čísla N.

Faktorizační základ označuje množinu čísel, takových, s jejichž výhradním použitím jsme schopni dané číslo faktorizovat. Velikost nejvyššího čísla obsaženého ve faktorizačním základu ovlivňuje rychlost, s jakou jsme schopni dané číslo faktorizovat s pomocí ECM. Pokud je číslo složeno z velkého množství faktorů s malou bitovou velikostí, faktorizaci nalezneme velmi rychle.

Nechť $\mathbb{G}$ je faktorizační základ $\mathbb{G} = \{1, 2, 3, ..., g_n\}$ a platí, že $\forall i \in \{1, \cdots, n\}, g_i \leq B$. Takové číslo označujeme jako *B-hladké*. Volba parametru B udává horní mez běhu algoritmu v první fázi, popsaného v Algoritmu 1.

Algoritmus 1 ECM

```
1: i = 1
2: while i ≤ B do
3:     vypočti P = iP
4:     i++
5: end while
```

Výpočet probíhá v prostoru modulo N a využívá operace *sčítání* a *násobení bodu*. Pokud v průběhu jedné z těchto operací není možné vypočítat inverzi, řád bodu P je násobkem řádu podgrupy q, získáváme prvočíselný faktor q. Hodnoty násobku i mohou být zvoleny různě, například pouze jako lichá čísla, či můžeme vybírat pouze prvočísla.

Algoritmus ECM lze rozšířit o druhou fázi běhu v případě, že první končí neúspěchem. V takovém případě testujeme, zda $kR \equiv O \pmod{q}$, kde k je prvočíslo $B < k \leq B_2$ a B_2 zvolený parametr.

Při neúspěchu algoritmu, volíme jiné parametry křivky, případně i jiný počáteční bod. Tato změna parametrů ovlivní řády podgrup generovaných děliteli čísla N a tedy složitost nalezení jeho dělitelů v rámci stanovených mezí B a B_2.

3 Operační náročnost výpočtů nad eliptickou křivkou

Pro výpočet násobku bodu P na eliptické křivce využíváme *sčítání* a *násobení bodu*. Tyto operace lze provést v různých souřadných systémech a měnit tak instrukční náročnost pro potřeby konkrétní implementace. Rozlišovat budeme mezi operacemi *inverze* - INV, *násobení* - MUL, *mocnění* - POW, *sčítání* - ADD, *odečítání* - SUB a *bitový posuv* - SHIFT.

3.1 Porovnání souřadných systémů

Porovnáme z hlediska složitosti nejpoužívanější souřadné systémy - *afinní*, *projektivní* a *Jacobiho*. Výpočty ve všech souřadných systémech vycházejí z následujících rovnic pro afinní souřadný systém. Odvozené souřadné systémy za pomocí rozšíření 2D prostoru o další dimenze eliminují při výpočtu směrnice λ inverzi, za cenu zvýšení počtu ostatních operací, jako je násobení, mocnění, atd.

Systém	Operace	INV	MUL	POW	ADD	SUB	SHIFT
Afinní	Sčítání bodů	1	2	1		6	
	Zdvojování bodu	1	2	2	2	3	3
Projektivní	Sčítání bodů	(-1)	15 (+13)	6 (+5)		6	1 (+1)
	Zdvojování bodu	(-1)	7 (+5)	6 (+4)	2	3	13 (+10)
Jacobiho	Sčítání bodů	(-1)	15 (+13)	9 (+8)		6	1 (+1)
	Zdvojování bodu	(-1)	4 (+2)	6 (+4)	2	3	8 (+5)

Tabulka 1: Porovnání počtu operací vzhledem k Afinnímu souřadnému systému

Pro zdvojení bodu $P[X_P, Y_P]$ platí:

$$\lambda = \frac{3X_P^2 + a}{2Y_P}$$
$$X_R = \lambda^2 - 2X_P$$
$$Y_R = \lambda(X_P - X_R) - Y_P$$

Pro sčítání bodů $P[X_P, Y_P]$ a $Q[X_Q, Y_Q]$, kde $P \neq Q$, platí:

$$\lambda = \frac{Y_Q - Y_P}{X_Q - X_P}$$
$$X_R = \lambda^2 - X_P - X_Q$$
$$Y_R = \lambda(X_P - X_R) - Y_P$$

Operace jsou definovány na křivce popsané Weierstrassovou rovnicí $y^2 = x^3 + ax + b$. Podrobné přepisy využitých rovnic a jejich substitucí pro zbylé souřadné systémy je možno nalézt v [8]. Výsledky provedené analýzy shrnuje Tabulka 1.

Z výsledků je patrný nejmenší počet operací u *afinního* systému. Je však nutno vzít v potaz výpočetní náročnost inverze, která je vyšší než u ostatních operací. Rozdíl v náročnosti je možno snížit realizací za pomoci speciálních algoritmů, jako je Left-Shift [2], čímž se budeme dále zabývat.

4 Navrhovaný přístup k výpočtu ECM

Z provedené analýzy v předchozí sekci lze pozorovat množství výpočetních operací, které je nutno provést pro eliminaci výpočtu inverze. Naším cílem je provést inverzi v rámci tohoto počtu operací a předejít tak převodu mezi souřadnými systémy. *SPHERE* využívá algoritmu Left-Shift [2] pro realizaci inverze, upravuje rovnice výpočtu bodů pro potřeby Double-and-Add (D&A) algoritmu a následný výpočet distribuuje po výpočetních uzlech v rámci HPC systému.

4.1 Optimalizace výpočtu D&A

ECM nevyžaduje znalost obou souřadnic X a Y u všech mezilehlých bodů. Toho využijeme společně s faktem, že v rámci každé iterace jsou jen dně možnosti jak lze pokračovat:

- Zdvojení - „Double“

- Zdvojení a sečtení - „Double and Add“

Samotné „sečtení bodů“ není prováděno nikdy. Toho využívá námi navrhované vylepšení D&A výpočtu, které provádí zdvojení a sčítání bodů v jednom kroku. Výhodou takového přístupu je v obecném případě eliminace 25% inverzí během výpočtu za přijatelnou cenu zvýšení počtu ostatních operací. Výpočet je realizován pomocí následujících rovnic:

$$A = 3X_P + a \quad B = A^2 \quad C = 2Y_P \quad D = 2Y_P^2 \quad E = 2D \quad F = ((E \cdot X_T) - B + 2E \cdot X_P)$$
$$G = (F \cdot C)^{-1} \quad H = G \cdot F \quad X_R = (A \cdot H)^2 - 2X_P \quad I = (Y_T \cdot C - A(X_P - X_R) + D)E$$
$$J = I \cdot G \quad X_Q = J^2 - X_T - X_R \quad Y_Q = J(X_T - X_Q) - Y_T$$

Bod P je aktuální bod na křivce v rámci D&A algoritmu, na tento bod aplikujeme *zdvojení* a výsledný bod R je *sečten* s bodem T, který je počátečním bodem při vstupu do D&A algoritmu a v jeho průběhu se nemění. Výsledkem je bod Q.

Operační náročnost tohoto postupu je:

$$\Theta(1INV + 10MUL + 5POW + 4ADD + 8SUB + 6SHIFT)$$

což vyjádřeno relativně oproti postupnému aplikování sečtení a zdvojení bodu znamená následující:

$$\Delta\Theta_A(-1INV + 6MUL + 2POW + 2ADD - 1SUB + 3SHIFT)$$

Náročnost operací ADD a SUB lze v tomto případě zanedbat. Získáváme tak výpočet inverze za cenu $6MUL$ a $2POW$ operací. Pro testování efektivity s knihovnou GMP jsme využili jak knihovní funkci pro inverzi *mpz_inverse*, tak algoritmus *Left-Shift*. Výpočty probíhaly v afinním souřadném systému a měřena byla doba běhu standardního algoritmu D&A a verze využívající výše popsaných rovnic. Výsledné urychlení bylo řádově 10%, což poskytlo hrubou představu o složitosti výpočtu inverze v aktuálním stavu.

Pokud postup porovnáme s operační složitostí v projektivním souřadném systému, rozdíl bude znatelnější:

$$\Delta\Theta_P(1INV - 12MUL - 6POW + 2ADD - 1SUB - 9SHIFT)$$

Z rozdílu v počtu operací s přihlédnutím na naměřené urychlení v rámci afinních souřadnic $\Delta\Theta_A$ plyne, že operační složitost $12MUL$ a $6POW$ poskytuje dostatečný prostor pro výpočet inverze.

4.2 Výpočet inverze algoritmem Left-Shift

Optimalizace výpočtu inverze je základem množství algoritmů. Jmenujme například modulární inverzi v Montgomeryho bázi [6] nebo Right-Shift algoritmus připisovaný *M. Penkovi* [7]. Oba algoritmy dosahují dobrých rychlostí v porovnání s *rozšířeným Euclidovým algoritmem*. Ještě lepších výsledků dosahuje algoritmus Left-Shift [2]. Tento algoritmus je primárně navržen pro HW zařízení a využívání operací jako je bitový posun, sčítání a odečítání. Tyto operace lze v HW realizovat velmi rychle, což je samozřejmě oproti SW realizaci rozdíl.

V současné době pracujeme na optimalizaci tohoto algoritmu pro realizace v SW s využitím knihovny GMP, které se žádá úpravy stávající formy. Disponujeme již fungujícím prototypem realizovaným s pomocí high-level funkcí GMP a nyní se zaměřujeme na realizaci s využitím low-level funkcí GMP, které jsou obvykle napsány v assembleru a nebývají obaleny dalšími testy a funkcemi pro zajištění koherence dat. Taková realizace je náročnější, měla by však přinést další urychlení.

4.3 Možnosti paralelního zpracování výpočtu

Pro implementaci je možno vycházet ze dvou základních schémat rozložení:

1. *křivka na procesor*
2. *křivka na uzel*

Další možností je *hybridní rozložení*, které kombinuje obě možnosti. Část procesorů se v takovém případě řídí Schématem 1 a druhá část Schématem 2. Pro konečnou realizaci se právě toto řešení zdá nejlepší volbou, neboť umožní využívat výhod obou schémat a není tedy tak citlivé na správné nastavení parametrů B, respektive B_2.

4.3.1 Křivka na procesor

1. schéma představuje jednoduchý přístup s absencí jakékoliv komunikace s ostatními výpočetními uzly. Přístup může být navržen tak, že volba parametrů další křivky bude odvozena od počtu procesorů na úlohu, označme n, a čísla procesoru v této množině, označme p. Parametr a *k-té* křivky je potom možné definovat jako $a = (p + k \cdot n)$. Hodnotu parametru B lze měnit pomocí heuristické, či inkrementální, funkce, avšak neznámý řád grupy neumožňuje provádět tyto změny sofistikovaným způsobem.

- Parametr B se může ukázat jako příliš malý a tak neumožní nalézt faktor.
- Rychle narůstá počet testovaných křivek a tím pádem možnost nalézt nejvhodnější křivku.

4.3.2 Křivka na uzel

2. schéma umožňuje využít lépe výpočetního potenciálu celého uzlu (node). Procesory sdílející křivku si v definovaných rozestupech volí intervaly pro parametr B. Každý procesor vychází ze stejného počátečního bodu na křivce a v daném intervalu B použije algoritmus ECM. Po dosažení meze B každý procesor informuje řídící procesor a pokračuje dále ve výpočtu *druhou fází*. Jakmile řídící procesor získá informace o dosažení meze B všemi procesory, rozesílá příkaz pro ukončení výpočtu a zaslání souřadnic posledního bodu. Tyto body následně sečte a protože operace byly prováděny nad stejným počátečním bodem, řád bodu se adekvátně zvýší. Pokud nebyla inverze nalezena, je vygenerována další křivka a postup se opakuje.

- Procesory v uzlu určitou dobu vykonávají stejné výpočty.
- Nalézt vhodnou křivku trvá delší dobu.
- Rychleji vypočteme vysoký řád bodu kombinací všech mezivýsledků.

5 Cíle dizertační práce a závěr

Postupy popsané v tomto článku jsou základem běžícího projektu faktorizace s využitím ECM. Projekt je primárně zamýšlen pro HPC systémy typu TIER, nejedná se tak o aplikaci pro specializovaný HW typu GPU či FPGA, což ovlivňuje efektivitu jednotlivých metod, avšak nabízí možnost nasazení na mnohem výkonnějších paralelních sestavách. Výzvou se tak stává optimalizace použitých algoritmů a návrh přístupu k řešení výpočtu, které mají stěžejní význam pro rychlost realizace *SPHERE*. Naším cílem je realizace metody ECM, která nabídne vyšší rychlost než metody využívající pro výpočet odlišné souřadné systémy, tuto metodu otestovat na superpočítačích a otestovat vlastnosti této metody na vyšších číslech, než v předchozích publikacích zaměřujících se na implementace této metody - optimálně tedy čísla kolem

300b a výše. Dosažení dobrých výsledků na takto vysokých číslech by mohlo ovlivnit rámec bezpečnosti určitých konfigurací asymetrických šifer, neboť při využití ECM jako rychlého dílčího výpočtu pro GNFS, bychom byli schopni faktorizovat opět o něco větší čísla poměrně efektivně.

Našemu projektu byl v současné době přiřazen grant *OPEN-3-14* na HPC systému TIER-1 v rámci soutěže pořádané *it4innovations*, ve kterém navrhované metody a postupy plánujeme realizovat a publikovat další výsledky práce. V tuto chvíli je třeba dále optimalizovat výpočet inverze a realizovat paralelní zpracování pomocí MPI knihovny. Plánujeme otestovat různé možnosti přístupu k dělení řešení na jednotlivé uzly a procesory, stejně jako možnosti generování násobku bodu pomocí různých číselných řad, či například pomocí náhodných vysokých čísel s definovanou minimální Hammingovou váhou.

Reference

[1] Lenstra, H., W., Jr.: Factoring Integers with Elliptic Curves, The Annals of Mathematics, 1987, Volume 126, pp. 649—673

[2] Lórencz, R.: New algorithm for classical modular inverse, Cryptographic Hardware and Embedded Systems-CHES, 2003, pp. 57—70

[3] Franke, J., Kleinjung, J., Paar, Ch., Pelzl, J., Priplata, Ch., Šimka, M., Stahlke, C.: An Efficient Hardware Architecture for Factoring Integers with the Elliptic Curve Method, Special-Purpose Hardware for Attacking Cryptographic Systems — SHARCS, 2005

[4] Gaj, K., Kwon, S., Baier, P., Kohlbrenner, P., Le, H., Khaleeluddin, M., Bachimanchi, R.:Implementing the Elliptic Curve Method of Factoring in Reconfigurable Hardware, Cryptographic Hardware and Embedded Systems - CHES, 2006, pp. 119—133

[5] Bernstein, D. J, Chen, T. R, Cheng Ch. M, Lange, T, Yang, B. Y: ECM on Graphic Cards, EUROCRYPT, 2009, pp. 483—501

[6] Kaliski, B., J., Jr.: The Montgomery Inverse and Its Application, IEEE Transaction on Computers 44 No. 8, 1995, pp. 1064—1065

[7] Knuth, D., E.: The Art of Computer Programming 2 / Seminumerical Algorithms, Addison-Wesley, Reading, Mass. Third Edition, 1998

[8] Kobrle, D.: Implementační aspekty kryptografie eliptických křivek, DP ČVUT, 2013

Komponenty pro polymorfní číslicové obvody na bázi ambipolárních tranzistorů

Radek Tesař
Informatika a výpočetní technika, ročník první, kombinované studium
Školitel: Richard Růžička

FIT VUT Brno
Božetěchova 2, Brno

itesar@fit.vutbr.cz

Abstrakt. Téma disertace je experimentovat s nekonvenčními technologiemi (polymorfní elektronika, tištěná elektronika, prvky na bázi nanostruktur), nalézt vhodná řešení a aplikace, kde použití nekonvenčních řešení přináší výhody. Navrhnout výhodné způsoby kombinace nekonvenčních technologií s konvenční elektronikou.

Klíčová slova. Ambipolární tranzistor, nanodráty, grafenový tranzistor, tištěná elektronika, organická elektronika, polymorfní elektronika, logické hradla, číslicové obvody.

1 Úvod

V současné době se na poli polovodičových součástek objevují materiály, které mají ambice nahradit křemíkové struktury. Takovými materiály jsou například organické polovodiče [1], které mají mimo jiné řadu zajímavých vlastností. Příkladem takové vlastnosti může být ambipolarita – unipolární tranzistor tvořený takovým materiálem se na základě určitých podmínek může chovat jako tranzistor P-typu, zatímco za jiných podmínek pak jako tranzistor N-typu. Tento tranzistor se dá využít při vývoji polymorfní elektroniky. Ta má ambice zjednodušit elektronické obvody, nebo vnést do zapojení další funkcionalitu [3]. Toho se dá využít například při změně prostředí, ve kterém se zařízení s polymorfní elektronikou nachází (řídící obvod solární elektrárny bude mít jinou funkci za denního světla a jinou v noci), nouzovém nebo havarijním stavu (vlivem zvýšení teploty se řídící elektronika přepne do nouzového stavu), a podobně.

V principu polymorfismus funguje tak, že obvod, který má v normálním režimu funkci f_1, se při změně prostředí (nouzový stav, porucha napájení, atd.) rekonfiguruje a tím změní svoji funkci na f_2 [2] [3]. Takové chování je běžné například u mikroprocesorů nebo hradlových polí. Ty však mají jiné negativní vlastnosti (nutnost použít větší počet logických členů a tím větší spotřeba, pomalá rekonfigurace, riziko chyby programu, a podobně). Z uvedeného je tedy zjevné, že polymorfní obvody by neměly mít tyto negativní vlastnosti. Musí být snadno a rychle rekonfigurovatelné (jednoznačná a rychlá odezva na požadovaný podnět), díky využití stejných obvodů (hradel a logických celků) pro dvě různé funkce by měly být menší, než stejné obvody realizované konvenční technologií (nutnost použít pro každou funkci jiný obvod). Nehrozí u něj také chyby programu, protože tyto obvody není nutno programovat (jejich funkce je dána zapojením, stejně jako u klasických číslicových obvodů).

Na základě toho byla stanovena hypotéza, že pro určitou třídu aplikací bude implementace s použitím polymorfních logických hradel s ambipolárními tranzistory efektivnější co do velikosti než implementace konvenčními logickými obvody.

Většina současných polymorfních obvodů využívá obvody založené na MOS (Metal Oxide Semiconductor) technologii, například [2], nebo CMOS technologii [4]. To jsou však běžné křemíkové technologie, které nejsou pro polymorfní obvody příliš vhodné. Pro získání polymorfních vlastností křemíkové technologie se používají různé triky, například různá velikost použitých tranzistorů na čipu. Typickým příkladem takových hradel jsou již zmiňované [2], nebo [4]. Protože se tak snaží dosáhnout neobvyklých vlastností křemíkových prvků, dochází zároveň ke zhoršení jiných parametrů čipu (vyšší spotřeba, nižší mezní frekvence, atd.). Tyto problémy nemají zmiňované organické materiály. Ty mají přirozeně polymorfní vlastnosti jak bude uvedeno dále, navíc lze prvky z organických materiálů používat dříve nevídaným způsobem. Příkladem může být tisk organických tranzistorů a celých logických obvodů na inkoustové tiskárně [5], [6]. Tím se otevírá široké pole použití elektronických obvodů například ve wearable electronics, potisk oděvů pro jejich digitální ochranu a zatraktivnění, nebo tisk elektronických obvodů na papír, které se pak stanou součástí novin, knih, nebo jiných tiskovin.

Pokud chceme začít využívat polymorfní obvody, je nutno nejprve realizovat základní logické funce (hradla), ze kterých budeme následně tvořit větší logické prvky. Polymorfní obvody mohou měnit logické funkce reakcí na změnu vstupního signálu (například pomocný gate ambipolárního tranzistoru), což vyžaduje implementaci dalšího vstupního pinu elektronického obvodu, nebo reakcí na změnu prostředí, která se distribuuje nezávisle v celém obvodu a nevyžaduje žádný další vstupní pin. Tím může být zmíněná změna teploty nebo osvětlení prostředí, ve kterém se polymorfní elektronika nachází, změna velikosti napájecího napětí, změna polarity tohoto napětí a podobně.

Cílem je tedy vytvořit systém polymorfních hradel, které budou reagovat na změnu prostředí a bude z nich možno sestavit libovolný logický obvod. Avšak tyto obecné logické obvody nebudou předmětem našeho výzkumu. Použití ambipolárních tranzistorů přímo nabízí jako vhodný signál změnu polarity napájecího napětí obvodu, proto se budeme následně zabývat pouze obvody reagujícími na tuto změnu. Pro tento cíl bude nutno vytvořit úplný systém logických funkcí – nejlépe funkci NAND nebo NOR (některé již byly vytvořeny, viz [7]).

Protože však žádný obvod není složen výhradně z polymorfních hradel, je nutno navrhnout také rezistentní hradla. To znamená takové, které nebudou na tuto změnu reagovat. Ty si uchovají svoji funkcionalitu při jakékoliv polaritě. Standardní logické hradla nelze přepólovat bez rizika zničení, navíc po přepólování nebudou fungovat. To je možno řešit přímočaře tím, že každé hradlo bude mít svůj Graetzův můstek v napájecí části. To řeší problém funkce hradla při změně polarity napájení, ale zvýší obvodovou složitost. Cílem však je přidat další funkci a navíc zachovat obvodovou složitost, nebo ji ještě snížit. Nejjednodušším takovým rezistentním hradlem je běžný invertor složený z ambipolárních tranzistorů.

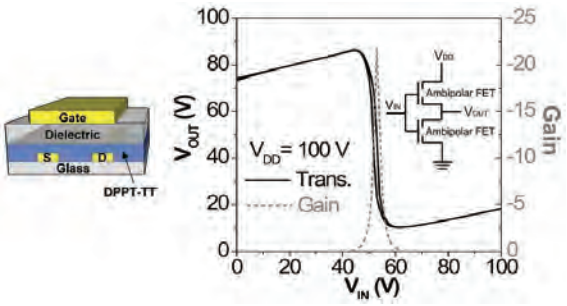
Mimo tyto logické funkce však bude pro realizaci polymorfních obvodů potřeba ještě další méně běžná polymorfní hradla, například identita/negace, nebo dvouvstupý multiplexor. Všechny tyto hradla budou uvedeny dále.

2 Ambipolární polovodiče

Pro konstrukci ambipolárních tranzistorů se používá organický polovodič, uhlíkové nanotrubičky, grafen, a podobně. Na obrázku 1 je vidět ambipolární tranzistor, který je tvořen Diketopyrrolopyrrole-Thieno [3,2-b]thiophene kopolymerem [1]. Strukturu tohoto tranzistoru tvoří D-A kopolymer DPPT-TT.

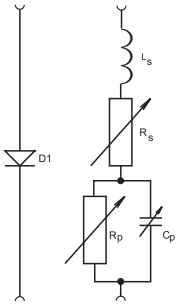
Vpravo na stejném obrázku je přechodová charakteristika a zesílení komplementárního invertoru tvořeného dvěmi stejnými tranzistory. Díky ambipolaritě tranzistorů se jeden z nich chová jako tranzistor typu P a druhý jako typu N. Této vlastnosti pak lze dále využít při konstrukci polymorfních hradel a elektroniky z nich složených.

V současné době jsou ambipolární tranzistory předmětem intenzivního vývoje, nelze tedy zatím jednoznačně definovat jejich typické vlastosti (například životnost, stabilitu, hystereze atd). Různí se také názory na použití materiálů, že kterých jsou tyto tranzistory tvořeny, stejně jako použitá výrobní technologie. Proto je třeba zatím vyčkat na stabilizaci trhu s těmito tranzistory.



Obrázek 1: Schéma ambipolárního tranzistoru (vlevo), přechodová charakteristika a zesílení invertoru.

2.1 Model polovodičové diody



Obrázek 2: Náhradní schéma diody.

Základní prvek, který je nutný pro konstrukci uvedených rezistentních hradel, je polovodičová dioda. Je to dvojpól, který využívá vlastností přechodu PN. To je oblast na rozhraní příměsového polovodiče typu P a polovodiče typu N. Přechod P-N se chová jako hradlo, tzn. propouští elektrický proud pouze jedním směrem [8].

Ačkoli krystalová mřížka obou částí diody na sebe plynule navazuje, vzniká v okolí přechodu PN vlivem elektrostatického pole pevně vázaných iontů akceptoru a donoru vyprázdněná oblast, která se chová jako izolační vrstva oddělující navzájem část P od části N. Na vyprázdněnou oblast mezi polovodičem P a N můžeme pohlížet také jako na deskový kondenzátor o ploše desky rovné ploše PN přechodu a vzdáleností desek rovnou šířce vyprázdněné oblasti, nepůsobí-li na přechod vnější napětí. Tento kondenzátor má tzv. Bariérovou kapacitu, která způsobuje vedení el. proudu v závěrném směru při vysokých frekvencích signálu. Odpovídající kapacita je dost velká, neboť relativní permitivita křemíku je 12, germania 16 a arzenidu galia 11. Její velikost dosahuje podle plochy přechodu hodnoty několik pikofaradů až několik desítek nanofaradů [8].

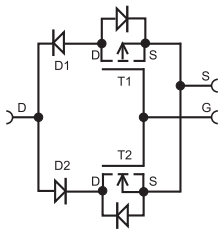
Na obrázku 2 vlevo je schématická značka polovodičové diody a vpravo náhradní schéma této diody dle [8], strana 96. Obvod Rp, Cp nahrazuje PN přechod diody a je doplněn odporem Rs představující odpor zbývajícího polovodičového materiálu a přívodů. Stejně tak indukčnost přívodů diody znázorňuje cívka Ls. Ta se uplatňuje při velmi vysokých frekvencích.

2.2 Model ambipolárního tranzistoru

Na obrázku 3 je model ambipolárního tranzistoru tvořený mosfet tranzistory. Každý unipolární mosfet tranzistor obsahuje z principu body diodu, proto je nutno eliminovat jejich vliv antisériovými diodami D1 a D2.

Funkci modelu popisuje tabulka 4. Sloupce D,S,G označují jednotlivé vývody modelu tranzistoru. Nabývají hodnoty + nebo -, což odpovídá napájecímu napětí (Vcc, GND). D1, D2 značí diody modelu a mají hodnoty P (propustný směr) nebo Z (závěrný směr). T1 a T2 jsou tranzistory modelu, kde hodnota OFF znamená, že tranzistor je zavřený a ON že je otevřený. Sloupec D-S značí chování modelu, kde HiZ je High Impedance (model „rozpojen“), ON znamená, že model v dané konfiguraci propouští proud.

Z důvodů prozatímní nedostupnosti reálných ambipolárních tranzistorů jsme prakticky realizovali uvedený model a pro všechny pokusy s polymorfními nebo rezistentními hradly byly použity ambipolární tranzistory sestavené ze silikonových mosfet tranzistorů.



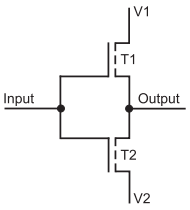
Obrázek 3: Model ambipolárního tranzistoru.

D	S	G	D1	D2	T1	T2	D-S
+	-	-	Z	P	OFF	OFF	HiZ
-	+	-	P	Z	OFF	ON	ON
+	-	+	Z	P	ON	OFF	ON
-	+	+	P	Z	OFF	OFF	HiZ

Obrázek 4: Popis stavů modelu ambipolárního tranzistoru

2.3 Ambipolární invertor

Pro naše pokusy jsme zvolili nejběžnější pozitivní logiku. To znamená, že logickou 0 bude představovat napětí blízké GND a logickou 1 napětí blízké V_{cc}. Tuto konvenci budeme dodržovat v celém dokumentu.



Obrázek 5: Ambipolární invertor

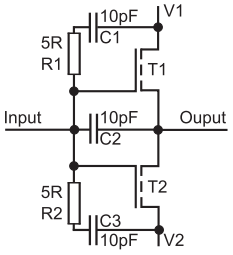
In	Out	V1	V2	T1	T2
0	1	-	+	OFF	ON
1	0	-	+	ON	OFF
0	1	+	-	ON	OFF
1	0	+	-	OFF	ON

Obrázek 6: Popis stavů invertoru

Nejjednodušším hradlem, vytvořeným z ambipolárních organických tranzistorů, je invertor, viz např. [1]. Jeho zapojení je na obrázku 5. U invertoru ze silikonových tranzistorů je horní tranzistor typu P a spodní typu N. Použijeme-li ambipolární tranzistory, jsou oba stejného typu, takže se mění typ tranzistoru podle jeho zapojení. Díky tomu je ambipolární invertor rezistentní vůči přepólování napájení – pokud prohodíme Vcc a GND, změní se podle toho také typ tranzistorů (N na P a obráceně). Tohoto principu se využívá v polymorfních hradlech NAND/NOR, jak bylo popsáno například v [7].

3 Rezistentní ambipolární hradla

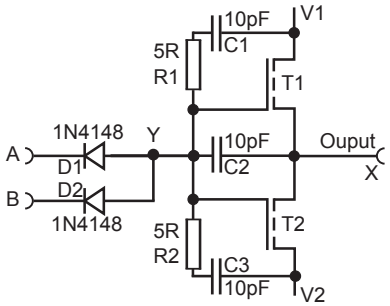
Pokud se zaměříme na rezistentní hradla, využijeme vlastnosti ambipolárních tranzistorů, kterou je možno pozorovat na obrázku 1. Mezi gate a elektrodami (Drain, Source) je dielektrikum, díky kterému tvoří elektrody kondenzátor. Na obrázku 7 je vidět náhradní schéma ambipolárního invertoru, kde gate nahradíme kondenzátory. Pokud v této konfiguraci připojíme vstup invertoru na GND, nabije se horní kondenzátor, zatímco spodní zůstane vybitý. Pokud připojíme tento vstup na V_{cc}, nabije se naopak spodní kondenzátor a horní zůstane vybitý. To, který kondenzátor se nabije, nám pak určuje, který tranzistor se otevře (v prvním případě horní, v druhém spodní tranzistor). Tím získáme na výstupu příslušné napětí. Obvodem protéká proud pouze v době, kdy se nabíjí kondenzátory. Po jejich nabití je pak spotřeba invertoru nulová, ovšem pouze za předpokladu, že je otevřen pouze jeden tranzistor. Výše uvedeného principu tedy použijeme při tvorbě rezistentních hradel.



Obrázek 7: Náhradní schéma invertoru

3.1 NAND

Na obrázku 8 je zapojení rezistentního hradla NAND, tvořeného ambipolárními tranzistory. Pro funkci NAND se využívá diodové logiky, jejíž výsledek je pak invertován ambipolárním invertorem.



Obrázek 8: Princip zapojení hradla NAND

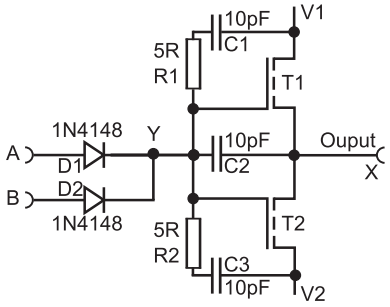
A	B	X	V1	V2	D1	D2	C1	C2	T1	T2
0	0	1	+	-	P	P	N	V	ON	OFF
0	0	1	-	+	P	P	V	N	OFF	ON
0	1	1	+	-	P	Z	N	V	ON	OFF
0	1	1	-	+	P	Z	V	N	OFF	ON
1	0	1	+	-	Z	P	N	V	ON	OFF
1	0	1	-	+	Z	P	V	N	OFF	ON
1	1	0	+	-	Z	Z	V	V	OFF	OFF
1	1	0	-	+	Z	Z	V	V	OFF	OFF

Obrázek 9: Popis stavů rezistentního hradla NAND

Pokud jsou na vstupech A a B na obrázku 8 logické 1, jsou diody zapojeny v závěrném směru a nemůže jimi procházet žádný proud. Pokud na některý vstup A nebo B (případně na oba) připojíme logickou 0, může příslušnými diodami procházet proud, který způsobí nabití kondenzátoru C1 nebo C2 (podle polarity napájecího napětí) a tím otevření příslušného tranzistoru. Na výstupu se pak vždy objeví logická 1, nezávisle na tom, jak je polarizované napájecí napětí. Celou situaci ukazuje tabulka na obrázku 9. Význam jednotlivých sloupců je stejný jako u tabulky 4, navíc jsou zde kondenzátory, jejihž sloupec nabývá hodnot N - kondenzátor je nabitý (prakticky nabití kondenzátoru trvá nějaký čas, který však můžeme zanedbat), nebo V - kondenzátor je vybitý (vybíjení také zabere nějaký čas, který zanedbáváme).

3.2 NOR

Na obrázku 10 je zapojení rezistentního hradla NOR, tvořeného ambipolárními tranzistory. Stejně jako u předchozího hradla se pro funkci OR využívá diodové logiky, jejíž výsledek je pak invertován ambipolárním invertorem.



Obrázek 10: Schéma zapojení hradla NOR

A	B	X	V1	V2	D1	D2	C1	C2	T1	T2
0	0	1	+	-	Z	Z	V	V	OFF	OFF
0	0	1	-	+	Z	Z	V	V	OFF	OFF
0	1	0	+	-	Z	P	V	N	OFF	ON
0	1	0	-	+	Z	P	N	V	ON	OFF
1	0	0	+	-	P	Z	V	N	OFF	ON
1	0	0	-	+	P	Z	N	V	ON	OFF
1	1	0	+	-	P	P	V	N	OFF	ON
1	1	0	-	+	P	P	N	V	ON	OFF

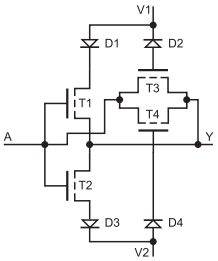
Obrázek 11: Popis stavů rezistentního hradla NAND

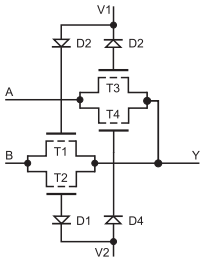
Pokud jsou na vstupech A a B na obrázku 10 logické 0, jsou diody zapojeny v závěrném směru a nemůže jimi procházet žádný proud. Pokud na některý vstup A nebo B (případně na oba) připojíme logickou 1, může příslušnými diodami procházet proud, který způsobí nabití kondenzátoru C1 nebo C3 a tím otevření příslušného tranzistoru. Na výstupu se pak objeví logická 0. V tabulce 11 jsou popsány jednotlivé stavy hradla. Význam sloupců je stejný jako v tabulce 9.

3.3 Identita – negace, multiplexer

Pro tvorbu polymorfních logických funkcí budeme dále potřebovat hradla identita/negace a dvouvstupý multiplexor. Pro jejich zapojení je typické použití transmission gate (TG) a invertorů. Přepínání funkcí

se provádí stejně jako u výše uvedených polymorfních hradel NAND/NOR, to znamená změnou polarity napájení. U polymorfního hradla ID/NOT je možné malou změnou zapojení změnit funkci na NOT/ID, což může být často potřebné.





Obrázek 12: Schéma polymorfního hradla ID-NOT Obrázek 13: Schéma polymorfního multiplexeru

Na obrázku 12 je zapojení hradla identita – negace, přepínané polaritou napájecího napětí. V případě, že V1 je kladné napětí (Vcc) a V2 zem (GND), pak hradlo funguje díky tranzistorům T1 a T2 jako invertor, tranzistory T3 a T4 jsou bez funkce (v rozepnutém stavu). Pokud zaměníme polaritu napájecího napětí, bude hradlo fungovat jako identita díky tranzistorům T3 a T4 zapojeným jako transmission gate, zatímco T1 a T2 budou bez funkce.

Podobně funguje i polymorfní multiplexer přepínaný polaritou napájecího napětí (obrázek 13). Ten je tvořen dvěmi transmission gate. První případ nastane pokud bude V1 kladné napětí (Vcc) a V2 zem (GND). Hradlo pak propojí vstup A s výstupem Y díky tranzistorům T1 a T2 které tvoří první transmission gate a tranzistory T3 a T4 jsou bez funkce (v rozepnutém stavu). Pokud opět prohodíme polaritu napájecího napětí, propojí hradlo vstup B na výstup Y díky tranzistorům T3 a T4 zapojeným jako druhý transmission gate, zatímco T1 a T2 budou bez funkce.

4 Závěr

Příklady uváděné v této práci jsou jen zlomkem možností polymorfních obvodů, ale pro jejich praktické využití je nutno nejprve zpřístupnit základní stavební prvky pro takovou elektroniku – ambipolární tranzistory a hradla z nich sestavené. Cílem práce je tedy prokázat, že lze vytvořit ucelený set logických hradel pomocí ambipolárních tranzistorů. V tom budeme dále pokračovat a vytvářet různé typy logických hradel, které pak bude možno použít při vývoji polymorfní elektroniky, stejně jako je současná elektronika tvořena například pomocí hradel řady CMOS 4000. Návrh této elektroniky však je mimo rozsah této práce. Jak bylo řečeno, v současné době zatím nelze získat prakticky použitelné ambipolární tranzistory, proto byly veškeré pokusy provedeny s modely vytvořenými ze silikonových tranzistorů. Výsledkem je experimentálně ověřený soubor takových hradel, které lze použít pro vytvoření libovolných logických funkcí. Tyto hradla byly vytvořeny pomocí CMOS tranzistorů tak, aby simulovaly chování ambipolárních tranzistorů a budou dále sloužit pro výzkum v oblasti polymorfní a tištěné elektroniky. V dalším výzkumu se budeme po zlepšení dostupnosti ambipolárních tranzistorů také postupně zaměřovat na jejich reálné použití například v komerční elektronice.

Reference

[1] *High-Performance Ambipolar Diketopyrrolopyrrole-Thieno[3,2-b]thiophene Copolymer Field-Effect Transistors with Balanced Hole and Electron Mobilities*, Zhuoying Chen, Mi Jung Lee, Raja Shahid Ashraf, Yun Gu, Sebastian Albert-Seifried, Martin Meedom Nielsen, Bob Schroeder, Thomas D. Anthopoulos, Martin Heeney, Iain McCulloch, and Henning Sirringhaus. Advanced Materials 2012, 24, pages 647 — 652. DOI: 10. 1002/adma.201102786.

[2] *Polymorphic electronics*, Stoica Adrian, Zebulum Ricardo, Keymeulen Didier. Evolvable Systems: From Biology to Hardware, 2001, pages: 291 – 302, Springer Berlin Heidelberg, ISBN: 978-3-540-42671-4 (Print), 978-3-540-45443-4 (Online)

[3] *Taking evolutionary circuit design from experimentation to implementation: some useful techniques and a silicon demonstration*, Stoica Adrian, Zebulum RS, Guo Xin, Keymeulen Didier, Ferguson MI, Duong Vu, 2004, IEE Proceedings-Computers and Digital Techniques vol. 151(4), pages: 295 – 300, doi: 10.1049/ip-cdt:20040503

[4] *REPOMO32 - New reconfigurable polymorphic integrated circuit for adaptive hardware*, Sekanina, L.; Ruzicka, R.; Vasicek, Z.; Prokop, R.; Fujcik, L., Evolvable and Adaptive Hardware, 2009. WEAH '09. IEEE Workshop on, vol., no., pages 39 – 46, April 30 2009 – March 2 2009 doi: 10.1109/WEAH.2009.4925666

[5] *Inkjet-printing-based soft-etching technique for high-speed polymer ambipolar integrated circuits* Dongyoon Khim at al, Dongguk University, Seoul, Republic of Korea, ACS Applied materials & interfaces, 2013

[6] *High-Performance Printed Carbon Nanotube Thin-Film Transistors Array Fabricated by a Non-lithography Technique Using Hafnium Oxide Passivation Layer and Mask* Sueres Kumar Raman Pillai and Marry B. Chan-Park, Nanyang Technological University, Singapore, ACS Applied materials & interfaces, 2012

[7] *Polymeric Polymorphic Electronics: Towards Multifunctional Logic Elements Based on Organic Semiconductor Materials*Růžička, R., Šimek, V., Proceedings of CSE 2012 International Scientific Conference on Computer Science and Engineerig, Košice, SK, FEI TU v Košiciach, 2012, pages 154 – 161, ISBN 978-80-8143-049-7

[8] *Elektronika* Ing. Jan Maťátko, SNTL 1987, 272 stran, ISBN 8003000386.

ENERGETICKY ÚSPORNÉ SMĚROVÁNÍ V MOBILNÍCH WSN

David Široký
Distribuované systémy, 1. ročník, prezenční studium
Školitel: Jiří Šafařík

Fakulta aplikovaných věd, Západočeská univerzita
Univerzitní 8, 306 14 Plzeň

dsiroky@kiv.zcu.cz

Abstrakt. Bezdrátové senzorické sítě, označované z pohledu směrování také jako ad-hoc bezdrátové sítě, slouží především k plošnému sběru dat z oblastí, kde není možné postavit pevnou síťovou infrastrukturu a není možné z jednotlivých uzlů posílat data přímo do bázových stanic. Důležitým aspektem je úsporný provoz, protože většina uzlů má omezené zdroje energie a je tedy kladen důraz na nízkou spotřebu.

Klíčová slova. bezdrátové senzorické sítě, směrování, mobilita, energetická úspora, optimalizace, decentralizace

1 Úvod

Základním komunikačním principem WSN je předávání zpráv z uzlu na uzel a jejich postupné doručování do bázových stanic. Uzly mají nízký vysílací výkon a „vidí“ jen své nejbližší sousedy. Úkolem směrování je doručit zprávu v co nejkratším čase a za nízkou cenu. Cenou je míněna především spotřeba energie. Tyto dva požadavky jsou ale protichůdné, protože neustálé využívání stálé nejkratší cesty co do počtu přeskoků bude znamenat, že uzly po této cestě se vyčerpají dřív, než ostatní ve zbytku sítě. Je tedy potřeba hledat kompromis.

Spotřeba energie je nejvyšší v komponentách transceiveru a řídící jednotky, např. mikrokontroléru. Ušetřit spotřebu transceiveru lze snížením množství komunikací a u mikrokontroléru uspáváním v době nečinnosti. Tento článek si klade za úkol řešit úsporu energie v rádiové komunikaci.

Základním problémem WSN je drahé získání globálního stavu. Aby bylo možné zjistit aktuální stav energetických zdrojů v celé síti, např. baterií, kdo s kým sousedí, kvality rádiových linek atd., je zapotřebí velkého množství odeslaných zpráv. Stav sítě se může navíc neustále měnit. Jakékoliv centrální zpracování je energeticky zcela nevýhodné musí se hledat distribuované metody.

Bez centrálního zpracování a s neustále měnící se sítí není možné dosáhnout optimálního směrování v rámci zadaných parametrů. Vhodná metoda by měla mít vyvážené požadované vlastnosti v daném modelu sítě.

Nejjednoduššími metodami, které fungují bez jakékoliv znalosti topologie sítě, jsou záplavové a náhodné směrování. Záplavové směrování doručuje zprávy tak, že uzel vyšle zprávu všem sousedů a ti operaci opakují. Je tedy garantováno, že se zpráva doručí do bázové stanice v nejkratším čase, ale s nadměrnou zátěží velké části sítě. U náhodného směrování pošle každý uzel zprávu vždy jen jednomu náhodnému sousedovi. V ideálním případě dorazí zpráva vinou náhody po nejkratší/nejlevnější cestě, v nejhorším případě bude zpráva „bloudit“ po síti dokud nevyprší její TTL (time to live). Postupně vznikaly

nové metody, které již berou na zřetel topologii sítě a spotřebu energie. Žádná z metod není univerzální a každá se hodí na jiný scénář a uplatnění sítě.

2 Klasifikace

Pro snazší popis vlastností směrovacích protokolů zavedeme následující klasifikaci:

Proaktivní/reaktivní - proaktivní protokoly vytvářejí směrovací tabulky předem. Buď při inicializaci sítě nebo v pravidelných intervalech. Výhodou je, že při častém posílání zpráv není nutné opakovaně zjišťovat, kterému ze sousedů ji má uzel poslat. Nevýhoda je, že se opožděně adaptuje na změny v síti a má vyšší paměťovou náročnost. Příkladem je protokol DSDV [1]. Reaktivní protokoly zjišťují trasu až když potřebují odeslat zprávu. Výhodou je rychlá adaptace na změny a uzly nemusí udržovat žádné tabulky, ale nehodí se pro časté odesílání zpráv, neboť opakovanými dotazy na směrování budou síť zahlcovat. Příkladem je protokol AODV [2].

Deterministické/pravděpodobnostní - při rozhodování, kterému sousednímu uzlu se má zpráva poslat na základě dostupných informací a aktuálního stavu, se uzel s deterministickým protokolem rozhodne vždy stejně. Naopak pravděpodobnostní protokoly přiřadí jednotlivých sousedům pravděpodobnosti podle dostupných informací a posléze zprávy mezi ně patřičně rozdělují. Výhodou determinismu je predikovatelnější čas doručení, ale v případě narušení struktury sítě se začnou všechny zprávy ztrácet, dokud nedojde opětovnému obnovení nebo reinicializaci směrování. Pravděpodobnostní přístup sice nedokáže zaručit, kdy bude zpráva doručena, ale poskytuje vyšší míru odolnosti vůči změnám a navíc stačí méně častá reinicializace směrování, pokud nejsou kladeny velké nároky na kvalitu.

(Ne)podporuje vícecestné směrování - velmi důležitá vlastnost, má-li být směrování robustní. Má-li uzel na výběr z více cest a všechny využívá, např. cyklicky nebo podle pravděpodobnosti, zvyšuje se pravděpodobnost doručení zpráv. Tento princip byl popsán v předchozím odstavci.

(Ne)podporuje výpadky uzlů/nové uzly - robustnost již byla zmíněna. Nelze předpokládat, že bude síť neměnná. Uzly se mohou poškodit, dojde jim baterie, nebo budou ukradeny. Naopak do sítě můžou přibývat nové uzly, např. v rámci inovace nebo zvýšení hustoty sítě. V podstatě každý směrovací protokol by měl obsahovat mechanizmus na řešení takových situací.

(Ne)podporuje mobilní uzly - podpora mobility spočívá ve schopnosti rychle reagovat na změny vzájemných poloh. Předpokládá se, že se všechny uzly nepohybují stejným směrem a stejnou rychlostí. Bližší rozbor bude v kapitole 5.

(Ne)řeší energetickou úsporu - hlavní téma tohoto článku a jeden z nejdůležitějších problémů WSN. Jsou-li uzly napájené z omezených zdrojů a umístěné např. v těžko dostupném prostředí, je požadavkem dlouhodobý provoz bez nutnosti zásahu obsluhy. Jiným aspektem může být cena za údržbu, kde je také snahou minimalizovat četnost zásahů.

(Ne)podporuje QoS - QoS (quality of service, kvalita služby) řeší dva úkoly - čas doručení a rozložení objemu dat. Tyto úkoly mohou být protichůdné. Budou-li např. k dispozici dvě cesty k cíli a budou-li se zprávy posílat vždy jen tou kratší, může dojít k zahlcení této cesty a k zahazování zpráv. Využijí-li se obě cesty, sníží se riziko zahlcení a zahazování.

Odolnost proti útokům - na směrování jsou v zásadě zaměřeny dva hlavní typy útoků - DoS (denial of service, odepření služby) a vyčerpání. Útokům založeným na rušení komunikačního kanálu nebo zahlcením nadměrným množstvím zpráv se dá jen těžko algoritmicky bránit. V ostatních případech se lze bránit detekcí, adaptací, odstraněním příčiny nebo redundancí [3].

Synchronní/asynchronní - energetických úspor lze docílit jak při vysílání, tak i při příjmu. Aby mohly uzly komunikovat, musí být probuzeny, jeden musí být v danou chvíli připravený vysílat a druhý přijímat. Jde tedy o to, jak často a kdy se mají probouzet. V případě synchronní komunikace dochází k synchronizaci vnitřních hodin a probouzení probíhá podle předem domluveného plánu. Je-li komunikace asynchronní, plánuje se probouzení bez znalosti stavu okolí, ale tak, aby se zvýšila pravděpodobnost, že během bdělosti bude k dispozici potřebný soused.

Tato klasifikace není vyčerpávající, ale postačuje pro základní orientaci ve vlastnostech směrovacích protokolů.

3 Vymezení modelu sítě

Problémů k řešení je u směrování ve WSN mnoho, není možné je zcela obsáhnout a je tedy potřeba vymezit oblast, kterou se budeme zabývat:

- **Malé množství pevných bázových stanic** - zprávy budou z uzlů vždy posílány jen do bázových stanic, které nebudou měnit svou polohu a poměr jejich počtu vůči počtu všech uzlů bude velmi malý.

- **Neřešit linkovou vrstvu** - předpokladem je nerušená komunikace a permanentní bdělost - tento předpoklad je protichůdný k požadavku na úsporu energie, ale protože rozsah problematiky uspávání a probouzení je velký, nebudeme se tím nyní zabývat, ale budeme s tím do budoucna počítat.

- **Absence útočníků** - podobně jako u předchozího bodu je rozsah problematiky útoků velký, bude se zatím předpokládat, že nikdo zvenku nebude zasahovat do chodu směrování.

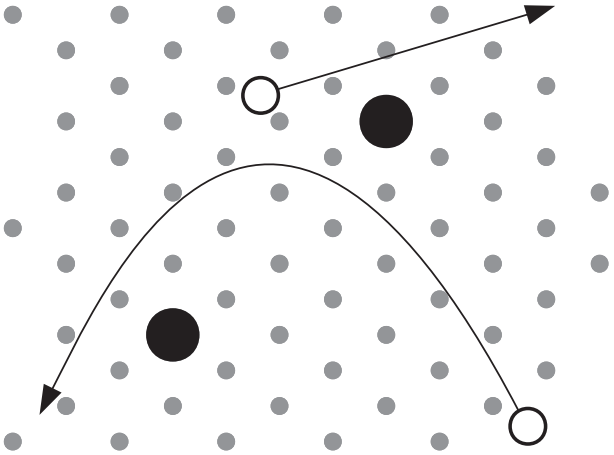
3.1 Postup výzkumu

Nemá smysl zkoumat hned z počátku všechny možné situace, ale je vhodné je rozdělit do postupných částí:

1. **Mnoho rovnoměrně rozložených statických uzlů** - v počátku výzkumu budeme zkoumat chování existujících a nově navržených směrovacích protokolů ve velkých sítích, které budou mít pravidelně rozložené uzly, tedy každý bude mít v průměru stejný počet sousedů a linky mezi nimi budou mít stejnou propustnost a cenu. Taková síť by mohla být trojúhelníková, čtvercová nebo hexagonální. Naměřené hodnoty budou sloužit jako etalon pro porovnání s „horšími“ konfiguracemi. Postupně budeme síť degradovat a sledovat, jak se bude v daných situacích měnit chování směrování.

2. **Malé množství mobilních uzlů s predikovatelným pohybem** - ve velké statické síti se bude pohybovat několik mobilních uzlů, jejichž pohyb bude takový, aby se dalo s danou pravděpodobností určit jejich polohu po uplynutí jednotky času. Postupem času budeme ve výzkumu navyšovat poměr mobilních uzlů vůči statickým.

Názorná ukázka, jak by mohl takový model vypadat je na obrázku 1. Černé kroužky jsou bázové stanice a bílé jsou mobilní uzly. Šedé jsou pak zbytek statické sítě.



Obrázek 1: Příklad vymezeného modelu sítě

4 Cíl optimalizace

Cílem optimalizace úspory energie je maximalizovat sumu energie v celé síti a minimalizovat odchylky od průměru. Tedy prodloužit životnost sítě jako celku. Nemělo by se stát, že část sítě zbytečně odumře předčasně vyčerpáním, protože přes ní bylo posíláno nadměrné množství zpráv, i když bylo možné tok lépe rozložit. Jde o správné vyvážení úspory energie s rychlostí doručování.

5 Mobilní uzly

Na mobilní uzly lze pohlížet dvěma způsoby - uzel je zdroj dat, které potřebuje doručit z aktuálního místa v síti, nebo funguje jako kurýr, který posbírá data z aktuálních sousedů a převeze je do jiné části sítě, čímž ušetří energii mezilehlých uzlů.

Žádný z nejčastěji citovaných směrovacích protokolů, které jsou označeny pro použití v mobilních bezdrátových sítích [4], neřeší mobilitu. Reaktivní protokoly vždy před vysláním zprávy zjišťují cestu, takže nepotřebují předem znát, kde se probudí a komu pak data poslat. U proaktivních protokolů se zase předpokládá, že změny a pohyby v síti budou natolik pomalé, že budou stačit periodické aktualizace směrovacích informací.

Mějme tři scénáře k výzkumu vycházející z praktických požadavků: sledování vybraných jedinců zvěře v přírodě, logistické řízení pohybu kontejnerů v námořním přístavu a sledování a řízení dopravy ve městě. Každý scénář je jiný co do struktury sítě, rychlosti pohybu objektů a hlavně počtu mobilních objektů. Jak již bylo zmíněno, nelze vytvořit univerzální protokol na všechny situace, ale postupným vývojem při přechodech mezi definovanými scénáři lze obsáhnout široké spektrum.

6 State of the art

I přes vymezení modelu sítě stále zůstává oblast výzkumu velmi široká. Omezme tedy řešení problémů jen na několik principů, které lze dále vylepšovat a kombinovat.

6.1 Optimalizace mravenčí kolonií

Úkolem optimalizace mravenčí kolonií (Ant Colony Optimization, ACO) [5] je nalezení vhodných cest. ACO je pravděpodobnostní metoda. Je inspirována chováním mravenců při hledání potravy. Když vyrazí mravenec pro potravu a nalezne ji kratší/výhodnější cestou, cestou zpět tuto označí feromonem. Další

mravenec bude při hledání označenou cestu preferovat, ale nemusí se jí držet. Může hledat jiné a nalezené lepší řešení opět označí feromonem. Tímto způsobem lze konvergovat k optimálnímu řešení, neboť čím bude cesta výhodnější, bude feromonová stopa silnější. Díky pravděpodobnostnímu přístupu se nemůže stát, že optimalizace dokonverguje k lokálnímu optimu.

V této podobě však bude metoda velmi pomalu reagovat na změny v síti. Zavádí se tedy ještě princip vyprchávání feromonů v čase. Čím silnější bude stopa, tím rychleji bude vyprchávat. Bude-li cesta stále nejvhodnější, bude po ní stále chodit hodně mravenců a budou jí obnovovat. Nalezne-li se vhodnější řešení, provoz se tím rychleji přesune.

6.2 Zónové směrování

Zónové směrování je rodina protokolů, které využívají kombinaci proaktivního a reaktivního směrování. Jsou dva možné přístupy.

První je, že se sít rozdělí do jednotlivých zón, které se mohou i nemusí překrývat. Tento přístup se také nazývá shlukováni (clustering). V rámci zóny se zvolí hlava (head) zóny, jehož úkolem je proaktivně spočítat směrování uvnitř zóny. Tyto informace pak rozešle ostatním. Při přechodech zpráv mezi zónami se pak uplatňuje reaktivní směrování. Úkolem takového přístupu je v podstatě rozdělení velkého výpočetního problému na menší, čímž se řeší škálovatelnost směrování.

Podobný přístup se používá u hierarchického shlukového směrování. Rozdíl je ale v tom, že se předpokládá, že hlava shluku bude mít dostatečný vysílací výkon na to, aby dosáhl přímo na bázovou stanici nebo na další uzel v hierarchii. Nelze to tedy použít v sítích s rovnocennými uzly. Příkladem je protokol LEACH [6].

Druhý přístup je takový, kdy si zónu okolo sebe do určitého počtu přeskoků tvoří každý uzel. Příkladem je protokol ZRP [7]. Odpadá nutnost volby hlavy zóny, ale zvyšuje se výpočetní náročnost všech uzlů. Tento přístup lze chápat jako kešování reaktivních metod.

6.3 Plošné metriky

Většina směrovacích protokolů, které berou v potaz úsporu energie, počítá při výpočtu cesty jen s aktuálním stavem energie v jednotlivých uzlech, cenou spojů a případně vytížeností jednotlivých uzlů. Tento přístup může vést k pomalejšímu rozkládání toku v síti a předčasnému vyčerpávání některých uzlů. Zavede-li se pomocná metrika, která se bude počítat plošně a určí vhodné/nevhodné oblasti místo jednotlivých uzlů, pak by se mohl tok lépe a rychleji rozložit. Určovala by plošnou míru zatížení.

Jedním z takových algoritmů je PageRank [8]. V současnosti jsou na PageRanku založené protokoly PR-RAM [9] a VOL-RAM [10]. Přistupují k problému ale trochu jinak. Při inicializaci sítě se nalezou všechny nejkratší cesty, bráno počtem přeskoků, ze všech uzlů do bázových stanic. Ve vytvořeném orientovaném grafu, kde bázové stanice tvoří stoky, se listům grafu přiřadí hodnoty 1 a všem ostatním se přiřadí již podle výpočtu PageRanku. Výsledné hodnoty určují pravděpodobnostní zatížení uzlů. Čím vyšší hodnota, tím vyšší pravděpodobnost, že bude uzel sloužit jako mezilehlý článek pro přenos zprávy. Bude-li se uzel rozhodovat, kam zprávu poslat, měl by preferovat souseda s nižším PageRankem, kde bude nižší pravděpodobnost přetížení.

U velkých a/nebo proměnlivých sítí by bylo počáteční sestavování grafu náročné. Vycházelo by se tedy z jiné představy a to takové, kdy by se PageRank počítal v neorientovaném grafu bez bázových stanic. PageRank lze počítat iterativně, tedy decentralizovaně, a lze do něj zakomponovat váhy uzlů. Váhy by se přiřazovaly podle jejich stavu energií a datového vytížení. Musely by se však řešit problémy jako např. postupná divergence hodnot. Není také jisté, zda by výpočet vedl k očekávanému výsledku. To je předmětem dalšího výzkumu.

7 Závěr

Jak bylo naznačeno v sekci 5 je v oblasti výzkumu směrování mobilních WSN velký prostor pro zakomponování predikce pohybu. Další výzvou je komprese informace o cestě v případě mravenčí optimalizace. V kombinaci s rozdělením sítě do zón nemusí mravenci cestovat příliš daleko a přidáním plošných metrik by se mohly urychlit konvergence k vhodnějším cestám.

Poděkování

Tato práce byla podpořena grantem ZČU SGS-2013-029 Pokročilé výpočetní a informační systémy.

Reference

[1] T. Wan, E. Kranakis, and P. Van Oorschot. Securing the Destination Sequenced Distance Vector Routing Protocol (S-DSDV). in 6th International Conference on Information and Communications Security, 2004, pp. 27-29

[2] Charles E. Perkins and Elizabeth M. Royer, Ad- hoc On-Demand Distance Vector Routing. Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, 1999, pp. 90-100

[3] Chen, X., Makki, K., Yen, K., Pissinou,N.: Sensor network security: A survey, IEEE Communications Surveys and Tutorials, 2009, Vol. 1, pp. 52–73

[4] Bakht, H.: Survey of Routing Protocols for Mobile Ad-Hoc Network, International Journal of Information and Communication Technology Research, 2011, Vol. 1, pp. 258–270, ISSN-2223-4985

[5] Kannan, S., Kalaikumaran, T., Karthik, S., Arunachalam, V. P.: Ant colony optimization for routing in mobile ad-hoc networks, International Journal of Soft Computing, 2010, Vol. 5, pp. 223–228

[6] W. B. Heinzelman, A. P. Chandrakasan, H. Balakrishnan: An Application-Specific Protocol Architecture for Wireless Microsensor Networks, IEEE Transactions on Wireless Communications, 2002, Vol. 1, pp. 660–670

[7] Samar, P., Pearlman, M. R., Haas,Z. J.: Independent zone routing: An adaptive hybrid routing framework for ad hoc wireless networks, IEEE/ACM Transactions on Networking, 2004, Vol. 12, pp. 595–608

[8] Brin, S.: The anatomy of a large-scale hypertextual Web search engine 1, Computer Networks, 1998, Vol. 30, pp. 107–117

[9] Yoon, S., Ko, D., Koh, S., Nam, H., An, S.: PR-RAM: The Page Rank Routing Algorithm Method in Ad-hoc Wireless Networks, 2011 IEEE Consumer Communications and Networking Conference, CCNC'2011, 2011, pp. 96–100

[10] Kumar, G., Mishra, N. ,Singh, A. P., Kushwaha, O. P.: A novel (VOL-Routing) Page Rank based on Visit of Links Routing algorithm method in ad-hoc wireless networks, Proceedings of the 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques, ICICT 2014, 2014, pp. 435–438

BLOCK CIPHERS' RESISTANCE TO LINEAR AND DIFFERENTIAL CRYPTANALYSIS

Josef Kokeš
Informatics, 1st class, full-time study
Supervisor: Róbert Lórencz

Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9, 16000 Prague 6, Czech Republic

josef.kokes@fit.cvut.cz

Abstract. We discuss the current results of cryptanalysis of the AES, and propose an alternative technique for overcoming the computational problems related to them, which is building a reduced-size model of the cipher and applying the cryptanalysis to that, while gradually increasing the size to get an estimate for the level of scaling of particular cryptographic attacks. Our current results suggest that this is a promising idea, with a potential for further understanding of the conditional security of the cipher. We also present several research directions using this technique, and our dissertation goals.

Keywords. Advanced Encryption Standard, AES, Cryptanalysis, Encryption, Rijndael, Security.

1 Introduction

We live in the age of information. The volume of information produced by mankind grows exponentially [11], which presents us with a number of challenges, including simply keeping up to date with current information. One of the most important challenges is information security: We need to be able to maintain integrity, availability and confidentiality of information. The recent Snowden revelations about the widespread collection and manipulation of private information by NSA and other information agencies brought this topic into the public's eyes.

Mankind has a powerful tool for helping achieve confidentiality, and that tool is encryption. There are many ciphers currently used all around the world, on many different levels: encryption is not limited to governmental officials or secret agents, even ordinary people frequently come into contact with ciphers[1] when reading e-mail, using online banking, identifying themselves with a chip card and any number of other situations. For this reason it is imperative that we know that our ciphers are secure.

In our dissertation, we focus on one particular problem: How well are the current symmetric block ciphers able to resist the known techniques of linear and differential cryptanalysis. We are specifically interested in the resistance of AES[2] as probably the most widely used symmetric cipher in the world.

We would like to independently verify the security of AES as related to the linear and differential cryptanalysis. As a secondary target, we hope that our research will reveal new information on the inner

[1] Though they may not realize it.
[2] Advanced Encryption Standard.

working of both cryptanalyses, with the hope of combining their strengths for a synergistic effect on their power. Eventually, we would like to add the techniques of algebraic cryptanalysis into the mix and extend the focus to other block ciphers, further enhancing our ability to assay the conditional security of a given cipher.

Note that we are intentionally limiting ourselves to the cryptanalysis of the *algorithm itself*, abstaining from attacks against the *implementation* of the algorithm such as various side-channel attacks, including timing attacks or fault introduction, or attacks against the *user* of the algorithm such as implementing and using keyloggers or tools for searching computer memory for stored keys.

2 The problem

AES is a widely used cipher, selected in 2000 in a NIST[3]-initiated open contest from among 15 candidate ciphers. The proposed goal was to create the best symmetric cipher for the new century, and to this end all candidates underwent a strenuous process of evaluation by not only both NIST and the creators of competing ciphers, but by general public as well. As a result, all the finalists are considered strong ciphers who resisted all attacks known at the time and provided a sufficient security margin for the future.

Despite that, a number of attacks have been developed since the Rijndael cipher was selected as AES. Some promising but not yet realized results have been derived from the relatively simple algebraic description of the cipher, which may be exploited because its security depends on as-yet unproved hypotheses (see [9] and [15]. Extensions of the earlier Square attacks[4] were shown to apply, to a certain degree, to Rijndael as well ([7], [8], [10]). Many authors experimented against reduced versions of AES, i.e. AES with a reduced number of rounds (e.g. only 6 or 7 rounds as compared to 10 rounds of AES-128), and indeed described some successful attacks in these conditions ([10], [4] and others).

Despite the fact that both linear and differential cryptanalyses were known at the time of the AES selection process and that all AES candidates underwent extensive testing under these techniques, today's most successful attacks against AES are indeed extensions of the differential cryptanalysis:

- A related-key attack was proposed in 2009 with a complexity of 2^{119} for the 256-bit version of the full cipher, shortly improved to complexity of $2^{99.5}$ [5]. While this is a significant improvement of the known attacks, the requirements on the related keys would make it impractical even if we had computers fast enough to handle the attack's complexity.

- Another attack was proposed in 2011 [6], one which works against full (non-truncated) AES and allows key recovery without placing specific restraints on the key. Unfortunately, the complexity of the attack is prohibitive, as the attack is only approximately four times faster than the brute force (e.g. $2^{126.1}$ for AES-128).

This suggests that despite the fact that modern ciphers were designed with linear and differential cryptanalysis in mind, and their authors attempted to make the ciphers invulnerable to these attacks, it may actually be possible to achieve success with these techniques, if only we can apply them creatively enough.

The prohibitive complexity of known attacks is one of the significant issues with cryptanalyzing AES. The cipher was designed to provide security for a foreseeable future, which enforced design choices which would prevent all attacks known at the time by sheer size if not by actual resistance to them. Unfortunately, this also makes a proper cryptanalysis difficult: while new attacks can be proposed and theoretically verified, we cannot execute – and verify – them in practice.

[3] National Institute of Standards and Technology.

[4] Rijndael's design was based on an older cipher Square, designed by the same authors.

3 Our approach

We attempt to overcome these challenges by first analysing a significantly reduced model of a given cipher, which would, however, reflect the properties of the full cipher. This way we can quickly evaluate, by implementing a *practical demonstration*, whether a proposed attack is worth further study. The idea is, if an attack isn't practical even against a reduced model, then it likely won't be able to succeed against the full cipher, either. On the other hand, an attack successful against the model may possibly scale to the full cipher well enough to be practicable.

This approach needs to deal with several challenges, though:

3.1 Designing the model

We must be able to design a suitable model for a particular cipher. Fortunately, this is easy with AES, due to the way the original Rijndael (of which AES is a formalized variant) was designed: An important aspect of the design was the desire of Daemen and Rijmen to prevent any possible suspicion of hidden backdoors in the cipher[8]. To this end they abstained from using "magic constants" in their design, opting instead for defining a set of rules which need to be satisfied and then arbitrarily selecting any one of the implementations which would satisfy the rules, with an implied suggestion that if anyone finds a particular choice suspicious, he or she can easily select another.

It follows that if we could select a different set of primitives in such a way that the design choices and set rules were respected, we would get a cipher which should behave in a similar way to Rijndael (and thus AES). We could, for example, reduce the cipher's state matrix to smaller dimensions while keeping all other primitives unchanged, generating a cipher equivalent to Rijndael, only reduced to a state of e.g. 144 or 32 bits (with a 3×3 or 2×2 state matrix, respectively).

This idea was used by Cliff Bergman of Iowa State University to design a Baby Rijndael cipher [1], and indeed the cipher proved quite useful for cryptanalysis [16]. We expanded on this idea in our diploma thesis [12] and our dissertation aims to expand on that.

3.2 Designing and applying attacks

In the first phase, our research is simplified by the fact that there have been numerous theoretical attacks on Rijndael suggested, so the design was already done. It remains for us to adapt the proposed techniques to a particular model (Baby Rijndael, at the moment) and write a program which would verify whether that attack was successful or unsuccessful. This way we can quickly sift through available attacks, selecting only those with promising results.

The second phase is much more difficult: We will need to design new attacks, by combining known attacks or adding new ideas to them, or possibly design completely new attacks. We would particularly like to attempt to find ways in which different kinds of cryptanalysis (linear, differential or algebraic) could "share information" with each other in such a way as to amplify the results. It is unclear as yet whether such sharing of information is even possible, but we hope that at least a limited co-operation of the cryptanalytic techniques will be discovered.

3.3 Extending the attacks to the full cipher

If a promising new attack should be found, it will be crucial to verify how it behaves if we change the model.

A particularly important information is the way the attack scales when we enlarge our model, which would help us evaluate the effect of the attack against the full cipher. For example, in our earlier work [12] we discovered that linear cryptanalysis of Baby Rijndael indeed can achieve some success; however,

this may have been caused by the reduced size of elements of the cipher's state matrix developing a "false linearity" in the substitution function, which may disappear if we increase the size of the elements.

It should be noted, however, that even if a proposed attack does not scale to the full cipher, it can still give us important information on the conditional security of the cipher: it identifies which particular component of the cipher's design is most susceptible (or most resistant) to the attack, and it enables us to estimate the security margin of the cipher as related to this attack.

4 Current results

First of all, we expanded upon our research of the properties of Baby Rijndael cipher in [12] in order to precisely show that the cipher is indeed a suitable model of AES, as suggested in section 3.1. We fixed several omissions and inaccuracies and now believe the result now proves the properties we need for our research. The article detailing the results [13] is now undergoing a review process at Information Processing Letters.

In the course of writing the diploma thesis [12], where we applied four different linear approximations to the Baby Rijndael, we discovered several interesting properties of the cipher, which we consider the cornerstones of our current analyses:

4.1 Correlation of the value of master key and the success rate of the linear cryptanalysis

The three-round version of Baby Rijndael exhibits a significant correlation between the value of the *master key* of the cipher[5] and the ability of our linear approximation to discover the correct *last round key*. This is very disturbing, as there should be no such correlation in a properly designed cipher, so its apparent presence here may signify a fatal flaw in the cipher.

We are currently attempting to discover what causes this correlation. Unfortunately, we are getting significantly hampered by the fact that the four-round version of the cipher does not exhibit this behavior, at least not to the naked eye. We expect that a correlation occurs even in the four-round version of the cipher, but we are finding it difficult to devise a proper metrics which would enable us to measure the size of the correlation. Devising one is the most important task for the near future.

4.2 Correlation of the number of active bits in linear approximation and the success rate of cryptanalysis

We observed that the average success rate of linear cryptanalysis depends, all other conditions being equal, on the number of active bits of the used linear approximation[6]. We are not aware of any existing theoretical explanation of this phenomena, and are trying to establish both the validity of the observed behavior and the theoretical reasons for it. A success here would expand the knowledge of the workings of linear cryptanalysis significantly, because as of now only the probability bias of the linear approximation is considered the key factor for the success rate of a linear cryptanalysis.

4.3 Implementation aspects

The analysis of the prior two phenomena was made difficult by the implementation aspects of the original programs designed for [12]; particularly, the long calculation time was preventing us from comprehensively testing ideas, as a full calculation of one test could take as much as several days. This has been

[5] The key provided by the user to the encryption algorithm. Round keys required for the function of the cipher are derived from the master key by a process called key schedule.

[6] A bit of plaintext or ciphertext is active if it appears in the linear equation of our approximation.

solved by a complete rewrite of the implementation, which resulted in more than hundred-fold increase of speed.

5 Future work

Aside from completing the research of the two problems described above, we have several research plans for our dissertation. We list them here in the order in which we would like to approach them:

- The classical linear cryptanalysis attempts to recover key with a granularity of a full S-box[7]. As this approach exhibits significant limitations of the success rate, we will research the possibility of recovering the key with a smaller granularity, e.g. recover only three bits of a 4-bit S-box, but with a higher probability of success.

- So far we have been only using "algorithm 2" of linear cryptanalysis, suggested by Matsui in [14]. Baby Rijndael, however, may be particularly vulnerable to Matsui's "algorithm 1", as the cipher's S-box construction allows for creating multiple linear approximations with high probability bias. We may be able to use this fact to construct a system of linear equations which would be able to recover more key bits than "algorithm 2".

- All tasks above need to be applied not only to Baby Rijndael, but to other models of AES as well. Specifically, we need to establish how the three critical variables of a model – the number of rounds, the size of the state matrix, and the size of one state matrix element – influence the effect of a particular approach. With sufficient data, we can then extend our results to the full AES, whose cryptanalysis would be computationally infeasible.

- Find some way to combine linear, differential and, eventually, algebraic cryptanalysis into a complex system. The main idea is that each of these cryptanalyses attempt to reach the same goal, recovering the key, but using different approaches. If we could devise a method in which all these techniques could co-operate and tranfer information between themselves, we hope that we could recover the key with a lower complexity than each technique can achieve on its own.

6 Conclusion

Evaluating the conditional security of AES, a modern widely used symmetric block cipher, as related to the techniques of linear and differential cryptanalysis, is a huge project, with many complicating factors along the way. But we believe we are off to a good start: We have determined a method which overcomes the computational infeasibility of traditional approaches, we have solidified its prerequisites, and we have a working and efficient implementation of the method. It remains to be seen which results can we achieve: The cipher may indeed prove to be resistant to our attacks, which in itself would be an important result, as it would add to the trustworthiness of the cipher. But we hope some of the promising leads we have will result in a successful cryptanalytic attack on AES, which would at the same time provide new insights into the security of symmetric block ciphers, and force a development of new, even stronger ciphers. The potential for improvement in the current cryptanalytic techniques is not to be overlooked, either. We believe any one of these results would be highly dissertable, as they would add to the understanding of information security, a highly relevant topic in today's world.

[7] That is, if a cipher uses 4-bit S-box, then the linear cryptanalysis is expected to find 4, 8, 12 etc. bits of the key.

References

[1] Bergman, C.: A Description of Baby Rijndael. Iowa State University, 2005.

[2] Biham, E., Shamir, E.: Differential Cryptanalysis of DES-like Cryptosystems. Lecture Notes in Computer Science Volume 537, 1991, pp 2-21.

[3] Biham, E., Shamir, E.: Differential Cryptanalysis of the Full 16-round DES. Lecture Notes in Computer Science Volume 740, 1993, pp 487-496.

[4] Biryukov, A., Dunkelman, O., Keller, N., Khovratovich, D., Shamir, A.: Key Recovery Attacks of Practical Complexity on AES Variants With Up To 10 Rounds. Cryptology ePrint Archive, Report 2009/374, 2009.

[5] Biryukov, A., Khovratovich, D.: Related-key Cryptanalysis of the Full AES-192 and AES-256. Lecture Notes in Computer Science Volume 5912, 2009, pp 1-18.

[6] Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique Cryptanalysis of the Full AES. Advances in Cryptology – ASIACRYPT 2011.

[7] Daemen, J., Rijmen, V.: AES proposal: Rijndael, in AES Round 1 Technical Evaluation CD-1: Documentation. NIST, August 1998.

[8] Daemen, J., Rijmen, V.: The design of Rijndael: AES – the Advanced Encryption Standard. Springer-Verlag, 2002, ISBN 3-540-42580-2.

[9] Ferguson, N., Schroeppel, R., Whiting, D.: A simple algebraic representation of Rijndael. Lecture Notes in Computer Science Volume 2259, 2001, pp 103-111.

[10] Ferguson, N., Schneier, B., et all: Improved Cryptanalysis of Rijndael. Lecture Notes in Computer Science Volume 1978, 2001, pp 213-230.

[11] Gantz, J., Reinsel, D.: Extracting Value From Chaos. IDC, 2011.

[12] Kokeš, J.: Cryptanalysis of Baby Rijndael. Diploma thesis, Faculty of Information Technology, Czech Technical University in Prague, 2013.

[13] Kokeš, J., Lórencz, R.: Baby Rijndael as a Reduced-size Model of AES/Rijndael. 2014. Not yet published (pending review).

[14] Matsui, M.: Linear Cryptanalysis Method for DES Cipher. Lecture Notes in Computer Science 765, 1994, ISBN 978-3-540-57600-6, pp 386-397.

[15] Nover, H.: Algebraic Cryptanalysis of AES: An Overview. University of Wisconsin, 2004.

[16] Wrolstad, J.: A differential cryptanalysis of Baby Rijndael. Iowa State University, 2009.

Universal Generation of Test Vectors for Functional Verification

Ondřej Čekan
Computer Science and Engineering, 1st class, full-time study
Supervisor: Zdeněk Kotásek

Faculty of Information Technology, Brno University of Technology
Božetěchova 2, Brno 612 66

icekan@fit.vutbr.cz

Abstract. The goal of this paper is to sumarize information about test vector generation for functional verification. Test vector generation is based on problem of solving constraints which is equivalent to Constraint Satisfaction Problem. The problem consists of finding a solution (assignments for variables) that must satisfy certain constraints. In the paper, the principles of functional verification and Coverage Directed Test Generation as one of the latest techniques for functional verification are also described. In the final part of the paper we propose a solution of universal generation of test vectors based on solving the constraints.

Keywords. Test vector generation, Functional verification, Constraint solver, Constraint Satisfaction Problem, Coverage Directed Test Generation.

1 Introduction

These days, more and more emphasis is given to the testing of the accuracy of the circuit's behavior. Today's integrated circuits are very large and complex, so the earlier techniques used for testing the correctness of hardware are not sufficient. Number of new techniques and tools that are intended to detect errors in the circuit are being developed. Errors can be caused by faults in the design or manufacture. In the foreground is a notion of functional verification which is used by large companies such as IBM, Cadence, Synopsys or Mentor Graphics [2].

Functional verification [7] is very usefull and important means of circuit's verification. It helps to verify the correctness of the system according to the specification of the system. For a thorough verification of the system, a huge number of input test vectors is needed, although it is not possible to check all combinations in a reasonable time. Functional verification is focused on verifying of selected key functions of the system by using several random tests. The key functions are a set of properties based on the system specification. In the case that some functions are not checked, process of verification is directed to generation such tests that cover these functions. This significantly reduces the Cartesian product of possible inputs. Overall, functional verification reduces the time for thorough testing of the system. The basis of functional verification is a reference model [9] which performs the function according to the specification and its output is then compared with the tested circuit. An important element described in this article is a generator of test vectors that generates inputs for the verified circuit. These inputs must comply with certain constraints. The outputs of the generator are essential to thoroughly test the circuit. It is profitable to generate test vectors automatically and accurately. Described principle of functional verification shows Figure 1. The main principles of such generator are described in Section 2. Section 3 focuses on the *Constraint Satisfaction Problem* (CSP) whose purpose is to find values of variables that satisfy some restrictive conditions. Section 4 shows several clues how to solve the CSP. It also describes constraint solving and typical algorithms. Section 5 proposes our solution for generating test vectors and section 6 contains some concluding remarks.

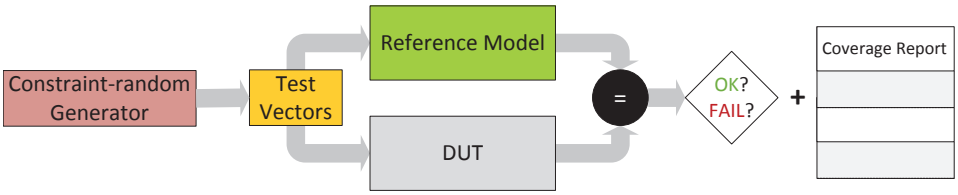


Figure 1: The principle of functional verification.

2 Coverage Directed Test Generation

Coverage Directed Test Generation (CDTG) [1] [8] is one of the latest techniques for the verification of large designs. This method generates test vectors according to the defined conditions and limitations which are called *constraints*. The main challenge for generating test vectors is to achieve maximal coverage of circuit functions. As some features of the circuit may still remain unverified, it is necessary to specify additional constraints. Therefore, the CDTG guide us to create these constraints from the coverage analysis in order to achieve as largest coverage as possible. Thus, also the uncovered portion of the circuit can be verified as is shown in Figure 2. Coverage report may be obtained through ModelSim [11] environment. Coverage report contains information about coverage of key functions of the system.

Although various CDTG techniques are used in different technologies developed by different groups independently, they contain two common parts: Constraint model/language and Constraint solver. To describe the restrictive conditions, we can use a constraint model. To find the solution or solutions for these constraints, we can use constraint solver engine.

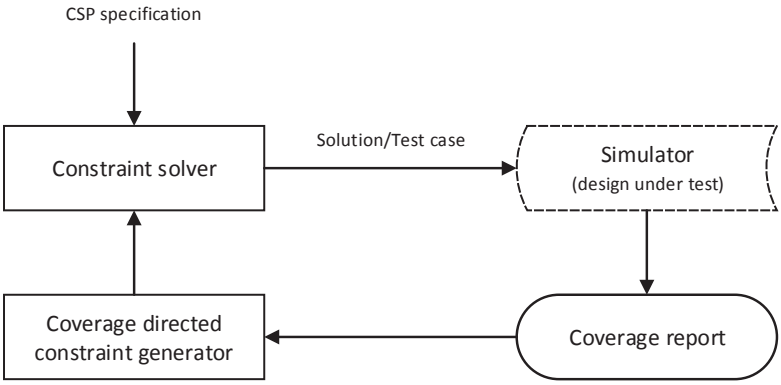


Figure 2: Coverage directed constraint random test generation.

By introducing CDTG we can gain two significant advantages. There is a possibility that the uncovered scenarios will be covered and a higher level of coverage will be achieved. The second advantage is that certain scenarios will be tested multiple times with different inputs.

Most problems in computer science that must satisfy certain constraints are special cases of the CSP or at least, they can be transformed into it.

3 Constraint Satisfaction Problem

Constraint Satisfaction Problem (CSP) [1] [4] [5] is a general mathematical problem defined as a set of variables that can take values from a finite and discrete domain and a set of constraints. The constraint is defined on a subset of variables and determines values from the domain that a variable can take. The result is a solution of one or all evaluations of variables so that the constraints are satisfied.

Among the typical examples of CSPs are N Queens problem, Map-Coloring problem (these two problems are described in the following text), Car sequencing problem, Magic Square, Social Golfers and more.

The N Queens Problem

The N Queens problem [4] is known from the chess game. On the playing board with dimensions N*x*N it is necessary to place the N chess queens so that diagonally, horizontally and vertically they do not jeopardize each other. The Queen can move in the same row, column or diagonal. The problem of the placement of the queens on the board, that have to fulfill certain restrictions, is the typical example of CSP. Example of this problem is shown in Figure 3.

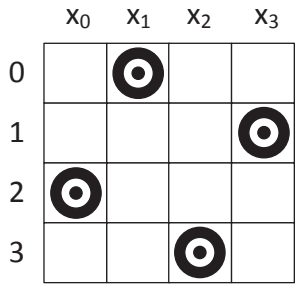


Figure 3: An example of the N Queens problem and solution for $N = 4$.

The Map-Coloring Problem

The Map-Coloring problem [5] can also be solved as a CSP. The problem consists of assigning colors (from a domain) to each region on the map so that two adjacent regions do not have the same color. This problem can be transformed into the constraint graph as shown in Figure 4, which is equivalent to the CSP. Each region of graph represents one variable and their mutual borders represent relationships and constraints between them.

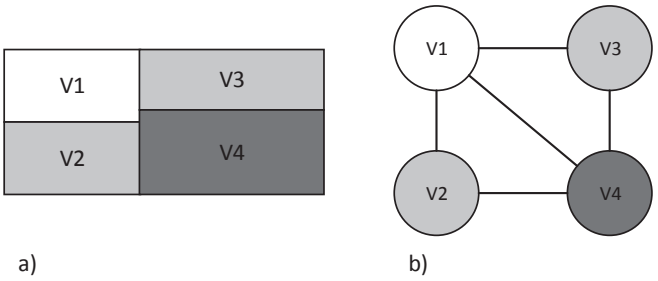


Figure 4: a) An example of the Map-Coloring problem. b) Equivalent constraint graph for the example.

4 Constraint Solver

As stated above, the solution to the CSP is assigning a value to each variable so that all imposed constraints are simultaneously satisfied. This raises the question whether there is a solution to a given CSP? This is the so-called NP-complete [3] [10] decision problem. Therefore, it cannot be conclusively decided in a deterministic polynomial time. As mentioned in the introduction, NP-hard does not hurt because the functional verification does not need all possible cases of input values. An environment for solving the CSP is called Constraint Solver.

A scheme of a constraint solver is shown in Figure 5. It reflects the main principle of how the most solvers work. The first element Pre-process only pre-processes a task of the CSP. The Search element works on the backtracking principle in the conjunction with the constraint propagation. Assigning a value to a variable is statical or is based on a heuristic and then a depth-first search or other searching algorithm can be performed. Backtracking is applied when a conflict in an assignment is detected. The Simplify element contains a queue of constraints and performs their promotion. On the basis of this promotion, values are taken from the domain of variables.

There are several techniques that are used for solving the CSP, hence, several basic types of them are described in the next paragraph.

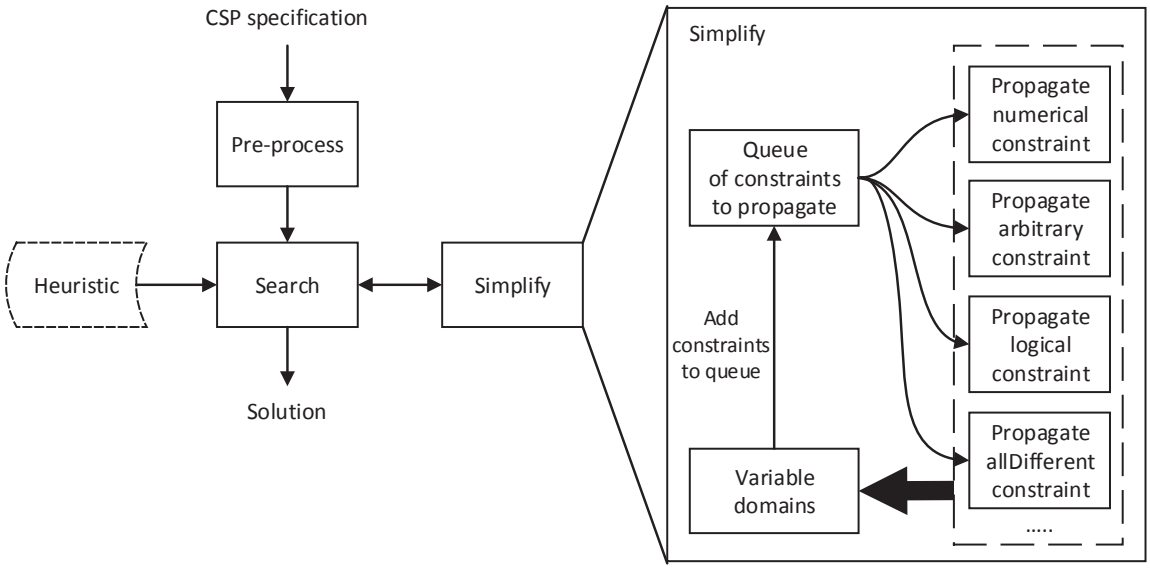


Figure 5: Scheme of a constraint solver.

Generate-and-Test

This method is the simplest possible way to solve the CSP. Generate-and-test [5] method systematically generates all possible combinations of values for the variables and then checks whether all constraints are satisfied. If they are, the solution was found. If not, it generates the next combination. The number of combinations that this solution can generate is equal to the size of the Cartesian product of the variable domains.

Backtracking

The second option is the method called backtracking [5]. This method has been known and used for decades. In contrast to the previous method, backtracking does not assign values to all variables directly but initializes variables sequentially and continuously verifies the validity of the restrictions. If any constraint is violated, assignments of variables are returned to the last valid instance that has another alternative assignment. Backtracking performs a depth-first search. Thanks to backtracking, it is possible to partially eliminate some of the violating passages and reduce the subspace of the Cartesian product.

Although this method is better than the previous one, there is a problem with exponential time complexity for non-trivial problems. Therefore, there are other methods based on backtracking with some extensions and improvements known as intelligent backtracking or systematic backtracking.

Propagating Constraints

Another frequently used method for finding solution is the method based on the Propagating Constraints [5] [6]. The Propagating Constraints method shows another way to solve the CSP. This method is based on two principles. The first principle is the propagation, which aims to reduce the search tree in a way that removes values that do not contribute to the solution. The second principle is to interleave enumeration (also called splitting or branching) that creates a new branch in the search tree. Enumeration always creates two branches, one branch for a valid instance variables ($x = a$) and the other branch for an invalid instance ($x \neq a$). The second branch is used in the case of a constraint violation at the first branch and serves as an alternative way to represent backtracking.

Hybrid Approaches

There are many other techniques [6] that include various combinations of previous approaches and other innovative approaches that belong to the hybrid techniques. For example, a solver based on a genetic algorithm.

5 Test Vector Generation

To prove the correct behaviour of the system according to its specification, testing the system on a wide set of input values is needed. We plan to adjust the generation of input test vectors to functional verification purposes and as an advantageous method seems to be an approach called (CDTG) which we presented in Section 2.

Figure 6 a) shows the proposed method of generating test vectors. It is basic idea of a universal approach that can be used to generate inputs for different kinds of systems. The basic elements of the universality of the generator are two separate pseudo-formal models. The first model labelled as the *Problem Description* contains information about the scenario we want to generate. It may contain information about variables, data types, static values or substitutes that we want to generate. In simple words, this model defines what we want to generate. The second model labelled as the *Constraints for the Problem* describes how the scenario defined in the Problem Description should be generated. This model thus contains constraints that should be taken into account while generating the scenario. This is essentially a limit for data values, such as a variable cannot take certain values from the range of the data type, or restriction of dependency, such as some combination of variables cannot occur after the currently generated combination. Both of these models are inputs to the generator of test vectors that is currently in the implementation phase. The program generates valid input for a specified problem by combining these two models. Typical examples of the use of the generator are processors, functional units, fault-tolerant units, etc. This approach is versatile for both hardware and software test vectors.

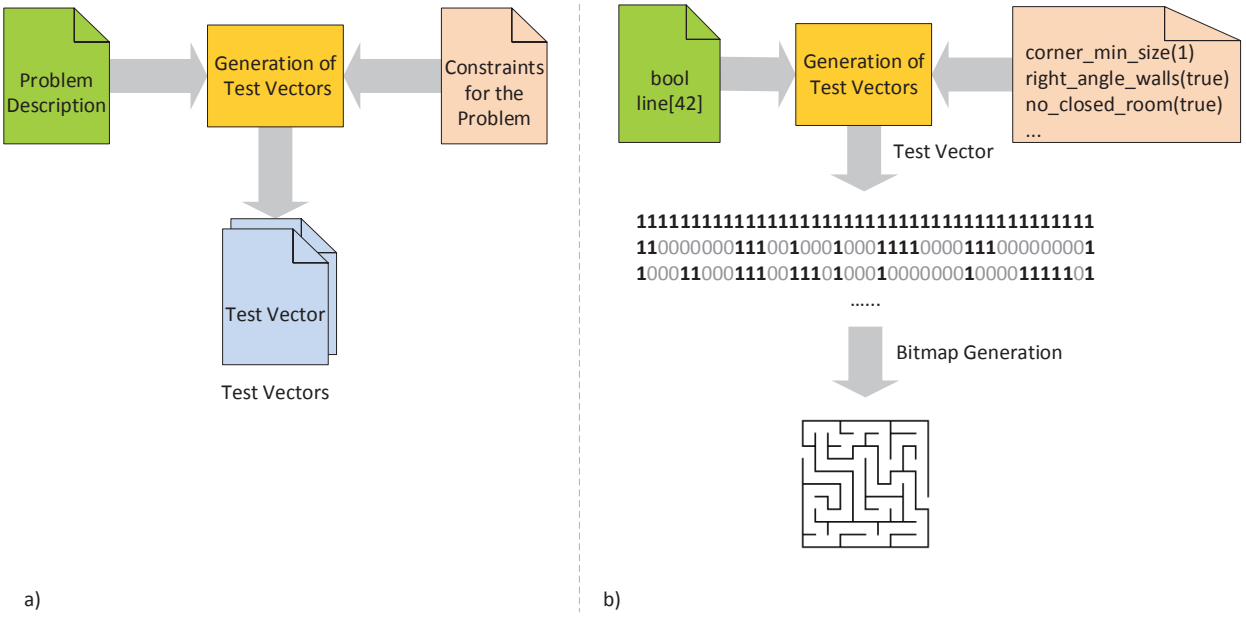


Figure 6: a) The principle of the constraint generator. b) An example of generating a maze for the robot controller.

Figure 6 b) shows an example of generating the mazes for the robot device. Robot device is developed in our department. This is a simple example that shows the use of above mentioned approach. The problem of generating the maze is defined as the generation of lines that are represented by the boolean array of specific size. The constraints restrict the minimal width of the corridor of the maze, the walls of the maze can be only rectangular and a room that has no path cannot appear in the maze. The result obtained by the generator is a sequence of rows that consists of zeroes or ones. Zeroes represent the corridors, ones represent the walls. This generated output may be further processed. In our case, this output is regenerated into a bitmap image representing the desired maze for the robot.

6 The Goals of the PhD Thesis

The topic of this PhD thesis is to study and design techniques for parametrized test vector generation according to the principle of random constraint generation that will be applied in the process of functional verification of various digital circuits (processors, functional units, fault-tolerant units, etc.). Inputs for generator will be obtained from a specially designed blocks. These blocks define the format of generated test vectors and conditions that will be applied in the process of generating these vectors. Outcome of this thesis will be developed methods for generating test vectors.

In future work, we want create test vector generator and generate test vectors for maze of robot controller and for some open source processor. In later work, we would like to generalize the process of generation and design such constraints that will be general and it will be possible to define and generate any test vector by them. The findings will be analyzed and based on them the principles of test vector generation will be defined.

Acknowledgment

This work was supported by the following projects: BUT project FIT-S-14-2297, National COST LD12036, project IT4Innovations Centre of Excellence (ED1. 1.00/02.0070), COST Action project "Manufacturable and Dependable Multicore Architectures at Nanoscale".

References

[1] George, M., Ait Mohamed, O.: Performance analysis of constraint solvers for coverage directed test generation. In: Microelectronics (ICM), 2011 International Conference on, pp. 1–5 (2011). DOI 10.1109/ICM.2011.6177404

[2] Graphics, M.: Verification academy - the most comprehensive resource for verification training (2013). URL www.verificationacademy.com

[3] Jefferson, C., et al.: The Minion Manual, Minion Version 0.8.1 (2009). [online, available at http://minion.sourceforge.net/files/Manual081.pdf; accessed 06-August-2009]

[4] Kotthoff, L.: Constraint Solvers: An Empirical Evaluation of Design Decisions. ArXiv e-prints (2010)

[5] Kumar, V.: Algorithms for constraint satisfaction problems: A survey. AI MAGAZINE **13**(1), 32–44 (1992)

[6] Monfroy, E., Castro, C., Crawford, B.: Using local search for guiding enumeration in constraint solving. In: J. Euzenat, J. Domingue (eds.) Artificial Intelligence: Methodology, Systems, and Applications, *Lecture Notes in Computer Science*, vol. 4183, pp. 56–65. Springer Berlin Heidelberg (2006). DOI 10.1007/11861461_8. URL http://dx.doi.org/10.1007/11861461_8

[7] Yuan, J., Pixley, C., Aziz, A.: Constraint-based verification. Springer, 2006. ISBN 978-0-387-25947-5. DOI 10.1007/0-387-30784-2

[8] Shen, H., Wang, P., Chen, Y., Guo, Q., Zhang, H.: Designing an effective constraint solver in coverage directed test generation. In: Embedded Software and Systems, 2009. ICESS '09. International Conference on, pp. 388–395 (2009). DOI 10.1109/ICESS.2009.39

[9] Tasiran, S., Keutzer, K.: Coverage metrics for functional validation of hardware designs. Design Test of Computers, IEEE **18**(4), 36–45 (2001). DOI 10.1109/54.936247

[10] Andrei A. Bulatov. 2006. A dichotomy theorem for constraint satisfaction problems on a 3-element set. J. ACM, **53**(1), 66–120 (2006). DOI 10.1145/1120582.1120584 URL http://doi.acm.org/10.1145/1120582.1120584

[11] Hatnik, U., Altmann, S.: Using ModelSim, Matlab/Simulink and NS for Simulation of Distributed Systems. PARELEC, pp. 114-119, 2004

VYUŽITÍ DYNAMICKÉ REKONFIGURACE VESTAVĚNÝCH SYSTÉMŮ PRO MONITOROVÁNÍ POČÍTAČOVÝCH SÍTÍ

Jan Viktorin
Výpočetní technika a informatika, 1-th class, full-time study
Školitel: Richard Růžička

Fakulta informačních technologií Vysokého učení technického v Brně
Božetěchova 1/2, 612 66 Brno, Czech Republic

iviktorin@fit.vutbr.cz

Abstrakt. Vestavěné systémy jsou typicky omezeny velikostí, výkonností a spotřebou. Pro zlepšování těchto parametrů lze mj. používat rekonfigurovatelná hradlová pole (FPGA). V současné době se do popředí dostávají FPGA čipy s integrovanými více-jádrovými procesory (zejm. rodiny ARM), které dávají výrazně větší prostor pro optimalizaci aplikací na výkon a velikost při zachování nízké spotřeby. Redukci příkonu je tedy možné provádět dynamicky na základě aktuálního zatížení. Cílem práce je využít dynamiky provozu k redukci příkonu zařízení s využitím částečné dynamické rekonfigurace. Na základě vytížení jednotlivých monitorovacích funkcí a charakteru síťového provozu budou časově kritické operace mapovány do FPGA.

Klíčová slova. FPGA, Partial Dynamic Reconfiguration, ARM, System-on-Chip, HW/SW codesign

1 Úvod

Systémy využivající rekonfigurovatelné obvody FPGA s integrovaným procesorem jsou označovány jako Rekonfigurovatelné Systémy na čipu (Reconfigurable System-on-Chip – RSoC). Obvody tohoto typu jsou na trhu již několik let, např. Virtex 5 s integrovaným procesorem PowerPC, popř. designy používající soft-procesory (Xilinx MicroBlaze, Altera Nios-II). Systémy postavené na těchto obvodech byly v minulosti analyzovány z hlediska návrhu (design flow), avšak dosud není prakticky dostupné žádné univerzální řešení pokrývající všechny tyto systémy, nebo alespoň jejich velkou část. Existují pouze řešení dostupná na míru konkrétním aplikacím. V současné době se navíc do popředí dostávají systémy s více-jádrovými procesory ARM (Xilinx Zynq[1], Altera Cyclone V[2], aj.), které nabízí výrazně vyšší výpočetní výkon (při zachování nízké spotřeby) než zmíněné starší architektury. Potřeba takového systému se proto stává stále aktuálnější, což se odráží i na poptávce komerčních firem.

Velkou výhodou RSoC je právě dynamicky rekonfigurovatelné FPGA. Dostupnost částečné dynamické rekonfigurace umožňuje za běhu systému dynamicky přesouvat výpočty z procesorových jader do logiky FPGA a zpět. Díky tomu lze snižovat prostor, který daná aplikace zabírá na čipu za pomoci časového multiplexu. Systém je díky tomu rekonfigurovatelný jak na úrovni strojového kódu, tak na úrovni hardware. Tento přístup lze přirovnat např. k připojení USB zařízení ke klasickému PC, kde

[1] http://www.xilinx.com/products/silicon-devices/soc/zynq-7000/
[2] http://www.altera.com/devices/fpga/cyclone-v-fpgas/hard-processor-system/cyv-soc-hps.html

operační systém automaticky zařízení detekuje a připraví jej k použití pomocí dostupných ovladačů. Připojené zařízení zvýší spotřebu systému až do odpojení, avšak po dobu svého běhu může akcelerovat výpočty, které by na stávajícím počítači trvaly výrazně déle a spotřebovaly výrazně větší množství energie. Blízká integrace procesorového systému a FPGA umožňuje výrazně snížit komunikační režii, která je pro akceleraci aplikace nezbytná. Nevýhodou rekonfigurace je typicky latence samotného procesu rekonfigurace, se kterou je nutné počítat.

Monitorování počítačových sítí přispívá k funkčnosti sítě pouze nepřímo. Z pohledu příkonu znamená monitorování režii, a proto je žádoucí, aby byla monitorovací zařízení optimalizována na spotřebu. Spotřeba monitorovací sondy se odvíjí od jejího zatížení, tedy je závislá na charakteristice provozu na síti, které se typicky mění v průběhu dne.

Aplikace zajišťující monitorování počítačových sítí se typicky skládají s bloků zajišťující operace jako *vyhledání nejdelšího shodného prefixu adresy* (Longest Prefix Match, LPM), *extrakce polí s hlaviček paketů* (Header Field Extraction, HFE), *hledání vzorů na L7 vrstvě ISO/OSI* (Pattern Matching/L7 Decoder, L7), *klasifikace toků podle definovaných pravidel* – např. na základě pětice (srcip, dstip, srcport, dstport, protocol). Tyto operace je možné provádět softwarově a v případě potřeby hardwarové akcelerace mohou být některé z nich přesunuty do hardware. Některé z operací může být dále výhodné analyzovat hlouběji. Např. klasifikace toků může sestávat s různých datově intenzivních algoritmů vč. LPM, nebo hashovacích funkcí, a tudíž může být výhodnější akcelerovat pouze část dané operace.

2 Související práce

V oblasti souběžného návrhu HW a SW (HW/SW codesign) jsou studovány postupy pro rozdělení úloh mezi software a hardware, plánování úloh v čase za běhu systému. Problém plánování úloh mezi hardware a software je obecně znám jako NP-úplný. [2] Proto se zejm. dynamické plánování (za běhu systému) implementuje heuristikami s aplikačně specifickými optimalizacemi.

Článek se věnuje automatickému přemapování volání funkcí sdílené knihovny do FPGA, a to na základě informací o době běhu a četnostech volání těchto funkcí. Pro každou funkci je tedy definován hardwarový blok, který je možné nahrát do FPGA.

Diessel, O. – ElGindy, H.: On Scheduling Dynamic FPGA Reconfigurations, 1998. Článek se zabývá plánováním dynamické rekonfigurace s využím přesunů hardwarových bloků na čipu. Tím se snižuje fragmentace rekonfigurovatelných oblastí a lze do FPGA přesunout více úloh.

Huang, C. – Hsiung, P.: Software-Controlled Dynamically Swappable Hardware Design in Partially Reconfigurable Systems, 2007. Článek popisuje způsob plánování přerušitelných hardwarových úloh. Autoři definují obálku pro rekonfigurovatelné hardwarové bloky, která zajišťuje dočasné uložení vnitřního stavu úlohy (kontextu).

Rullmann, M. – Merker, R.: A Cost Model for Partial Dynamic Reconfiguration, 2008. V článku je představen teoretický model pro optimalizaci rychlosti částečné dynamické rekonfigurace na základě grafu přechodů mezi možnými konfiguracemi FPGA. Graf přechodů vkonfiguracíyužívu toho, že některé části různých rekonfigurovatelných modulů mohou obsahovat stejné rekonfigurační rámce. Na základě tohoto grafu lze určit nejmenší počet dílčích rekonfigurací, které změní aktuální konfiguraci FPGA do cílové konfigurace.

3 Rekonfigurovatelné Systémy na čipu

Obvody typu RSoC sestávají ze dvou hlavních částí: procesorový systém (processing system) a programovatelná logika (programmable logic, FPGA). Tyto části jsou na sobě buď nezávislé, anebo je některá z nich řídicí, což má vliv zejm. na zavádění systému (boot).

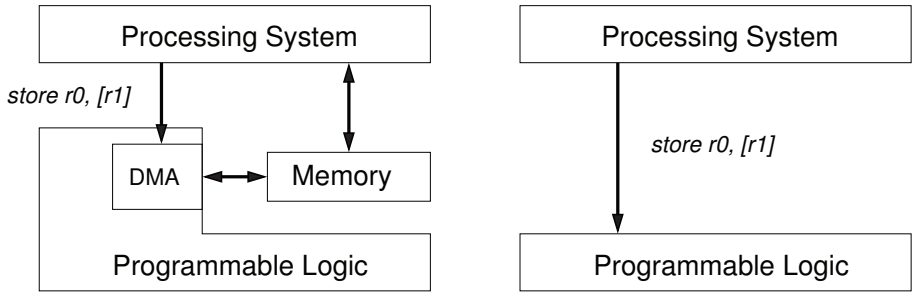
Např. RSoC systém postavený okolo soft-procesoru MicroBlaze má jako řídicí část programovatelnou logiku, protože v ní je samotný procesor realizován. Obvod Xilinx Zynq má jako řídicí část procesorový systém. Zde musí nejdříve bootovat procesor, který inicializuje FPGA. V obvodech Altera Cyclone V lze obě části provozovat nezávisle, popř. volit, který element je řídicí.

3.1 Komunikace v obvodech RSoC

Pro implementaci systému na RSoC potřebuje vývojář znát způsoby propojení mezi oběma částmi. V principu lze nalézt 2 způsoby propojů:

1. Přímé propojení, které je v procesorové části navázáno na instrukce pracující s paměťovým prostorem. V tomto případě je zřejmé, že se procesor výrazně podílí na komunikaci, protože pro každý zápis datového slova musí provést alespoň jednu instrukci modifikující paměťový prostor vybrané jednotky v programovatelné logice (např. `store r0, [r1]`). Výhodou tohoto přístupu je nízká latence, avšak nehodí se pro datově intenzivní přenosy.

2. Propojení přes paměť, kdy procesorový systém nejprve připraví data v paměti, která je dostupná oběma částem systému. Potom nakonfiguruje příslušný DMA řadič tak, aby tato data přenesl do vybrané jednotky v programovatelné logice. Tento způsob je vhodný pro datově intenzivní přenosy a má typicky vyšší latenci než předchozí přístup. Samotná konfigurace DMA řadiče vyžaduje 1 nebo více přímých přístupů do jeho adresového prostoru.



Obrázek 1: Způsoby komunikace v obvodech RSoC (vlevo: přes paměť, vpravo: přímo).

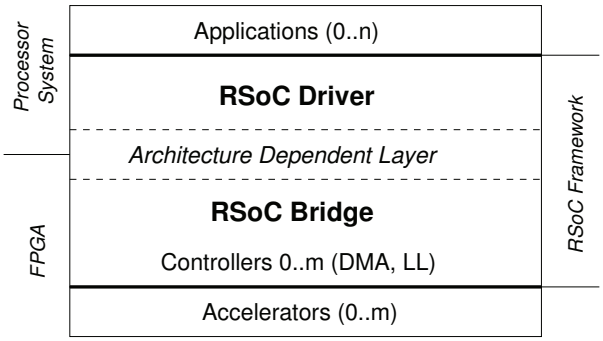
Přenosy opačným směrem (z programovatelné logiky do procesorového systému) je nutné podpořit přerušením některého procesorového jádra, popř. musí některé jádro provádět aktivní čekání (polling). Vlastnosti přenosů jsou dále výrazně ovlivněny architekturou konkrétního RSoC obvodu, která je dána výrobcem.

3.2 RSoC Framework

RSoC Framework [5] je subsystém zajišťující konzistentní rozhraní mezi softwarovou a hardwarovou částí aplikace. Aktuální implementace je postavená nad sběrnicovým systémem rodiny AMBA AXI [3], který je obvykle nativní na nejnovějších RSoC architekturách, ale je dostupný i na architekturách starších (např. Xilinx MicroBlaze). RSoC Framework předpokládá rozdělení systému na n softwarových aplikací a m akceleračních jednotek. Libovolná aplikace může komunikovat s libovolným akcelerátorem. RSoC Framework sestává ze dvou částí:

- RSoC Bridge – hardwarová komponenta (IP core) s platformově nezávislým rozhraním pro akcelerátory a s platformově závislým rozhraním upraveným pro konkrétní RSoC architekturu.

- RSoC Driver – softwarový ovladač (lze chápat např. jako modul jádra OS Linux) pro přístup k RSoC Bridge a zejm. k připojeným akcelerátorům. Ovladač poskytuje jednotlivým softwarovým aplikacím služby pro přístup k akcelerátorům bez podrobnější znalosti hardwarové architektury.



Obrázek 2: Architektura systému nad RSoC Framework

Hlavním cílem RSoC Frameworku je odstínění od platformově specifických problémů. Díky tomu by mělo být výrazně snažší portovat aplikace na různé čipy a různé operační systémy. Dalším cílem je zjednodušení vývoje aplikace. Komponenta RSoC Bridge vyřeší za vývojáře způsob přenosu dat mezi softwarovou a hardwarovou částí pomocí DMA řadičů. Pro aplikace, které požadují nízkou latenci (s ohledem na zvolený obvod RSoC) mohou být poskytnuty jiné řadiče, které nemají režii typickou pro DMA přenosy, beze změny rozhraní (na obrázku 2 označeno jako LL – Low-Latency). RSoC Driver je potom univerzální ovladač, který umí komunikovat s implementovanými řadiči a efektivně řídit předávání dat.

Pokud přihlédneme k faktu, že pro každou komunikaci mezi softwarovou a hardwarovou částí aplikace je třeba DMA řadič a jeho ovladač, je přidaná režie RSoC Frameworku minimální, protože řeší stejné problémy, který by nastaly i bez jeho zapojení. Režii mohou vkládat pouze vrstvy, které přizpůsobují interní rozhraní RSoC Frameworku rozhraním veřejným, které jsou neměnné. Na platformách s nativní podporou sběrnicového systému AMBA AXI je této režie minimum. Pro konkrétní aplikaci lze potom upravit činnost interních částí RSoC Frameworku tak, aby se přizpůsobily jejím požadavkům.

4 Částečná dynamická rekonfigurace

Částečná dynamická rekonfigurace FPGA spočívá v modifikaci interní konfigurační paměti SRAM. Díky tomu je možné změnit funkci části obvodu bez ovlivnění zbytku systému. Samotná rekonfigurovaná oblast musí být definována při návrhu hardwarového designu. Při rekonfiguraci musí být navíc vybraný obvod v rekonfigurovatelné oblasti vhodně pozastaven, aby nedošlo ke ztrátě dat, a také odpojen od všech sběrnic, aby nedošlo k nežádoucímu ovlinění ostatních komponent systému, popř. dokonce i k poškození FPGA čipu.

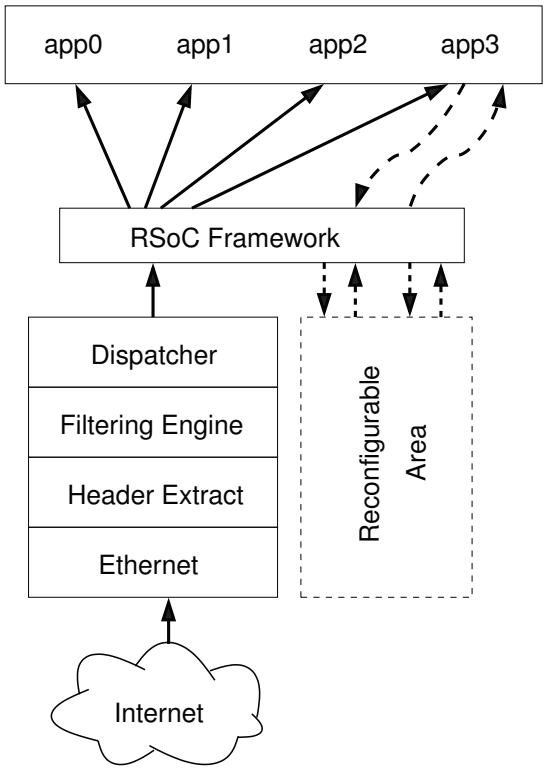
Pro tyto úkony lze s výhodou rozšířit RSoC Framework. Stávající aplikace získají podporu částečné dynamické rekonfigurace pouze přidáním rekonfigurovatelných oblastí, do kterých lze nahrávat různé akcelerační jednotky s ohledem na požadované komunikační vlastnosti (vysoká propustnost, nízká latence). Podpůrnou logiku, která je nezbytná pro každý rekonfigurovatelný blok, lze vložit do komponenty RSoC Bridge a tím usnadnit přenositelnost rekonfigurovatelného systému mezi platformami.

5 Monitorování počítačových sítí

V současné době se pro monitorování sítí začíná používat koncept Software Defined Monitoring [1] (SDM), který cílí zejm. na vysokorychlostní sítě s propustnostmi od 10 do 100 Gb/s. SDM počítá s nasazením na výkonném více-jádrovém serveru se specializovanou akcelerační kartou osazenou výkonným FPGA (např. karty COMBO-100G [4]). SDM využívá několika principů pro snížení zátěže procesorové části systému:

1. Úlohy jsou distribuované na procesorová jádra, která mezi sebou typicky nekomunikují.
2. Software řídí, která toky z provozu chce dostávat kompletní (pro hlubší analýzu), od kterých toků chce získávat pouze metadata (hlavičky) a u kterých toků stačí pouze sbírat agregované údaje (statistiky).
3. Hardware provádí předzpracování na základě požadavků ze softwarové vrstvy.
4. Tento způsob je efektivní, protože největší část provozu tvoří statisticky pouze několik nejsilnějších toků, které má smysl hlouběji analyzovat. Menší toky, u kterých by vznikla značná režie při zpracování, jsou analyzovány hardwarově.

Princip SDM je vzhledem ke své struktuře přímo portovatelný na architektury RSoC. Výhodou takového řešení může být snížení spotřeby a zmenšení celého zařízení. Je však třeba přihlédnout k faktu, že současné RSoC architektury nejsou dimenzovány na provoz nad 10 Gb/s a obsahují typicky max. 2 procesorová jádra.



Obrázek 3: Schéma SDM na obvodech RSoC.

Jak je znázorněno na obrázku 3, současnou hardwarovou architekturu SDM lze připojit přes RSoC

zpracování provozu procesory může znamenat zvýšení spotřeby systému, je možné některé části softwarových aplikací dynamicky přesouvat do zbývajícího prostoru v rekonfigurovatelné části čipu. Tím lze zvýšit propustnost systému i při zátěžích, které nemusí procesorový systém zvládat. Tato akcelerace nemá s principem SDM přímou souvislost, jedná se o rozšíření, které dovoluje provozovat SDM na platformě s omezeným výpočetním výkonem a velikostí čipu.

6 Závěr

V článku byl představen směr dizertační práce. Cílem práce je studium metod a návrh algoritmů pro využití částečné dynamické rekonfigurace ve vestavěných systémech v oblasti počítačových sítí. Praktickou ukázkou bude implementace Software Defined Monitoring (SDM) na obvodech RSoC. Protože RSoC obvody primárně nedisponují vysoce výkonnými procesory, je třeba lépe využívat FPGA a částečnou dynamickou rekonfiguraci. To vyžaduje vybrat vhodnou množinu akcelerovatelných operací, dále použití vhodného algoritmu pro plánování rekonfigurace s přihlédnutím k režii (latence), kterou s sebou částečná dynamická rekonfigurace přináší.

Poděkování

Tento článek vznikl za podpory projektů Architektury paralelních a vestavěných počítačových systémů, FIT-S-14-2297 a Moderní prostředky pro boj s kybernetickou kriminalitou na Internetu nové generace, VG20102015022.

Reference

[1] Kekely L., Puš V., Kořenek J.. Software Defined Monitoring of Application Protocols. In Proceedings of IEEE INFOCOM 2014 – IEEE Conference on Computer Communications. Toronto, 2014, pp. 1725–1733.

[2] Diessel, O. et al: Dynamic Scheduling of Tasks on Partially Reconfigurable FPGAs. IEE Proceedings – Computers and Digital Techniques, Volume 147, Issue 3, May 2000.

[3] ARM: AMBA Open Specifications. http://www.arm.com/products/system-ip/amba/amba-open-specifications.php.

[4] COMBO-100G webpage. https://www.liberouter.org/combo-100g/.

[5] RSoC-Framework webpage. rsoc-framework.com.

Polymorfní elektronika a metody syntézy

Adam Crha
Počítačové systémy, 1. ročník, Prezenční studium
Školitel: Richard Růžička

Fakulta informačních technologií VUT v Brně
Božetěchova 2, Brno, 612 66

icrha@fit.vutbr.cz

Abstrakt. Tato práce popisuje výzkum týkající se nekonvenční elektroniky. V úvodu jsou diskutovány principy, výhody a nevýhody nekonvenční elektroniky. Další část se zabývá elementárními stavebními prvky polymorfní elektroniky, tedy ambipolárními tranzistory. Poslední část je věnována dosud navrženým technikám pro syntézu polymorfní elektroniky a nakonec je zmíněna idea nové syntézní techniky.

Klíčová slova. Ambipolarita, polymorfní elektronika, syntéza, tranzistor, hradlo, číslicový obvod.

1 Úvod

V současné době je drtivá většina počítačových systémů založena na prvcích na bázi anorganických polovodivých materiálů, jako je křemík. Tyto prvky představují tranzistory, ze kterých jsou sestavena logická hradla realizující základní boolevské funkce. Z hradel jsou nakonec pomocí syntézy sestavovány složitější obvody vykonávající složitější funkci. Taková konvenční elektronika je navrhována známými automatizovanými postupy.

V dnešní době již existují zajímavé technologie, které mohou přinášet jisté výhody do systému, ve kterém jsou použity. Jedná se zejména o organické polovodiče, polovodiče na bázi grafenu, které vykazují rozdílná chování v závislosti na stavu okolního prostředí. Tohoto nestabilního chování je možné využít v takzvané polymorfní elektronice. Polymorfní elektronikou lze nazvat elektroniku, která je schopná provádět více funkcí v závislosti na stavu okolí. Cílem polymorfní elektroniky je šetřit a sdílet prostředky, které by byly požadovány při realizaci konvenční elektronikou.

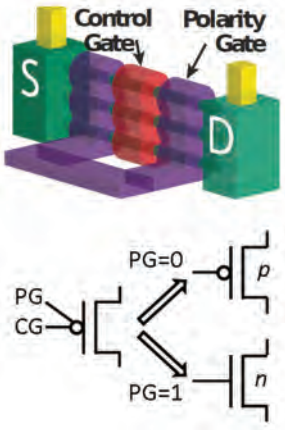
Tento článek pojednává o principech nekonvenční elektroniky na úrovni tranzistorů. Dále je popsán aktuální stav syntézních metod. V poslední části článku je lehce zmíněn stav práce a její cíl, jakožto nové metody návrhu nekonvenční elektroniky.

2 Ambipolarita

V několika posledních letech se začínají objevovat nové polovodičové materiály, které by mohly v budoucnu nahradit křemíkové polovodiče. Křemíkové polovodiče jsou považovány za stabilní polovodičové struktury, avšak dnes se již naráží na technologické limity. Mezi nové materiály je možné řadit například organické polovodiče, které mohou vykazovat oproti křemíku zvláštní chování. Příkladem zvláštního chování může být ambipolarita. Tranzistor, vyrobený z takového materiálu se pak může za určitých podmínek chovat jako tranzistor typu N, zatímco za jiných podmínek jako tranzistor typu P.

2.1 Ambipolární tranzistor

Jak již bylo řečeno v předchozím odstavci, tranzistor s ambipolárními vlastnostmi dokáže vykazovat rozdílné chování v závislosti na nějaké další fyzikální veličině. Nejčastěji jsou ambipolární tranzistory konstruovány se čtyrmi elektrodami. První tři elektrody, *GATE*, *SOURCE* a *DRAIN*, jsou totožné s konvenčními tranzistory typu N a P. Čtvrtá elektroda, často nazývaná „Polarity gate“, se používá k výběru požadovaného chování, tedy chování jako tranzistor typu N, nebo P. Na obrázku 1 je zobrazen čtyřelektrodový ambipolární tranzistor.



Obrázek 1: Ambipolarní tranzistor se 4 elektrodami [4].

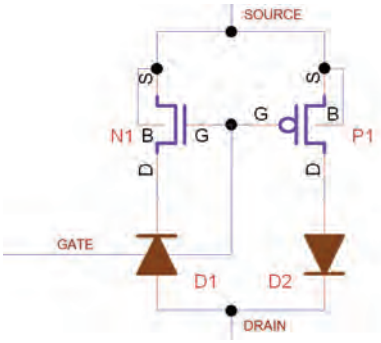
Tyto tranzistory již reálně existují a mnoho laboratoří je takový tranzistor schopno vyrobit. Nevýhodou tohoto tranzistoru je navíc řídící elektroda, která tak zvyšuje počet připojených vodičů k tranzistoru. V případě zvyšování počtu tranzistorů pak počet vodičů navíc narůstá lineárně. Výzkumná skupina na FIT VUT v Brně, zabývající se touto problematikou, se obává, že elektroda navíc je krokem zpět.

Snahou je tedy hledat ambipolární tranzistor, který má pouze tři elektrody. Oproti konvenční technologii nevzniknou nevýhody spojené se čtvrtou elektrodou. Selekce požadovaného chování tranzistoru by bylo možné například provádět polaritou přiloženého napětí na elektrody *SOURCE* a *DRAIN*. Avšak není prozatím známo, že by takový tranzistor reálně existoval. Následovaly tedy testy ambipolárního chování u konvenčních tranzistorů.

2.2 Ambipolární chování konvenčního tranzistoru

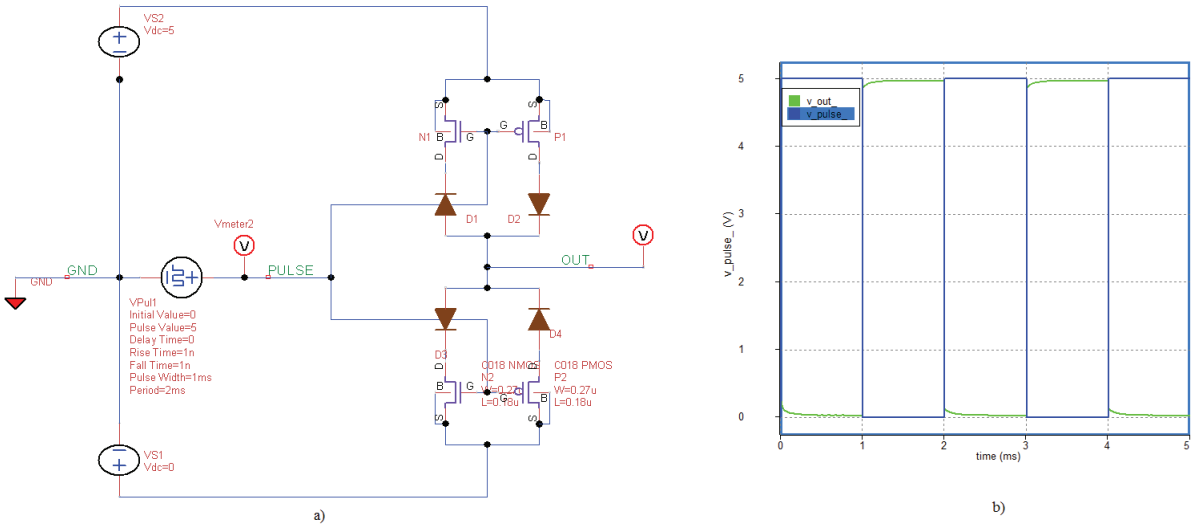
Jelikož není známo, že by existoval reálný tříelektrodový ambipolární tranzistor, bylo provedeno několik testů s konvenčními tranzistory typu N a typu P. Testy byly prováděny v simulátoru SPICE a na reálných součástkách. Při záměně polarity na těchto typech tranzistorů docházelo k požadované funkci částečně, správnost požadovaného napětí výstupu byla velmi závislá na zátěži. V reálné aplikaci tak není možné použít konvenční tranzistor a vyžadovat od něj ambipolární chování.

Na tuto situaci reagoval Ing. Radek Tesař pokusem o návrh náhradního zapojení ambipolárního tranzistoru složeného z více konvenčních polovodičových součástek. Náhradní schéma ambipolárního tranzistoru se skládá ze dvou konvenčních tranzistorů a dvou polovodičových diod, jak je možné spatřit na obrázku 2. Volba režimu tranzistoru je definována polaritou napájecího napětí mezi elektrodami *SOURCE* a *DRAIN*. Správnost chování náhradního schématu ambipolárního tranzistoru bylo ověřeno v simulátoru SPICE a taktéž pomocí reálného zapojení.



Obrázek 2: Náhradní schéma ambipolárního tranzistoru.

Na obrázku 3 vlevo je možné spatřit schéma zapojení invertoru, složeného z náhradního zapojení ambipolárních tranzistorů. Jeho chování je vždy korektní, nehledě na polaritu přiloženého napětí. Na obrázku 3 vpravo je průběh simulace tohoto zapojení. Je zde uváděn průběh pouze pro pozitivní polaritu přiloženého napětí, jelikož pro negativní polaritu je průběh výstupního signálu totožný [5].



Obrázek 3: a) Invertor, složený z náhradních ambipolárních tranzistorů. b) Simulace obvodu s pozitivní polaritou.

3 Polymorfní elektronika

V oblasti počítačových systémů se polymorfní elektronikou rozumí elektronické číslicové obvody, které dokážou vykonávat více než jednu funkci, zatímco zapojení elektronického obvodu je stále stejné. Volba funkce, kterou obvod vykonává je závislá na stavu okolního prostředí (teplota, tlak, vlhkost, polarita napětí, ...). Všechny požadované funkce obvodu jsou navrženy úmyslně. Jedná se tak o požadované funkce obvodu, nikoliv například o poruchový stav vyvolaný překročením provozních parametrů obvodu. Stav okolního prostředí je možné přesně popsat, typicky nějakou fyzikální veličinou. Pak je možné pro konkrétní hodnotu této veličiny určit, jakou funkci bude polymorfní obvod realizovat.

Takový polymorfní obvod je nejčastěji reprezentován acyklickým grafem $G = (V, E, \phi)$, kde V je množina uzlů (V/V hradel), $E = \{(a, b)|a, b \in V\}$ je množina hran (spojů) a $\phi = \{\varphi_1, ..., \varphi_n\}$ je množina zobrazení a platí $|\phi| > 1$. Každé zobrazení $\varphi_i \in \phi$, přiřazuje každému uzlu z V hradlo z množiny K, $\varphi_i : V \to K$ pro $\forall i = 0..n$.

3.1 Návrh polymorfních obvodů

Návrh polymorfního obvodu může být popsán jako hledání grafu G, který reprezentuje vnitřní zapojení obvodu tak, aby byl obvod schopný vykonávat jednu ze všech požadovaný funkcí v závislosti na stavu prostředí. Při změně funkce obvodu se tedy může změnit pouze funkce uzlů, graf G (zapojení obvodu) zůstává stejný.

Návrh číslicových obvodů probíhá v současnosti na úrovni hradel. Samostatná hradla pak na úrovni tranzistorů.

Na základě experimentů návrhu polymorfních obvodů vyšlo najevo, že navrhovat obvody pouze z polymorfních hradel není příliš vhodné. Jako vhodné se jeví navrhovat polymorfní obvody jež obsahují jak polymorfní, tak konvenční hradla. Je nutné podotknout, že počet konvenčních hradel přesahuje počet polymorfních hradel navrženého obvodu. V mnoha případech také stačí použít polymorfní hradlo jednoho typu, jedná-li se o hradlo, které realizuje logicky úplné funkce (např. NAND/NOR). Pokud by bylo v návrhu použito více typů polymorfních hradel, mohlo by to vést k lepšímu řešení, avšak za cenu složitosti problému návrhu (zvětšení stavového prostoru) [1].

3.2 Dosud známé metody návrhu polymorfních obvodů

V současnosti již bylo nalezeno několik metod pro návrh polymorfních obvodů, avšak každá z nich nese nějaká omezení.

Následuje výčet metod pro návrh polymorfních obvodů:

3.2.1 Ad hoc

Ad hoc přístup je považován za návrh obvodů bez použití jakýchkoliv návrhových technik a nástrojů. Předpokládají se pouze elementární znalosti a zkušenosti návrháře. Touto metodou lze navrhovat pouze velmi, velmi malé obvody. Metoda je tedy pro větší obvody nepoužitelná.

3.2.2 Evolucí

Evoluční návrh polymorfních obvodů je v současnosti jedním z nejefektivnějších přístupů. Evoluční návrh je schopný pracovat na velmi velkém prostoru logických funkcí ve srovnání s konvenčními metodami syntézy [3]. Algoritmus tak nachází mnoho řešení, které často nejsou korektní, avšak postupem algoritmu se nacházejí řešení kvalitnější. Algoritmus generuje nová řešení tak dlouho, dokud řešení neodpovídá pravdivostní tabulce požadované funkce, eventuálně dokud obvod nesplňuje nějaké další kritérium.

Evolučním návrhem polymorfních obvodů se zabýval na území FIT VUT v Brně výzkumný tým L. Sekaniny. K návrhu obvodů využívali zejména Kartézské genetické programování (CGP). Návrh polymorfních obvodů pomocí CGP je téměř stejný ve srovnání s návrhem konvenčních obvodů. Rozdíl spočívá pouze ve fitness funkci, ve které je nutno zajistit, aby korektnost obvodu byla ohodnocena pro všechny funkce/režimy, které má obvod vykonávat.

Nevýhody evolučního návrhu spočívají například v mnohdy malé škálovatelnosti nalezených řešení. Nalezení složitějších obvodů vyžaduje prohledávání velkého stavového prostoru a tím se zvyšuje časová náročnost k nalezení kvalitního řešení [1].

3.2.3 Polymorfní multiplexování

Další technikou pro návrh polymorfních obvodů je polymorfní multiplexování. Tuto techniku navrhl Gajda a Sekanina. Jedná se o jednoduchou metodu, která se snaží využívat principy konvenčního návrhu obvodů. Ve zkratce je princip takový: Každá funkce, kterou má polymorfní obvod vykonávat, je navržena konvenčně z konvenční elektroniky. Výstupy každého takto navrženého obvodu se připojí na takzvaný polymorfní multiplexor, který provádí selekci daného vstupu v závislosti na stavu okolního prostředí.

Tento přístup není příliš efektivní z hlediska plochy (žádná funkce nesdílí podobné části). Což se právě od polymorfní elektroniky očekává [1] [2].

3.2.4 PolyBDD

Metoda, kterou navrhl Zbyšek Gajda v rámci své disertační práce. Tento přístup je určen pro návrh polymorfních obvodů a využívá binárních rozhodovacích stromů, odtud PolyBDD. Je využíváno tzv. multiterminálních uzlů, což znamená, že terminální uzel může nést celočíselnou hodnotu. Tato celočíselná hodnota reprezentuje primitivní polymorfní hradlo. Velmi zjednodušený princip BDD: Z pravdivostní tabulky se vytvoří BDD dle algoritmu popsaného v [2]. Poté se BDD převede na schéma obvodu tak, že neterminální uzly se přímo napojí na dvouvstupové multiplexory řízené danou proměnnou a terminální uzly se implementují jako polymorfní primitiva typu (identita/negace, negace/identita, ...).

Nevýhody metod PolyBDD a polymorfního multiplexování spočívají zejména v tom, že polymorfní hradla jsou v nich zastoupena ve velmi malém množství a slouží prakticky jako přepínače vstupů / výstupů. Není tak využito potenciálu polymorfních hradel v maximální možné míře [1] [2].

4 Disertační téma - nové metody syntéz polymorfních hradel

Vzhledem k nedokonalostem stávajících metod je vhodné, aby výzkum syntézy polymorfních hradel stále pokračoval. Je žádoucí, aby polymorfní hradla byla ve výsledném obvodu maximálně využita a aby výsledný polymorfní obvod sdílel co největší množství hradel pro všechny požadované funkce. Cílem disertační práce je najít metodu, která bude schopná přímočaře navrhnout polymorfní obvod bez negativních syndromů dosud známých metod.

Obvod je možné reprezentovat graficky, stromem, kde uzly reprezentují hradla a hrany propoje. Každá funkce má vlastní strom. K sestavení takového stromu se používají přístupy konvenční syntézy. Jak již bylo řečeno, je žádoucí, aby všechny funkce sdílely co největší množství hradel, tedy aby dva různé stromy sdílely co největší počet uzlů. Dle dosavadních poznatků autor článku usuzuje, že mohou existovat dva přístupy:

První přístupem je hledání podobností mezi všemi stromy (jeden strom - jedna funkce) a snažit se tyto podobnosti sdílet ve výsledném stromu (výsledný polymorfní obvod). To znamená - navrhnout obvody konvenčně a poté je „slepit“ dohromady. Tento přístup se však jeví jako velmi komplikovaný.

Druhým přístupem je sestavování polymorfního obvodu od počátku návrhu. Polymorfní obvod je tak tvořen od základů a předpokládá se, že díky tomuto přístupu by mohlo mnoho společných částí být odhaleno již ve fázi návrhu.

4.1 Idea syntézy polymorfních obvodů

Na základě předchozích úvah vznikla idea jak navrhovat polymorfní obvody. Tato idea je jakýmsi hybridem mezi oběma zmíněnými přístupy. Metoda však není stále dokončená a obsahuje zatím mnoho otázek. Proto je v následujících řádcích popsána velmi lehce.

Nejdříve bylo stanoveno několik omezení. Polymorfní obvod bude obsahovat pouze polymorfní hradla typu NAND/NOR. Metoda bude pracovat pouze s hradly, které jsou popsatelné booleovou algebrou a omezíme se pouze na obvody realizující pouze dvě funkce. Hlavní ideou je postupně sestavo-

vat dva různé obvody tak, aby obsahovaly co nejvíce podobností, které se později jednoduše spojí do jednoho polymorfního obvodu. Prvním krokem je vytvoření pravdivostní tabulky pro obě požadované funkce. Poté se z pravdivostní tabulky první funkce vytvoří formule v konjunktivní normální formě a z druhé funkce formule v disjunktní normální formě. Tím získáme dvě podobné formule, kde v jedné budou termy spojeny operátory AND a v druhé budou termy spojeny operátory OR. Tyto výrazy je vhodné upravovat pomocí booleovy algebry tak, aby se co nejvíce podobaly a hradla AND a OR se změnila na NAND a NOR. Tam, kde se bude ve funkci 1 vyskytovat operátor NAND a ve funkci 2 operátor NOR, bude použito polymorfní hradlo. Ostatní operátory jsou realizovány běžnými konvenčními hradly.

Protože některé části obvodů není možné spojit, je nezbytné použít některá nová polymorfní hradla, která budou sloužit jako polymorfní multiplexor, identita/negace a negace/identita. Tyto hradla byla v rámci doktorského studia již navrhnuta evolučním přístupem (funkce přepínána polaritou) na úrovni tranzistorů, avšak nebyla zatím publikována. Tyto polymorfní hradla (na úrovni hradel) také navrhl obecně Gajda ve své disertační práci [2].
Dle dosavadních experimentů metoda prozatím nedosahuje kvalitních výsledků ve srovnání se stávajícími metodami, ale již v této fázi neobsahuje faktor náhodnosti a nekontrolovatelnosti. Všechny kroky jsou uvědomělé a cílem metody je využít polymorfní hradla uvnitř obvodu, nikoliv na multiplexování vstupů / výstupů.

5 Závěr

Tato práce chronologicky popisuje průběh výzkumu týkající se polymorfní elektroniky. Nejdříve byly studovány principy polymorfní elektroniky na úrovni tranzistorů, tedy elementárních prvků, ze kterých jsou vytvořena hradla. V případě polymorfních hradel se jedná zejména o ambipolární tranzistory. Bylo zjištěno, že valná většina ambipolárních tranzistorů využívá čtyři elektrody. Tříelektrodové ambipolární tranzistory jsou dnes velmi vzácné. Vzhledem k nedostupnosti takového tranzistoru bylo vytvořeno náhradní schéma, které bylo otestováno simulátorem SPICE a na reálném HW. Této oblasti se poté začal intenzivně věnovat Ing. Tesař. Následně se výzkum posunul o úroveň výš, na úroveň hradel. Byly prostudovány současné principy syntézy polymorfních obvodů. Informace získané o stávajících metodách napověděly, že syntéza polymorfních obvodů není stále ideální. Začala tak vznikat idea o nové metodě navrhující polymorfní obvody a směr disertační práce začíná být přesnější.

Reference

[1] Růžička R.: Polymorfní elektronika, habilitační práce, FIT VUT v Brně, 2011.

[2] Gajda Z.: Evolutionary Approach to Synthesis and Optimization of Ordinary and Polymorphic Circuits, PhD thesis, Brno, FIT BUT, 2011.

[3] Miller, J., Thomson, P.: Cartesian Genetic Programming. Proc. of the 3rd European Conference on Genetic Programming EuroGP 2000, LNCS 1802, Springer 2000, str. 121 ? 132.

[4] Turkyilmaz, O.; Clermidy, F.; Amaru, L.G.; Gaillardon, P.-E.; De Micheli, G., "Self-checking ripple-carry adder with Ambipolar Silicon NanoWire FET,"Circuits and Systems (ISCAS), 2013 IEEE International Symposium on , vol., no., pp.2127,2130, 19-23 May 2013

[5] Tesař R., Šimek V., Růžička R. Crha A.: Polymorphic Electronics Based on Ambipolar OFETs, pages 106–111, EDS 2014 IMAPS CS International Conference Proceedings, 2014, Brno, CZ, FIT BUT, ISBN 978-80-214-4985-5.

ADAPTIVE PID CONTROLLER

František Kudlačák
Applied Informatics, first class, full-time study
Supervisor: Associate Professor Tibor Krajčovič

Affiliation (Faculty of Informatics and Information Technologies, Slovak University of Technology)
Ilkovičova 2, 842 16 Bratislava, Slovakia

frantisek.kudlacak@stuba.sk

Abstract. In this paper is presented state of art in field of tuning methods for PID (Proportional-Integral-Derivative) controllers. New tuning method for online tuning is proposed. The proposed approach is based on the dynamic processes with disturbance and on various conditions from environment. Adaptive controller computes these variables and creates control signal for process. Output of process should have the lowest error in compare to desired output value.

Keywords. PID controller, Online tuning method, Control system.

1 Introduction

Control systems are integral parts of our lives. They control dynamic systems whose behavior can change over time, often in response to the external stimuli. There are two main groups of dynamic system, the open loop systems and the closed loop systems [1].

The open loop systems react only on inputs. Closed loop system reacts on their output so these systems are interconnected into a cycle. This connection can be called feedback. Feedback has many different properties that can be exploited in designing system. Feedback can make a system resistant to the external disturbances. Feedback can be also used to create linear behavior form nonlinear components. This approach is commonly used in electronics. Feedback allows system resistance to individual variations of external disturbances. So There can be chosen parameter which will be ignored and which will effect system.

There are disadvantages of feedback as well. It creates dynamic instabilities in a process and may cause oscillation of outputs even runaways from desired values. In practical solutions feedback introduce unwanted noise and disturbance into sensor system, so there are required filters. In a feedback control system, information about performance of the system is measured and used to correct behavior of process.

1.1 PID controller

The PID controllers are the most common control algorithms. According to research 97 % of all industrial controllers utilize PID control logic [2]. They are easy to use, easy to implement and they are robust. PID controller calculates an error values between measured process output and desired set point. In each loop controller attempts to minimize error value by adjusting process through process inputs. Block diagram of PID controller is shown in Figure 1.

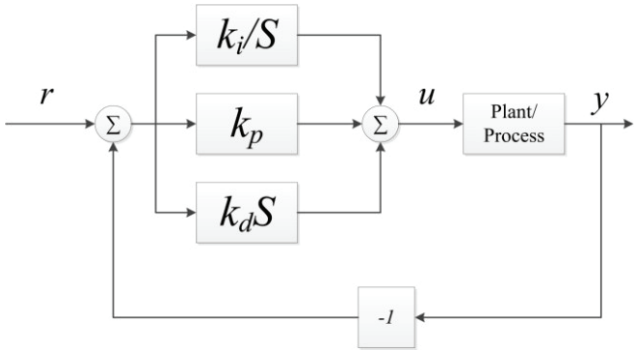


Figure 1. PID controller

Where *u* is control signal for process, and it is adjusted from error *e* and from command signal *r* called the reference signal or setpoint. Control signal *u* is computed from proportional, derivative and integral term. These terms are affected by error *e*. It is computed from desired value *r* and actual output value of the PID controller. Input output relation is stated in following formula:

$$u = k_p e + k_i \int_0^t e(\tau)d\tau + k_d \frac{de}{dt} = k_p\left(e + \frac{1}{T_i}\int_0^t e(\tau)d\tau + T_d \frac{de}{dt}\right) \quad (1.1)$$

Where k_p, k_i and k_d are parameter of stated PID controller and are called gains. There can be used another set of parameters k_p, T_i and T_d, where T_i is called integral time, T_d is called derivative time and τ is variable of integration, τ takes value from 0 to present time. The action is the sum of three terms: proportional feedback, the integral term and the derivative action.

2 Tuning methods

There are many approaches to tune PID controllers. First methods are based on manual setting of parameters. These methods needs experienced person who know plant process and conditions. Advantage of these methods is non added algorithm into tuning process. Second type of tuning methods are offline tuning methods. These tuning methods compute PID controller parameters outside of loop, measuring output of process, and responses of PID controller. After determination of best parameters, they are applied to PID controller. Third methods are online tuning methods. Parameters of PID controller are changing during executing PID controller function. There can be determined first values of parameters, or random approach can be chosen for first values.

2.1 Dominant pole assignment tuning method with genetic algorithm

The dominant pole assignment method is applied to a test group of plants. There is found correlation between process output behavior and the controller parameters. In this method, the dominant poles are assigned as integration of the error. Step load disturbance is minimized subject to the constraint on maximum sensitivity. A set-point weighting is used afterwards to improve set-point response of the system [3].

Genetic algorithms can be applied for nonlinear optimization [4]. In this case genes are dominant poles [5].

2.2 Ziegler-Nichols tuning method

Ziegler-Nichols tuning method is heuristic method [6]. Provides controllability and high performance in field of DC motor control [7]. It uses consecutive steps to determine PID parameters [8]. This method is applied to system output with step responses. Type of responses is typical for a first order system with transportation delay. In Figure 2 is shown response curve for this method [9].

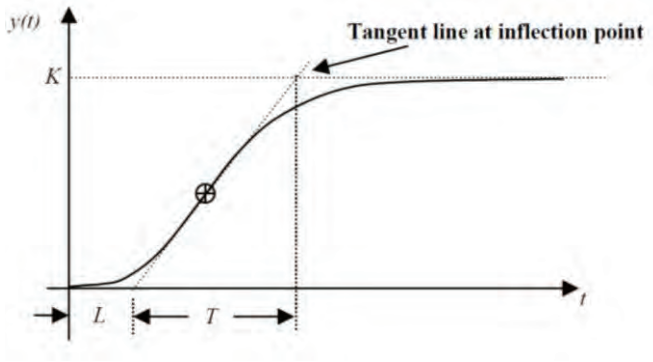


Figure 2. Response curve for Traditional Ziegler-Nichols method

Parameter L is the time delay and parameter T is time constant. Their values are found by drawing a tangent at the point of infection and intersection with time axis and intersection with stable state value. The plant model can be described by following equation:

$$G(s) = \frac{Ke^{-sL}}{TS+1} \tag{2.1}$$

If plant process cannot be derived, processes can approximate by previous model in many cases. If there can be recorded outputs of plant process, the output signal can be recorded into plot and parameters *K, L* and *T* can be extracted. But in many cases curve fitting approaches can be used to create desired model. If deviation between model and measured data is too big, PID controller does not have to work properly.

2.3 Tuning method based on particle swarm optimization

Particle in particle swarm optimization method represents a solution to the problem, and it is defined by its position and by its velocity. Particle is moving through solution space. And best solution is remembered. Advantage of this approach is in changeable velocity of searching in space. Solutions are parameters of PID controller, chosen combinations can be represented by particle. After achieving best solution, these parameters are applied to PID controller.

2.4 Internal model tuning method with neural networks

Effective method for robust control systems is the internal model control method. Application of this method depends on complexity of model and performance requirements stated by the designer [10]. Adding filtering into cascade with internal model controller, can improve robustness of whole system. In some cases the IMC (Internal Model Control) controller leads to PID controller construction. There was developed tuning techniques that are based on IMC-PID tuning rules [11] and improve robustness of system compared to classical tuning methods such as Ziegler-Nichols tuning method. IMC strategy for controlling processes involving theoretical model of controlled process. So output of model and real values can be compared. Adding another techniques like neural networks into system, is applicable for nonlinear modeling and inverse modeling [12].

2.5 Least square support vector machine with kernel tuning method

Effective Tuning method which use support vector machine does not stuck in local minima and it can provide great generalization with few training data. The disadvantage of solution with support vector machine is time consuming calculation [13]. The main component of support vector machine is the kernel component, which is nonlinear mapping function which convert linearly non-separable input

into high-dimensional space where data can be separable linearly. Kernel functions are generated parametrically, and these parameters can influence features of mapped data in working space. For parameters selection, there were used genetic approach [14]. Parameters have been set offline a obtained kernel parameters have been employed in online control loop. Particle swarm optimization has been used as well [15]. Another examples of offline computing kernel parameters are Cat Swarm Optimization [16], Grid-Diamond Search [17] and Simulated Annealing [18].

3 Proposed adaptive PID controller

Proposed adaptive online PID controller is shown in Figure 3.

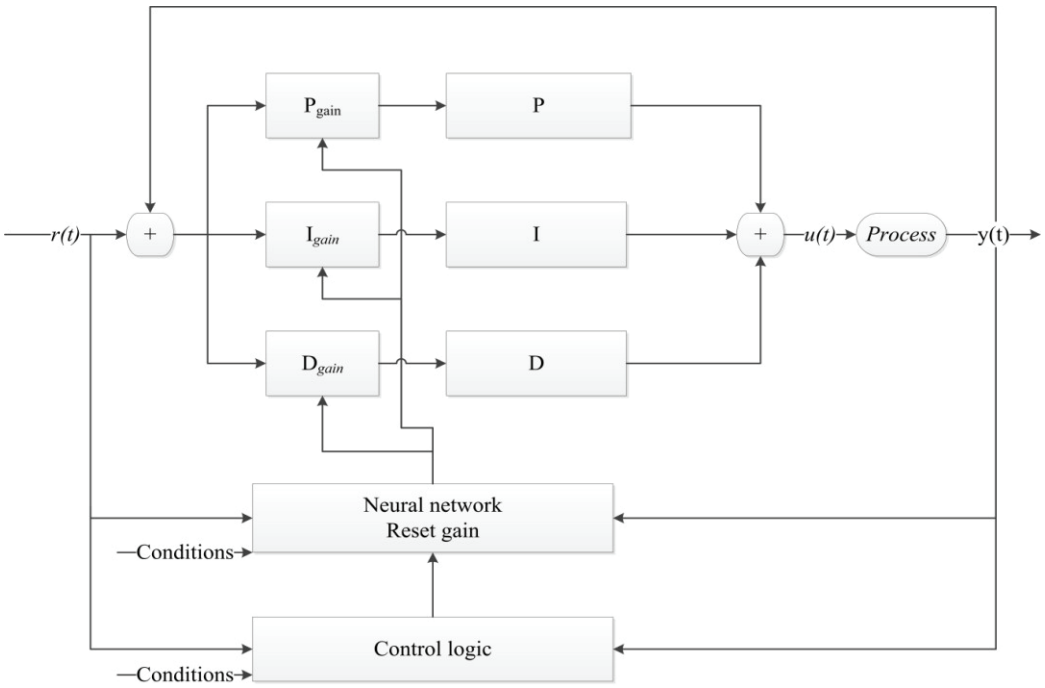


Figure 3. Proposed adaptive PID controller

Where *r(t)* is input or reference signal, *u(t)* is output function from PID controller and *y(t)* is output from process. Adaptive PID controller is composed from three main parts. First is PID controller function blocks. It controls gain parameters of PID controller and can control integration time. So how long into past will be error summed. Second part is neural network together with reset module. This part directly changes parameters of PID controller. Third part control logic acquire all information from input, output and from PID controller. Evaluates conditions of each part and controls behavior of module which directly interact with parameters of PID controller.

4 Proposed thesis

Present tuning methods do not provide sufficient controlling abilities for dynamic processes, with dynamic environment. They can provide sufficient controller for stationary processes with predictable conditions. The main goal of doctoral thesis is to create new type of PID controller with new tuning method, which will use dynamic neural networks with variable learn rate. This new PID controller should decrease output error of PID controller and adaptation time of process.

From main goal are derived these partial goals:

- Design a new structure of PID controller, with aiming on dynamic processes and changeable conditions
- Propose a new complex tuning method for parameters of adaptive PID controller.
- Create new algorithm for controlling tuning characteristics of proposed tuning method.
- Create a new environment for simulating dynamic processes and PID controllers related to them.
- Experiments on simulated data and comparison with existing online adaptive PID controllers.
- Implementation and experimental test on embedded system with comparison which existing solutions.

5 Conclusion and further work

In this paper is stated state of art in field of tuning methods for PID controllers. Each tuning method can use slightly different model of PID controller. Tuning methods can adjust PID parameters online or offline. In case of online adjusting, tuning algorithm has to be part of PID controller.
After analyzing problem field, today adaptive controller can react on changing condition within predefined speed. So in variable condition error decreases into excepting boundaries after long time, if desired points are changing rapidly or process is changing during execution. Internal model control system can provide better results if there is good description of plant process. But when parameters of plant process are changing over time, and these changes cannot be predicted, model of plant process in IMC controller will cause a lot of noise and disturbances, because model will not change like real plant process.
After analysis of these problems, there was defined a need for a new adaptive PID controller, which will be suitable for changing environment and for processes which are changing over time. Proposed new controller is adaptive online PID controller with variable speed of learning, so it can adjust to changes from outer environment or from different plant process. For example quad copter during flight decrease voltage of battery and response time of controller is changing too.

In next stages of project there need to be done theoretical description of proposed new PID controller. After description there will be proof of concept on simulated test data, and results will be compared to other tuning methods for PID controllers. After evaluation of results there will be implementation of proposed PID controller into real device. At last there will be testing of PID controller in real environment and results will be compared mainly to PID tuning methods which use neural networks.

Acknowledgment

This work has been supported by the grant No. 1/1008/12 of the Slovak VEGA Grant Agency.

Publications

1. F. Kudlačák: Variometer with GPS logger. *Information Sciences and Technologies. Bulletin of the ACM Slovakia Vol. 4*, No. 2. p. 47-50. ISSN 1338-1237.
2. F. Kudlačák, J. Laštinec: Riadiace a kontrolné systémy elektrickej formule. *Perspektívy elektromobility III.*, príloha časopisu Elektro. Praha, FCC Public, 2013 ISSN 1210-0889.
3. F. Kudlačák: Variometer with GPS Logger. *Student Research Conference 2012. Vol. 2 : 8th Student Research Conference in Informatics and Information Technologies*, April 25, 2012. Bratislava: Nakladateľstvo STU, 2012, p. 317-322. ISBN 978-80-227-3690-9.

4. F. Kudlačák: Atmospheric Modelling via Flying Platform. *Student Research Conference 2013. Vol. 2 : 9th Student Research Conference in Informatics and Information Technologies*, April 23, 2013. Bratislava: Nakladateľstvo STU, 2013, p. 325--330. ISBN 978-80-227-3911-5.
5. F. Kudlačák: Synthesis of Asynchronous Sequential Circuits in High-performance Computing. *Student Research Conference 2014. 10th Student Research Conference in Informatics and Information Technologies*, April 23, 2013. Bratislava: Nakladateľstvo STU, 2013, p. 423. ISBN 978-80-227-4153-8.

References

[1] K. J. Aström a R. M. Murray, Feedback Systems, New Jersey: Princeton University Press, 2010.

[2] L. Desborough a R. Miller, „Increasing customer value of industrial control performance monitoring — Honeywell's experience," rev. *Sixth International Conference on Chemical Process Control, Vol. 98*, 2002.

[3] K. J. Astrom a T. Hagglund, PID Controllers: Theory, Design, and Tuning., North Carolina: Research Triangle Park, 1995.

[4] M. Gen a R. Cheng, Genetic Algorithms & Engineering Optimization, John Wiley, 2000.

[5] J. C. Shen, „New Tuning Method for PID Controller," rev. *Proceedings of the IEEE International Conference on Control Applications*, Mexico City, 2001.

[6] J. G. Ziegler a N. B. Nichols, „Optimum setting for automatic controllers," rev. *Trans. ASME, vol. 64*, 1942.

[7] K. A. Naik a P. Shrikant, „Stability Enhancement of DC Motor using IMC Tuned PID Controller," rev. *International Journals of Advanced Engg. Science and Technologies, vol. 4, Issue No. I*, 2011.

[8] N. Kamaruddin, Z. Janin, Z. Yusuf a M. N. Taib, „PID Controller Tuning for Glycerin Bleaching Process Using Well-Known Tuning Formulas- A Simulation Study," rev. *Proc. of 35th Annual Conference of iEEE on Industrial Electronics*, 2009.

[9] P. Solatian, S. H. Abbasi a F. Shabaninia, „Simulation Study of Flow Control Based On PID ANFIS Controller for Non-Linear Process Plants," rev. *American Journal of Intelligent Systems*, 2012.

[10] D. E. Rivera, M. Morari a S. Skogestad, „Internal model control. 4. PID controller design," rev. *Ind. Eng. Chem. Process Des. Dev.*, 1986.

[11] R. Vilanova, „IMC based Robust PID design: Tuning guidelines and automatic tuning," rev. *Journal of Process Control, Vol. 18*, 2008.

[12] I. Rivals a L. Perzonnaz, „Internal Model Control Using Neural Networks," rev. *Proceedings of the IEEE International Symposium on Industrial Electronics*, Warsaw, 1996.

[13] J. Zhao, P. Li a X. S. Wang, „Intelligent PID Controller Design with Adaptive Criterion Adjustment via Least Squares Support Vector Machine," rev. *21st Chinese Control and Decision Conference*, 2009.

[14] S. Wanfeng, Z. Shengdun a S. Yajing, „Adaptive PID Controller Based on Online LSSVM Identification," rev. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Vols 1-3*, 2008.

[15] S. W. Lin, K. C. Ying, S. C. Chen a Z. J. Leel, „Particle swarm optimization for parameter determination and feature selection of support vector machines," rev. *Expert Systems with Applications Volume: 35 Issue: 4*, 2008.

[16] K. C. Lin a H. Y. Chien, „CSO-Based feature selection and parameter optimization for support vector machines," rev. *Joint Conference on Pervasive Computing*, Taiwan, 2009.

[17] L. K. Hou a Q. X. Yang, „Study on parameters selection of LSSVR based on Grid-Diamond search method," rev. *International Conference on Machine Learning and Cybernetics*, 2009.

[18] F. Yan, X. W. Wu a S. Wang, „SA optimizing algorithm of SVM super-parameters," rev. *International Workshop on Geoscience and Remote Sensing*, 2008.

ŠIROKOPÁSMOVÁ BEZDRÔTOVÁ KOMUNIKÁCIA PRE IMPLANTOVATEĽNÉ BIOSENZORY

Martin Kováč
Mikroelektronika, 1. ročník, denná prezenčná forma štúdia
Školiteľ: Viera Stopjaková

Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave
Ilkovičova 3, 812 19 Bratislava

martin_kovac@stuba.sk

Abstrakt. Tento príspevok pojednáva o analýze aplikovateľnosti implantovateľných biosenzorov v ľudskom tele a to najmä z pohľadu obmedzenia ich celkovej plochy, ktorá významným spôsobom limituje i dostupnosť elektrickej energie potrebnej pre napájanie takýchto implantátov. Výsledná plocha a spotreba energie sa tak stávajú primárnymi vstupnými požiadavky pre návrh integrovaných obvodov tvoriacich inteligentný senzorický systém. Nakoľko bezdrôtový komunikačný modul (vrátane antény) patrí aj energeticky aj plošne medzi najnáročnejšie časti implantovateľných biosenzorov, je práve jeho analýza a návrh predmetom nášho výskumu. Zameriame sa na širokopásmovú komunikáciu (angl. *Ultra-wideband communication* - UWB), ktorej nasadenie v takýchto systémoch je zatiaľ len v teoretickej rovine. Prvotným cieľom je teda priniesť do tejto oblasti nové riešenie, založené na integrácii širokopásmovej antény spolu so zvyškom systému na jednom čipe a tým ušetriť miesto pre prípadnú implementáciu ďalších mikromechanických štruktúr (senzor, energetický zberač, aktuátor a pod.)

Kľúčové slová. UWB technológia, špirálová anténa, biosenzory, implantáty, WBAN

1 Úvod

Zdravotníctvo rovnako ako aj iné oblasti spoločenského života, priemyslu a vedy, podliehajú nepretržitej inovácii a modernizácii. Stimulom je neustály nárast počtu ľudí, zvyšujúci sa podiel starnúcej populácie, či zvýšené požiadavky na lekársku starostlivosť. Zlepšovanie zdravotníckej starostlivosti a zvyšovanie kvality života starnúcej časti populácie sa tak stávajú jednou z hlavných priorít Európskej Únie v zmysle využitia najnovších technológií, čo dokazuje aj grantový program HORIZON 2020 [1].

Zdravotná starostlivosť poskytovaná priamo v nemocnici a klinických zariadeniach je však finančne nákladná. Z tohto dôvodu sa súčasný výskum zameriava na vývoj mobilných inteligentných senzorových systémov, umožnujúcich kontinuálne monitorovanie zdravotného stavu pacienta. Tieto asistenčné systémy sú umiestnené buď priamo na tele alebo v niektorých prípadoch dokonca realizované ako vnútrotelový biosenzor, kde tvoria tzv. komunikačný uzol. V obidvoch prípadoch je teda potrebný bezdrôtový prenos údajov, avšak pre implantovateľné biosenzory (IB) sú požiadavky stanovené oveľa striktnejšie. Hlavné požiadavky zahrňujú: minimálnu veľkosť (závisí od aplikácie, zvyčajne $\leq 1\ cm^3$) a hmotnosť ($\leq 1\ g$), veľmi nízku spotrebu, dostatočné napájacie napätie (závisí od technológie, zvyčajne $\geq 1,2\ V$ pre digitálne obvody), vysokú spoľahlivosť ($\geq 20\ rokov$), vysokú biokompatibilitu a nízku toxicitu (napr. použitie titánového puzdra), pomerne vysokú prenosovú rýchlosť a nízku latenciu (závisí od apli-

kacie, všeobecne sa predpokladá 1 MHz), striktnú bezpečnostnú politiku, maximalny vyžiarený výkon na jednotku hmotnosti, maximálnu povolenú intenzitu magnetického a elektrického poľa [2].

Pre IB aplikácie organizácia IEEE (angl. *Institute of Electrical and Electronics Engineers*) predstavila medzinárodný štandard IEEE 802.15.6.2012 definujúci fyzickú a prístupovu vrstvu v bezdrôtovej telovej sieti (angl. *Wireless Body Area Network* - WBAN). Táto norma zahŕňa aj širokopásmovú UWB technológiu, ale iba pre výmenu dát v rámci bezdrôtovej personálnej siete (angl. *Wireless Personal Area Network* - WPAN). Avšak existencia tohto štandardu podporuje myšlienku nasadenia UWB technológie v IB, a to vzhľadom na menej náročný proces vývoja požiadaviek pre vzájomnú kompatibilitu s mimotelovou sieťou podporujúcou UWB komunikáciu. Do budúcnosti sa predpokladá istá optimalizácia prístupovej vrstvy z dôvodu rozdielnych energetických nárokov monitorovacích jednotiek komunikujúcich v sieti.

2 Implantovateľné biosenzory a zdroje energie

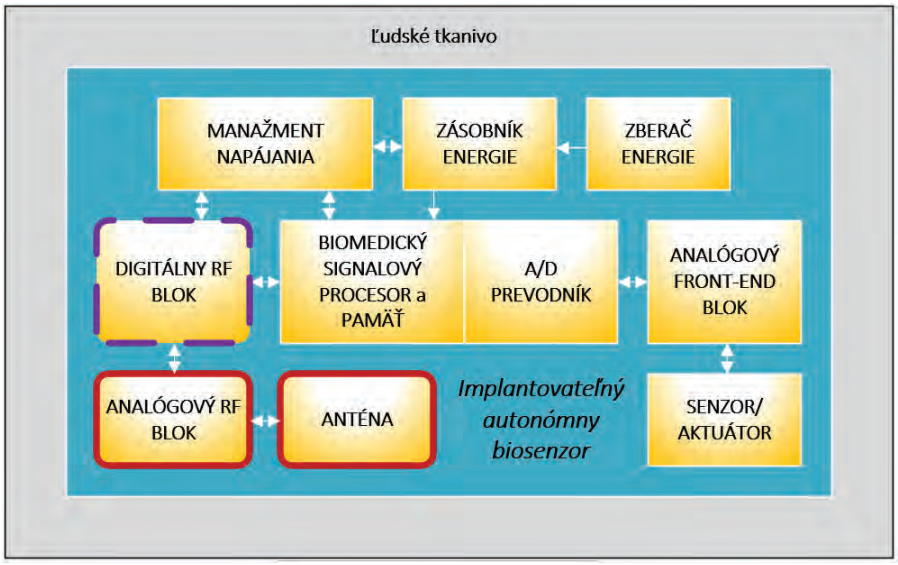
Na monitorovaní (stimulovaní) zdravotného stavu pacienta sa môže podieľať široké spektrum senzorov (stimulátorov), umožnujúcich kontinuálne monitorovanie (stimuláciu) chemických procesov a biosignálov v ľudskom organizme. Avšak nie všetky dôležité ukazovateľe zdravotného stavu je možné zachytiť len prostredíctvom senzorov implementovaných na ľudskom tele, respektíve mimo neho. Obzvlášť to platí v prípade stimulátorov, kde sa zväčša vyžaduje priamy kontakt stimulátora a stimulovaného objektu. Bežnými príkladmi sú aplikácie ako bio-resorbovateľný stimulátor pre termálnu terapiu, zariadenia určené pre stimuláciu diabetickej gastroparézy, stimulácia blúdivého nervu s cieľom redukcie srdcového infarktu, bezdrôtový kardiostimulátor, elektronické kapsule, atď. Takéto aplikácie tvoria výhradne uzavretý regulačný systém, ktorý je podporovaný existenciou snímacej jednotky v spätnej väzbe. Konkrétne príklady spolu s uvedenými zdrojmi demonštrujúce narastajúci význam IB je možné nájsť v [3].

Pre lepšie pochopenie uvažovaného konceptu IB systému uvádzame jeho všeobecnú blokovú schému (obr. 1), ktorá sa skladá zo štyroch základných častí:

- monitorovacia/stimulačná časť (senzor/aktuátor, analógový front-end blok),
- manažment elekrickej energie (zberač energie, zásobník energie, manažment napájania),
- modul spracovania dát (biomedicínsky signálový procesor a pamäť + A/D prevodník),
- komunikačný modul *(*anténa, analógový RF blok, digitálny RF blok).

Tri bloky vyznačené hrubými čiarami predstavujú hlavnú oblasť výskumu a vývoja pre dizertačnú prácu. Bloky ohraničené plnou čiarou patria z pohľadu plochy (anténa) a spotreby energie (analógový RF blok) medzi najkritickejšie časti (spolu so senzormi/aktuátormi, prípadne meničmi energie, zasobníkom elektrickej energie). Z toho dôvodu sa práve tieto dva bloky stávajú primárnym predmetom nášho výskumu, pričom otázka plochy a spotreby je stručne rozpracovaná v nasledujúcej časti príspevku. Digitálny RF blok (ohraničený prerušovanou čiarou) je taktiež súčasťou bezdrôtového komunikačného modulu a je rovnako zahrnutý už v spomínanom štandarde IEEE 802.15.6.2012. Preto považujeme za vhodné, aby bol v rámci dizertačnej práce vykonaný aj návrh tohto digitálneho bloku, aj keď z pohľadu spotreby energie a plochy čipu nepatrí medzi kritické časti. Jeho návrh bude teda našim sekundárnym cieľom.

Existujú štyri základné typy zberačov energie pre IB [3]. Ide o zberače energie založené na mechanickej (šírenie zvuku), elektromagnetickej (šírenie EM vĺn), mechanicko-kinetickej (vibrácie vyvolané prostredím) a chemickej energii. Pokiaľ sa v práci zameriame len na energeticky-autonómne IB, ktoré si dokážu samostatne generovať elektrickú energiu získanú premenou z prostredia, v ktorom sú umiestnené, v takom prípade môžeme uvažovať iba posledné dve zmieňované typy (nepotrebujú externý zdroj). V prípade glukózovo-kyslíkového bio-palivového článku, je maximálna plošná hustota výkonu okolo 200 $\mu W/cm^2$, zatiaľ čo zberač založený na kinetickej energii vibrácií môže dosiahnuť maximálnu plošnú hustotu výkonu okolo 56 $\mu W/cm^2$ [3]. Uvedené hodnoty sú dôležité z pohľadu celkového konceptu IB systému na čipe (angl. *System on Chip* - SoC) znázorneného na Obr. 3. IB vo forme SoC si vyžaduje integráciu antény, ktorej miniaturizácia je vo veľkej miere limitovaná aj konštrukciou ener-



Obr. 1: Bloková schéma implantovateľného biosenzorového systému.

getického zberača a výnimočne aj použitou mikrobatériou. Kapacita mikrobatérie, ktorá kombinuje najmodernejšie materiály elektród používané v Li batériach s 3D technologickým procesom na kremíkovej podložke, sa v súčasnosti pohybuje v rozsahu $1-5\ mAh/cm^2$. Táto hodnota závisí od hrúbky použitých plárnych elektród. Na druhej strane, napätie naprázdno závisí od použitého materiálu elektród a môže sa pohybovať v hodnotách od $1,5\ V$ od $4\ V$ [4]. Na ilustráciu parametrov mikrobatérie uvažujme konkrétny príklad prezentovaný v [5] a [6], kde bola analyzovaná 3D interdigitálna lítium-ionová batéria:

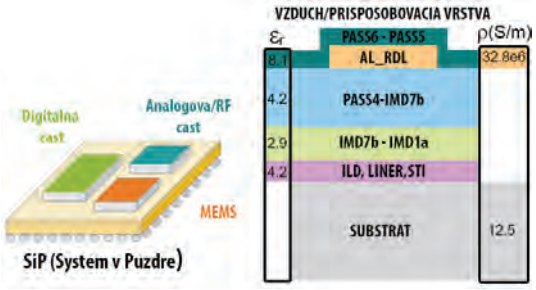
Podľa [5] predpokladajme, že spotreba impulzného UWB (angl. Impulse Radio UWB - IR UWB) príjmača/vysielača je $1\ nJ/pulz$ *(väčšina realizacií spadá pod túto hranicu), prenosová rýchlosť je* $1\ Mb/s$ *a jeden impulz predstavuje jeden bit. Ak vezmeme do úvahy údaje prezentované v [6], pre nominálne pracovné napätie mikrobatérie* $1,8\ V$ *je ekvivalentný statický odber približne* $556\ \mu A$*. To by znamenalo, že pri* $1,5\ mAh/cm^2$ *kapacite a uvažovanej maximálnej ploche mikrobatérie* $1,5 \times 1,5\ cm$ *by táto umožnila kontinuálny bezdrôtový prenos dát približne* 6 *hodín bez dobíjania.*

Je zrejmé, že pri takto zadefinovaných podmienkach, IB senzor bez mikrobatérie by nebol schopný kontinuálneho bezdrôtového prenosu dát. Avšak je nutné poznamenať, že IR-UWB vysielač zvyčajne preukazuje nižšiu spotrebu ako $1\ nJ/pulz$, ktorá bola v príklade uvažovaná. V takom prípade by mohlo byť kontinuálne vysielanie zabezpečené len pomocou energetického zberača. Netreba však zabúdať aj na ostatné funkčné bloky uvedené na Obr. 1. Predstavu o ploche a spotrebe energie procesnej jednotky nám ponúka publikácia [7], kde procesná jednotka spolu s analógovými snímacími obvodmi pre potenciálové a kapacitné biosenzory, a taktiež s implementovaným A/D prevodníkom je realizovaná na celkovej ploche len $1,95 \times 2,35\ mm$ (v $90\ nm$ CMOS technológii). Priemerná spotreba dosahovala hodnotu $46,6\ \mu W$ pre vypočtové jadro, $10,2\ \mu W$ pre potenciálový biosenzor, a $11,4\ \mu W$ pre kapacitný biosenzor. Keď zoberieme do úvahy plochu jednotlivých častí vrátane obvodu energetického manažmentu, prípadného budiaceho obvodu aktuátora a plochu maximálne do $2 \times 2\ mm$ (z prieskumu IR-UWB front-end časti komunikačného modulu v $90\ nm$ CMOS technológii), potom môžeme predpokladať, že IB bez vstavanej úložnej pamäte je možné realizovať na ploche do $5 \times 5\ mm$. Uvedený výsledok implikuje, že zvyšných $100\ mm^2$ plochy by mohlo byť rezervovaných pre statickú RAM pamäť, čo je dôležité pre zbezpečenie kontinuálneho monitorovania bez dlhodobej interakcie IB s externým čítačom. Technológia TSMC CLM90, v ktorej plánujeme výskum realizovať, disponuje tzv. vnorenou 6T-SRAM štandardnou bunkou s plochou $1,27\ \mu m^2$.

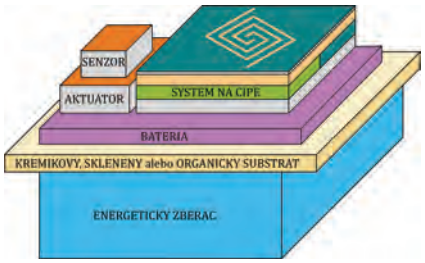
2.1 Návrh implementácie systému IB

Na obr. 2 je zobrazené klasické tzv. "side-by-side" 2D riešenie systému v puzdre (angl. *System in Package - SiP*) pre planárne štruktúry. Koncept SiP je nevyhnutný z hľadiska kompaktnosti IB a zároveň z hľadiska obmedzenej kompatibility s mikromechanickými štruktúrami, RF anténou a samotným štandardným CMOS procesom. Zvyčajne je každá časť spracovaná samostatne a následne sú všetky časti spojené do výsledného systému na spoločnej základovej podložke. Takto realizovaný SiP je však plošne neefektívny.

V našej práci preto uvažujeme vertikálny návrh 3D systému v puzdre [1] (obr. 3), kde analógová, digitálna a RF časť spolu s anténou tvorí jeden integrovaný 3D systém. Časti systému sú umiestnené priamo na batérii, pričom anténa je realizovaná na najvyššej metalizačnej úrovni, teda M9 alebo M8[2]. Pod anténou je potom umiestnená tieniaca vodivá vrstva eliminujúca interakciu elektromagnetického poľa antény so zvyškom SoC systému. Takto navrhnutý koncept poskytuje stále dostatočný počet metalizačných úrovní na realizáciu ostatných obvodových štruktúr celého systému a potrebných prepojení. Batéria je umiestnená na základovom substráte, ktorý bude zrejme tvorený nízko-teplotne vypaľovanou keramikou (tzv. LTCC) disponujúcou výbornými mechanickými a elektrickými vlastnosťami. Z druhej strany substrátu bude umiestnený zberač energie, ktorý je zvyčajne pomerne objemný. Mechanické časti (napr. aktuátor) zvyčajne nie sú súčasťou SoC, pretože požadujú priamu interakciu so stimulovaným objektom. Ich prispôsobovacie a riadiace obvody však môžu byť priamo súčasťou. Typickým príkladom je aktuátor na stimuláciu srdcovej činnosti, ktorý vyžaduje stimulačné napätie 5 V a vyššie, a tým pádom ho nie je možne implementovať v TSMC CLM90 technológii (nízko-napäťová technológia).



Obr. 2: Konvenčný SiP koncept spolu s radením TSMC CLM90 vrstiev pre HFSS simulátor.



Obr. 3: Koncept 3D systému pre IB

Takto navrhnutý 3D SoC koncept by umožnil nielen ušetriť predpokladanú plochu ($5 \times 5\ mm$), ale zároveň vytvoriť analógovú, digitálnu a RF časť IB systému v jednom výrobnom procese. Efektivita využitia plochy čipu však ostáva diskutabilná, pretože nie celú ušetrenú plochu je možné rezervovať pre SRAM pamäť. Dôvodom je existencia značného zvodového prúdu hradlom MOS tranzistora (tzv. *leakage current*), ktorý je typický pre submikrometrové technológie a jeho veľkosť môže byť až $400\ \mu A$ pre $8\ Mb$ pamäť [8]. Plocha štandardnej SRAM bunky je len $10,3\ mm^2$ a teda plocha $89,7\ mm^2$ by zostala stále nevyužitá (platí pre $90\ nm$ CMOS). Okrem toho parazitná kapacita, ktorá vzniká v dôsledku prídavnej tieniacej zeme môže významným spôsobom limitovať hornú hranicu frekvenčného pásma spracovaných signálov. Vrstva M8 (použitá na vyhotovenie antény) zase znehodnocuje modely induktora, ktoré predpokladajú jeho návrh v tejto vrstve. Dôsledkom toho je, že použitie EM simulátora na ich korekciu sa stane nevyhnutnou súčasťou druhej fázy výskumu spojeného s návrhom samotného IR-UWB prijímača/vysielača. Tým sa celkový návrh ešte viac skomplikuje.

Navrhovaný spôsob implementácie celého IB systému počíta s využitím UWB technológie, ktorá je v súčasnosti objektom mnohých výskumov, zaoberajúcich sa najmä charakterizáciou komunikačného kanála a vytvorením príslušného modelu pre IB, čo nám poskytuje dobrú východiskovú pozíciu. UWB

[1] Cieľom nie je poskytnúť techniku puzdrenia, ale iba ozrejmiť cieľ a predpokladaný prínos nášho výskumu.
[2] M8 vrstva je tzv. ultra hrubá kovová vrstva.

technológia je rovnako známa svojou nízkou spotrebou, vysokou prenosovou rýchlosťou, relatívnou jednoduchosťou IR-UWB vysielača, atď. [3]. Tiež je mnohými výskumníkmi označovaná ako nízkopríkonová alternatíva k existujúcim riešeniam, ktoré využívajú ISM (*Industrial, Scientific and Medical*) a MICS (*Medical Implant Communication Service*) pásma.

3 UWB anténa

V prvej fáze výskumu sa chceme zamerať na návrh samotnej UWB antény, ktorej vlastnosti v navrhnutom koncepte významným spôsobom ovplyvnia následné smerovanie výskumu, kde najmä plocha a zisk antény hrajú primárnu úlohu. Zvolili sme špirálovú anténu, ktorá i v prípade obdĺžnikového prierezu vykazuje širokopásmové vlastnosti. Toto je dôležitá vlastnosť antény, nakoľko zložitosť štruktúry antény je striktne daná návrhovými pravidlami zvolenej technológie, ktoré ju značne limitujú. Planárne UWB antény sú vo všeobecnosti tvorené práve plnými a nepravidelnými tvarmi, takže realizácia takýchto antén na čipe môže byť komplikovaná až nemožná. Špirálová anténa teda poskytuje lepšiu pravdepodobnosť vzájomnej kompatibility s návrhovými pravidlami danej technológie. Okrem toho má výhodný tzv. *form factor* - FF, prostredíctvom ktorého sme odvodili predpokladané rozmery čipu uvedené v sekcii 2. K tomu bola použitá nasledovná úvaha:

Predpokladajme, že vzdialenosť dvoch vedľa seba vedúcich vodičov je niekoľkokrát väčšia ako vzdialenosť medzi vrstvou antény a tieniacou vrstvou. Tieniaca vrstva sa nachádza v oblasti PASS4-IMD7b (obr. 3), kde sme uvažovali hodnotu reálnej časti komplexnej permitivity $4,2$*. Výsledkom je anténa s vlastnosťami mikropásikového vedenia s efektívnou permitivitou* $2,62$*. Aby podľa [9] došlo k efektívnej radiácii aj pre požadovanú spodnú hranicu frekvenčného pásma, musí platiť, že obvod tzv. Archimedeanovej špirálovej antény by mal byť* $1,25\ \lambda_{max}$*. Ak ešte zoberieme do úvahy tzv. "matching", vďaka ktorému v [10] dosiahli posuv centrálnej frekvencie k nižším frekvenciám približne o* $1,27$ *násobok pričom tiež došlo k rozšíreniu frekvenčného pásma, spodná frekvencia sa posunie z* $3,1\ GHz$ [3] *na* $3,937\ GHz$*. Táto frekvencia definuje* λ_{max} *rovné približne* $4,7\ cm$*. Aplikovaním FF, čiže* $(1,25/\pi)\lambda$ *pre kruhovú a* $(1,25/4)\lambda$ *pre štvorcovú Archimedeanovú anténu, dostáveme maximálny predpokladaný rozmer* $1,87\ cm$ *pre kruhovú anténu a* $1,47\ cm$ *pre štvorcovú anténu. A uvážime aj kontaktovacie plôšky (pady) a lem čipu, odhadovaná plocha čipu antény narastie na veľkosť* $1,5 \times 1,5\ cm$*.*

3.1 Dosiahnuté výsledky

Doterajší výskum bol venovaný návrhu 12-závitovej štvorcovej Archimedeanovej špirálovej antény, počas ktorého nebolo cieľom nájsť optimálny počet závitov a optimálne rozmery antény, ale sledovať trend vzťahu medzi rozmerom definujúcim veľkosť antény w_c a jej maximálnym ziskom G_{max}. Získané závislosti platné pre uvedený koncept sú na obr. 4 a obr. 5. Podrobný rozbor získaných výsledkov bude realizovaný počas prezentácie k príspevku, najmä v súvislosti s konceptom bez tieniacej zeme [3]. Stručne len skonštatujeme, že prítomnosť tieniacej vrstvy sa prejavila na menšom zisku (približne o $-30\ dB$) a podstatne menšej vstupej diferenciálnej impedancii. Je však evidentná lepšia uniformita jednotlivých charakteristík antény.

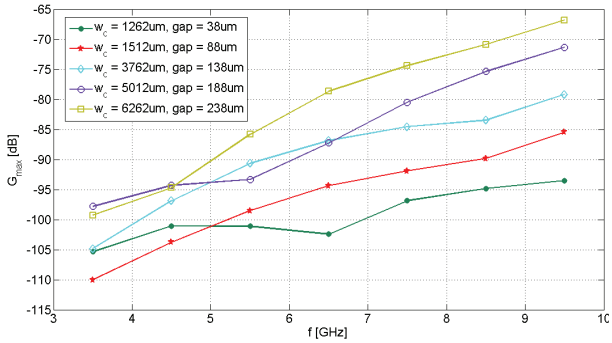
4 Ciele dizertačnej práce a záver

Zámerom dizertačnej práce je pojednať o možnosti integrácie UWB antény na čip spolu so zvyškom IB systému, s dôrazom na vyšetrenie jej zisku a vzájomnej kompatibility s návrhovými pravidlami štandardnej CMOS technológie. Druhá časť práce bude venovaná návrhu vybraných blokov systému, hlavne samotného nízkopríkonového IR-UWB príjmača/vysielača s variabilnou prenosovou rýchlosťou. Práca

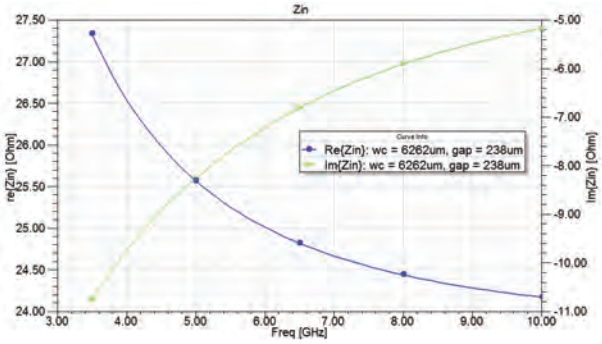
[3] UWB frekvenčné pásmo je komisiou FFC (angl. Federal Communications Commission) stanovené na $3,1 - 10,6\ GHz$

by tak mala poskytnúť kompletné originálne riešenie RF komunikačného modulu na báze UWB technológie, napomôcť k zefektívniu výrobného procesu a prispiesť k celkovej redukcii rozmerov a vylepšeniu vlastností energeticky-autonómneho systému bioimplantátu.

V rámci doterajšieho výskumu vzniklo spolu doteraz 5 publikácií, na ktorých som autorom resp. spoluautorom (2 články v karentovaných a impaktovaných vedeckých časopisoch a 3 príspevky na medzinárodných konferenciách).



Obr. 4: Maximálny zisk, pre rôzne hodnoty w_c (tzv. *return loss* nie sú uvažované).



Obr. 5: Vstupná impedancia pre rôzne hodnoty w_c.

Pod'akovanie

Tento príspevok vznikol vďaka podpore v rámci OP Výskum a vývoj pre projekt: Kompetenčné centrum inteligentných technológií pre elektronizáciu a informatizáciu systémov a služieb, ITMS: 26240220072, spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

Literatúra

[1] HORIZON 2020: AAL - Active and assisted living research and development programme, Article 185 (available 2.5.2014, http://www.welcomeurope.com/)

[2] den Bakker, W.; "Optimal Wireless Communication Method for Communication Inside the Human Body," (2013).

[3] Kováč, M.; Stopjaková, V.; Arbet D., "UWB communication for implantable biosensors within WBAN systems", Young Biomedical Engineers and Researchers Conference (YBERC), pp.6,11, 2-4 Jul. 2014

[4] Hahn, R., et al, "Development of Rechargeable Micro Batteries Based on Micro Channel Structures," Green Computing and Communications (GreenCom), 2012 IEEE International Conference on, pp.619,623, 20-23 Nov. 2012

[5] Fernandes, J.R.; Wentzloff, D., "Recent advances in IR-UWB transceivers: An overview," Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, pp.3284,3287, 2010

[6] Sun, K., et al, "3D Printing of Interdigitated Li–Ion Microbattery Architectures." Advanced Materials 25.33 (2013): 4539-4543

[7] Hsu, S.Y; et al, "A sub-100μW multi-functional cardiac signal processor for mobile healthcare applications," VLSI Circuits (VLSIC), 2012 Symposium on, pp.156,157, 13-15 June 2012

[8] Gerrish, P.; Herrmann, E.; Tyler, L.; Walsh, K., "Challenges and constraints in designing implantable medical ICs," Device and Materials Reliability, IEEE Transactions on , vol.5, no.3, pp.435,444, Sept. 2005

[9] Saynak, Uğur., "Novel rectangular spiral antennas," İzmir Institute of Technology: Electrical and Electronics Engineering, Thesis (Master),2007.

[10] Dissanayake, T.; et al,"Dielectric Loaded Impedance Matching for Wideband Implanted Antennas,"Microwave Theory and Techniques, IEEE Transactions on , vol.57, no.10, pp.2480,2487, Oct. 2009

Software Defined Monitoring: Nový prístup k monitorovaniu vysokorýchlostných počítačových sietí

Lukáš Kekely
Výpočetní technika a informatika, 1. ročník, prezenčná forma
Školiteľ: Jan Kořenek

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

ikekely@fit.vutbr.cz

Abstrakt. Neustále sa zvyšujúce rýchlosti liniek spolu s rastúcou významnosťou dát aplikačných protokolov pre monitorovanie vedú na nutnosť vytvoriť nový princíp hardvérovej akcelerácie spracovávania sieťových dát. V rámci dizertačnej práce *Softwarově řízené monitorování síťového provozu* je preto predstavený a skúmaný úplne nový koncept hardvérovej akcelerácie monitorovania sietí nazvaný *Software Defined Monitoring* (SDM). Základná myšlienka SDM je založená na úzkom prepojení softvérových monitorovacích aplikácií s výkonným hardvérovým akcelerátorom, ktorý predspracúva sieťové dáta. Softvérové aplikácie pritom môžu jednoducho ovládať stupeň detailov zachovávaných predspracovaním pre jednotlivé sieťové toky. Vďaka tomu je možné spracovanie menej zaujímavých dát prenechať akcelerátoru a v softvéri sa zamerať už len na podrobné spracovanie naozaj zaujímavých dát. Tým SDM umožňuje prakticky realizovať flexibilné monitorovanie s podporou podrobnej analýzy paketov aj na veľmi vysokých rýchlostiach – až 100 Gb/s.

Klúčové slová. FPGA, akcelerácia, monitorovanie, bezpečnosť, vysokorýchlostné siete

1 Úvod

Monitorovanie sieťových dát hrá jednu z kľúčových úloh pre oblasti správy a bezpečnosti moderných počítačových sietí. Dnes zaužívaným štandardom pre monitorovanie sietí je meranie na bázy sieťových tokov. Monitorovacie zariadenia zbierajú základné štatistiky o všetkých paketoch a agregujú ich do záznamov o tokoch. Tie zasielajú na centrálne úložisko (kolektor) pomocou protokolu NetFlow [1] alebo IPFIX [2]. V procese zberu a agregovania dát tak dochádza k istej strate informácií a kolektor (kde sa dáta ďalej analyzujú) má preto obmedzený pohľad na sieť. Z uvedeného dôvodu je aktuálnym trendom snaha rozširovať záznamy o tokoch pridaním nejakej informáciu navyše k základným veľkostným a časovým štatistikám. Pridaná informácia pritom často býva založená na dátach z aplikačných protokolov.

Implementáciu monitorovania obohateného o analýzu aplikačných protokolov je možné celú vytvoriť v softvéri. Priepustnosť takejto realizácie je však silne obmedzená výkonnosťou súčasných procesorov. Na druhej strane, čisto hardvérové riešenia majú slabú flexibilitu, z dôvodu náročnej hardvérovej realizácie komplexných analyzátorov aplikačných protokolov. Navyše nové bezpečnostné hrozby nestále vznikajú a je potrebné na ne dostatočne rýchlo reagovať, čo je pre hardvérové riešenia problémové. Uvedené zhodnotenie dvoch základných prístupov vedie na ideu vytvoriť niečo medzi, teda výkonný hardvérový akcelerátor spracovania dát plne kontrolovaný flexibilnými softvérovými aplikáciami. Práve softvérovému riadeniu vďačí navrhnutý koncept za označenie *Software Defined Monitoring* (SDM).

Úloha hardvérového akcelerátoru v SDM spočíva v redukcii objemu dát tečúcich k softvérovým aplikáciám tým, že nad zvolenými časťami dát realizuje analýzu hlavičiek paketov a prípadne aj ich agregovanie do tokov. Akcelerátor tak posiela zaujímavú časť paketov nedotknutých do softvéru na precíznejšiu analýzu, zatiaľ čo sám realizuje základné meranie na bázy tokov nad zvyškom dát. Navyše je podporované aj filtrovanie pre prípad, že aplikácie nepotrebujú agregované informácie o všetkých paketoch.

Výber spôsobu spracovania jednotlivých paketov v akcelerátore SDM je plne kontrolovaný monitorovacím softvérom a môže byť za behu prispôsobovaný aktuálnym potrebám konkrétnej aplikácie. Realizovaný je pomocou dynamicky sa meniacej množiny pravidiel nad sieťovými tokmi vytváranej aplikáciou na základe pozorovaných paketov. Uvedené pravidlá sú do akcelerátoru nahrávané jednotným rozhraním a každé určuje ako predspracovať ďalšie príchodzie pakety jedného konkrétneho toku. Vďaka jednotnosti ovládacieho rozhrania akcelerácie je systém flexibilný a je možné ho použiť na zvýšenie výkonnosti širokého spektra rôznych monitorovacích a bezpečnostných aplikácií.

Prínos práce prezentovanej v tomto príspevku je v troch oblastiach: (1) analýza dát z reálnej vysokorýchlostnej siete s ohľadom na rozhodnutie o vhodnosti akcelerácie založenej na popísanom koncepte SDM (sekcia 2); (2) rozpracovanie návrhu konceptu SDM pre vysokorýchlostné siete, čo zahŕňa návrh hardvéru (aplikačne špecifický procesor) aj jeho riadiaceho softvéru (sekcia 3); (3) implementácia a vyhodnotenie vlastností systému v niekoľkých prípadoch použitia (sekcia 4).

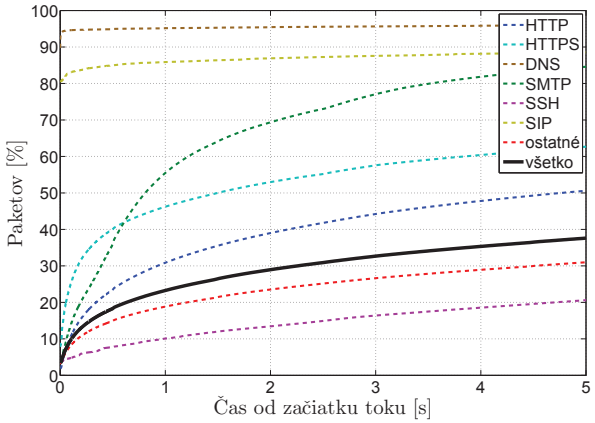
2 Analýza

Začiatok príspevku sa venuje analýze vlastností sieťových dát na reálnej vysokorýchlostnej sieti. Na základe zmeraných charakteristík je následne vytvorený podrobný návrh SDM systému tak, aby dosahoval optimálnu výkonnosť v reálnom nasadení. Všetky merania uvedené v celom príspevku sú realizované vo vysokorýchlostnej sieti CESNET2, ktorá má optické linky pracujúce na rýchlostiach do 100 Gb/s.

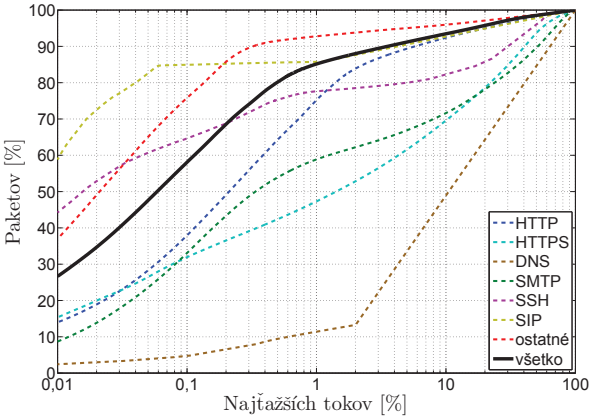
Pretože softvérové aplikácie rozhodujú o predspracovaní dát je časovanie príchodu paketov veľmi dôležité z pohľadu dosiahnuteľnej výkonnosti. Najlepší pohľad na časovanie paketov v tokoch je možné získať meraním relatívneho času príchodu paketov od začiatku toku. Čiže, prvý paket každého toku má nulový relatívny čas príchodu a jeho absolútny čas označuje moment začiatku toku. Potom relatívny čas príchodu každý nasledujúceho paketu je rozdiel absolútneho času jeho príchodu a poznačeného momentu začiatku toku. Výsledky popísaného merania sú zanesené v grafe na Obr. 1 , ktorý zobrazuje distribučné funkcie práve relatívnych časov príchodu paketov pre rôzne skupiny dát. Graf ukazuje, že všeobecne (čierna plná čiara) len malá časť paketov príde hneď po začatí toku (napr. len asi pätina paketov príde počas prvej sekundy tokov). To znamená, že aj ak bude oneskorenie softvérového riadenia pri zavádzaní pravidiel o tokoch relatívne vysoké, stále umožní pravidlami ovplyvniť spracovanie väčšiny paketov.

Ďalšou dôležitou vlastnosťou sieťových dát je charakter rozdelenia veľkostí tokov. Z grafu na Obr. 2 vidno, že podľa merania má distribúcia veľkostí tokov na reálnej sieti heavy-tailed charakter. Uvedený graf ukazuje podiel paketov prenesených istým percentom najťažších tokov. Je teda všeobecne (čierna plná čiara) vidno, že aj veľmi malé percento najťažších tokov prenáša významnú časť celkového počtu paketov (napr. 1 % tokov nesie až 85 % paketov). Z pohľadu navrhnutého SDM je tak možné aj zavedením len malého počtu pravidiel o tokoch zaistiť akceleráciu predspracovania väčšiny paketov.

Pre praktické využitie heavy-tailed charakteru v prospech výkonnosti SDM je ešte potrebné vyriešiť problém vhodného rozpoznania najťažších tokov. Presnejšie je problém definovaný ako schopnosť predpovedať, ktoré toky sú z najťažších len na základe pozorovania istých vlastností ich prvých paketov. Na riešenie uvedeného problému je možné použiť veľmi priamočiaru metódu: pre zvolenú hodnotu parametra k označ za ťažký tok každý taký, o ktorom je už známe, že má aspoň k paketov. Výhodou tejto jednoduchej metódy je nenáročnosť jej implementácie, pretože jedinou sledovanou vlastnosťou paketov je ich samotná existencia (netreba ich dodatočne analyzovať). Pritom aj takto jednoduchá metóda vedie na veľmi dobré výsledky rozpoznania ťažkých tokov z pohľadu konceptu SDM, ako je ukázané na grafoch 5 a 6 v sekcii s rozborom dosahovanej výkonnosti.



Obr. 1: Časovanie príchodu paketov v tokoch



Obr. 2: Heavy-tailed charakter dát

3 Architektúra

Ako už je spomenuté v úvode, základná myšlienka akcelerácie v SDM systéme spočíva v jemne kontrolovanej redukcii objemu dát dosiahnutej akcelerovaným predspracovaním paketov zo siete. Predspracovanie samotné je realizované v hardvéri, ale jeho použitie je plne kontrolované softvérovými aplikáciami. Práve preto, je niekoľko počiatočných paketov každého toku poslaných do softvéru, ktorý až na ich základe vyberie spôsob hardvérového predspracovania nasledujúcich paketov daného toku. Vhodné typy podporovaného predspracovania paketov pre oblasť monitorovania je možné rozdeliť do troch skupín:

Extrakcia zaujímavých informácií z paketov a posielanie len týchto informácií do softvéru v jednotnom formáte (unifikovaná hlavička - UH). Tým sa zníži jednak objem dát poslaných do softvéru, ale aj vyťaženie procesoru, pretože analýzu paketov realizoval už hardvér.

Agregovanie dát z paketov do záznamov o tokoch priamo v hardvéri vedúce na ešte vyššiu úsporu výkonnosti softvéru. Môžu pritom existovať rôzne spôsoby agregovania pre rôzne aplikácie.

Filtrovanie úplne nepotrebných paketov, čo môže napomôcť rôznym aplikáciám zameraným na pokročilú analýzu špecifickej podskupiny sieťových dát (napr. analyzátor HTTP).

Základnú konceptuálnu schému navrhnutého systému SDM je možné vidieť na Obr. 3. Dáta nesúce cesty sú značené plnými šípkami a kontrolné spätné väzby prerušovanými. Systém je zložený z dvoch častí (firmvér FPGA a softvér) prepojených dátovou zbernicou (napr. PCI Express). Dáta z firmvéru do softvéru prichádzajú po viacerých nezávislých kanáloch a to vo forme celých paketov, UH alebo záznamov o tokoch. Tieto dáta sú potom spracúvané užívateľom definovanou množinou monitorovacích a bezpečnostných aplikácií (napr. exportér tokov). Aplikácie, vo forme SDM zásuvných modulov, okrem čítania dát z vybraných kanálov môžu špecifikovať, ktoré toky sú pre ne nezaujímavé a môžu sa tak spracúvať vo firmvéri. Definície nezaujímavých tokov od všetkých aplikácií sú agregované v SDM radiči, ktorý na základe nich priamo konfiguruje predspracovanie vo firmvéri so snahou dosiahnuť maximálnu redukciu dát pri zachovaní dostatočnej úrovne detailov. SDM radič je tak jediným kontrolným prvkom celého systému, ktorý priamo riadi správanie sa firmvéru.

SDM firmvér začne spracovanie každého paketu jeho analýzou a extrakciou zaujímavých dát. Na základe extrahovaných dát a množiny pravidiel nakonfigurovaných od SDM radiča potom rozhodne o konkrétnom spôsobe predspracovania tohto paketu aj o smerovaní dát pre softvér do správneho kanálu. Podrobnejšie je možné spôsob realizácie akceleračné firmvéru SDM vidieť na Obr. 4. Popísané spracovanie paketov realizuje procesná zreťazená linka štyroch jednotiek. Dáta paketov pritom netečú priamo touto linkou, ale sú odložené v paralelnej FIFO pamäti. Celá konfigurácia procesnej linky je realizovaná cez špeciálnu jednotku, ktorá vie atomicky spravovať pravidlá priamo v pamäti vyhľadávacej jednotky. SDM firmvér je teda realizovaný piatimi jednotkami:

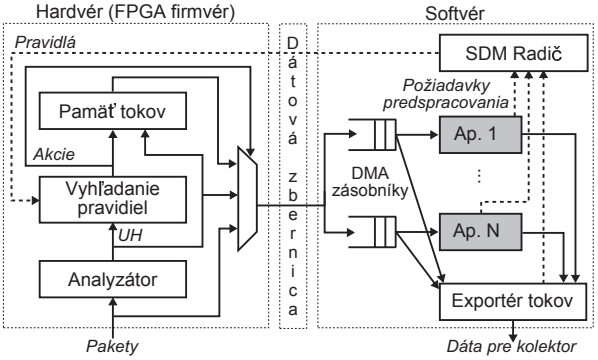
Analyzátor extrahuje zaujímavé informácie z hlavičiek paketov, najmä polia identifikujúce sieťový tok (IP adresy, čísla portov a protokol). Navyše je štruktúra analyzátora modulárna a umožňuje jednoduché pridanie ďalších analyzačných modulov (A1..An). Podrobnejšie analyzátor popisujem v [3, 4].

Hľadanie pravidiel s cieľom prideliť akciu (inštrukciu spracovania) každému paketu na základe identifikátoru toku a množiny softvérom nakonfigurovaných pravidiel. Efektívna implementácia je možná napríklad špeciálnou haš tabuľkou s kukučím hašovaním ako ukazujem v [5]. Ku tomu potrebné haš funkcie je ďalej možné v FPGA efektívne realizovať pomocou CRC ako uvádzam v [6, 7].

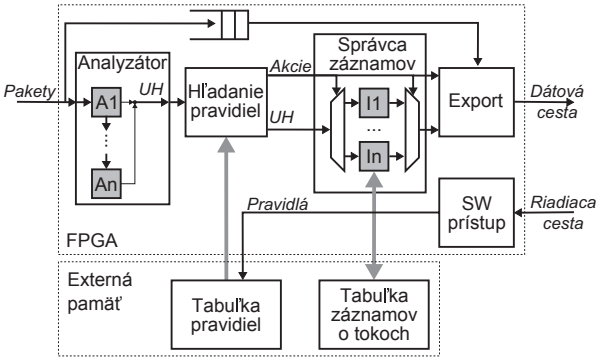
Správca záznamov spravuje stavové záznamy v tabuľke tokov. Stará sa hlavne o aktualizáciu ich hodnôt pomocou inštrukcií podľa paketom patriacich akcií. Každá akcia nesie okrem inštrukčného kódu aj adresu záznamu, na ktorú sa má aplikovať. Pri aktualizácii inštrukcie pracujú s aktuálnou hodnotou záznamu z pamäte aj s dátami z UH. Okrem aktualizačných inštrukcií podporuje jednotka aj špeciálnu inštrukciu exportovania (a nulovania) zvoleného záznamu, ktorá je iniciovaná na konci toku alebo v pravidelných intervaloch. Správcu záznamov je možné jednoducho rozširovať o nové inštrukčné moduly (I1..In). Túto problematiku podrobnejšie rozoberám napríklad v [8].

Export sa stará o smerovanie dát v správnom formáte a správnym softvérovým kanálom.

SW prístup je hlavným prístupovým bodom k SDM firmvéru zo strany softvéru. Zaisťuje správu pravidiel o tokoch a iniciuje export záznamov o tokoch na základe príkazov od SDM radiča.



Obr. 3: Konceptuálna schéma SDM systému

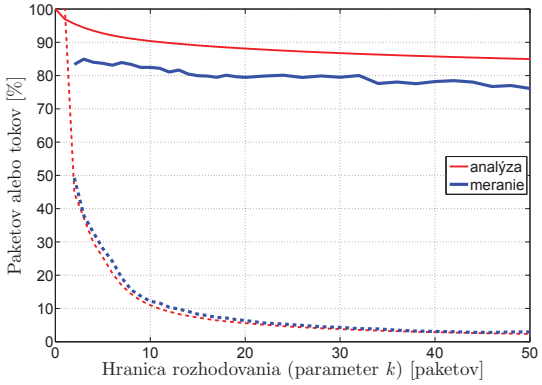


Obr. 4: Podrobnejšia schéma SDM firmvéru

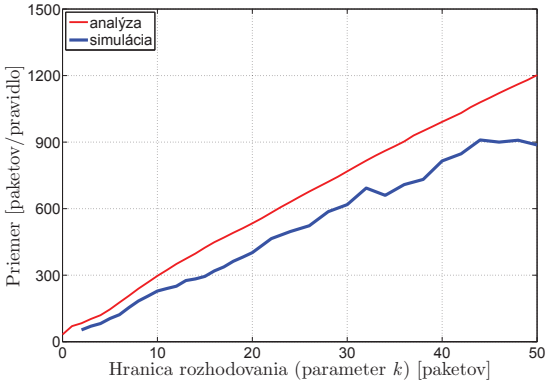
4 Výsledky

Navrhnutý SDM systém je implementovaný. Pritom je použitá akceleračná PCI Express karta s FPGA čipom rodiny Virtex-7 v troch variantoch sieťových rozhraní: 8×10 GbE, 2×40 GbE a 1×100GbE. Vo všetkých troch prípadoch pripadá na samotné funkčne jadro SDM len necelá štvrtina zdrojov firmvéru, ktorý celkovo zaberá necelú polovicu zdrojov čipu. Výkonnosť vytvorenej realizácie SDM je ďalej otestovaná s ohľadom na dosiahnuteľný stupeň akcelerácie.

Prvým testom je meranie percenta paketov, ktoré je SDM firmvér schopný spracovať na základe softvérom za behu vytvorených pravidiel o tokoch. Využité je pri tom pravidlo rozpoznania ťažkých tokov predstavené na konci sekcie 2. Výsledky testu sú zanesené v grafe na Obr. 5, ktorý ukazuje závislosti medzi hodnotou parametru k a časťou tokov považovaných za ťažké (prerušovaná čiara) a paketov ich výberom pokrytých (plná čiara). Je vidno, že s rastúcou hranicou rozhodovania rapídne klesá podiel vybraný tokov, ale podiel nimi pokrytých paketov klesá len pozvoľna. To vedie na rast priemerného počtu paketov pokrytých jedným pravidlom (zisk zo zavedenia pravidla), ako ukazuje aj graf na Obr. 6. V grafoch tiež vidno rozdiel medzi analyticky a meraním zistenou efektivitou systému. Rozdiel (5 až 10 % paketov) je spôsobený istým časovým oneskorením zavádzania pravidiel ako reakcie na prvé pakety toku v reálnom systéme, ktoré nie je pri analytickom vyhodnotení brané do úvahy.



Obr. 5: Zachytené percento paketov alebo tokov



Obr. 6: Počet zachytených paketov na pravidlo

Ďalšie testy efektivity akcelerácie SDM sú realizované pre reálnejšie prípady nasadenia systému a ich výsledky sú zanesené v tabuľke 1. Testované je nasadenie SDM na akceleráciu piatich rôznych prípadov: (1) základné NetFlow monitorovanie tokov [1], (2) detektor skenovania portov, (3) detektor útoku Hartbleed na HTTPS protokol, (4) podrobná analýza aplikačného protokolu HTTP a (5) základné monitorovanie tokov obohatené o podrobnú analýzu HTTP. Hodnoty zanesené do tabuľky sú dvojakého typu – podiel využitia podporovaných typov hardvérového predspracovania a objem redukovaného dátového toku do softvéru v jednotlivých prípadoch nasadenia. Všeobecne vidno, že aplikácie zamerané na podrobnejšiu analýzu špecifických dát (2, 3, 4) využívajú hlavne filtrovanie. Naproti tomu, aplikácie vyžadujúce štatistické informácie o všetkých paketoch na linke (1) využívajú hlavne agregovanie. Nakoniec aplikácie nepracujúce priamo s dátami paketov (1, 2) používajú do istej miery aj extrakciu. Z posledných dvoch stĺpcov tabuľky vidno, že dosiahnutá redukcia záťaže softvéru oproti prípadu bez použitia SDM je relatívne vysoká – väčšinou ide o redukciu počtu paketov aspoň päťkrát a bajtov ešte viac.

Prípad použitia	**HW akcia [% paketov]**				**HW akcia [% bajtov]**				**SW záťaž [%]**	
	∅	**Ex**	**Ag**	**Fi**	∅	**Ex**	**Ag**	**Fi**	**Paketov**	**Bajtov**
NetFlow	–	20.55	79.45	–	–	12.03	87.97	–	20.66	0.98
Port sken	–	17.54	–	82.46	–	10.35	–	89.65	17.54	0.86
Heartbleed	4.91	–	–	95.09	3.77	–	–	96.23	4.91	3.77
HTTP	22.82	–	–	77.18	27.82	–	–	72.18	22.82	27.82
HTTP+NetFlow	23.34	10.56	66.10	–	28.50	3.63	68.87	–	34.02	29.00

Tabuľka 1: Využitie podporovaných typov hardvérového predspracovania v rôznych prípadoch použitia

5 Stav a ciele dizertačnej práce

Príspevok predstavil súčasný trend zvyšovania prenosových rýchlostí v počítačových sieťach vedúci na nutnosť výkonnejších monitorovacích a bezpečnostných systémov. Práve touto oblasťou sa zaoberám v rámci dizertačnej práce, kde som navrhol realizoval a základne testoval práve popísaný unikátny koncept flexibilnej akcelerácie monitorovania označený SDM. Zatiaľ čo konkurenčné postupy akcelerácie monitorovania sa spoliehajú buď na čisto hardvérové riešenia, ktorým chýba flexibilita alebo na čisto softvérové riešenia, ktorým zase chýba výkonnosť, predstavený koncept SDM predstavuje cestu vhodného spojenia hardvéru a softvéru pri zachovaní ich výhod a limitovaní ich nedostatkov. Základný koncept SDM ako je popísaný v tomto príspevku bol už publikovaný na IEEE konferencii INFOCOM [9] a prezentovaný na viacerých sieťových konferenciách (napr. *IETF Meeting* či *TERENA Networking*

Conference). Okrem toho boli publikované aj riešenia viacerých špecifických častí systému, ako sú odkazované priamo z textu príspevku. Aktuálne sa tiež o SDM pripravuje článok na vyžiadanie do časopisu *IEEE Transactions on Computers*. Prototyp systému je taktiež aktuálne v testovacom režime nasadený na produkčnej siete združenia CESNET a očakáva sa jeho skoré produkčné nasadenie. O SDM prejavila záujem aj komerčná firma Invea-Tech, ktorá ho chce zaradiť do svojho portfólia produktov.

V rámci ďalšieho smerovania dizertačnej práce sa chcem v priebehu druhého ročníka zaoberať hlavne experimentmi s akceleráciou rôznych aplikácií z oblasti monitorovania a bezpečnosti pomocou SDM na produkčnej sieti ako aj jeho ďalším rozširovaním a vylepšovaním. Pričom výsledky tohto snaženia plánujem priebežne publikovať. Nakoniec v treťom ročníku by som sa zameral na skonsolidovanie všetkých získaných výsledkov a spísanie finálneho textu dizertačnej práce.

6 Záver

Príspevok ukazuje návrh a implementáciu nového konceptu (systému) flexibilnej akcelerácie monitorovania vysokorýchlostných počítačových sietí. Uvádza tiež vybrané výsledky analýzy a testovania výkonnosti na dátach z reálnej siete, ktoré ukazujú, že vytvorený systém je schopný napomôcť monitorovaniu aplikačných protokolov na rýchlostiach liniek až do 100 Gb/s. Prezentované výsledky sú dosiahnuté v rámci dizertačnej práce na tému *Softwarově řízené monitorování síťového provozu*, ktorej ďalším pokračovaním bude prehlbovanie experimentálnych výsledkov z nasadenia na reálnej sieti a ďalšie vylepšovanie vlastností predstaveného konceptu SDM.

Poďakovanie

Príspevok vznikol čiastočne za podpory projektu VUT v Brne FIT-S-14-2297, projektu Centra excelencie IT4Innovations CZ.1.05/1.1.00/02.0070 a výskumného zámeru MSM 0021630528. Prezentovaná práca je tiež súčasťou projektu MŠMT "Velká infrastruktura CESNET" s číslom LM2010005.

Literatúra

[1] B. Claise: Cisco Systems NetFlow Services Export Version 9, RFC 3954, IETF, 2004

[2] B. Claise, B. Trammell, and P. Aitken: Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information, RFC 7011, IETF, 2013

[3] L. Kekely, V. Puš and J. Kořenek: Design Methodology of Configurable High Performance Packet Parser for FPGA, International Symposium on Design and Diagnostics of Electronic Circuits & Systems, IEEE, 2014

[4] L. Kekely, V. Puš and J. Kořenek: Low-Latency Modular Packet Header Parser for FPGA, Symposium on Architectures for Networking and Communications Systems, ACM, 2012, ISBN 978-1-4503-1685-9

[5] L. Kekely, M. Žádník, J. Matoušek and J. Kořenek: Fast Lookup for Dynamic Packet Filtering in FPGA, International Symposium on Design and Diagnostics of Electronic Circuits & Systems, IEEE, 2014, ISBN 978-1-4799-4558-0

[6] L. Kekely, T. Závodník and V. Puš: CRC based hashing in FPGA using DSP blocks, International Symposium on Design and Diagnostics of Electronic Circuits & Systems, IEEE, 2014

[7] L. Kekely, T. Závodník and V. Puš: Using DSP blocks to compute CRC hash in FPGA, International Symposium on Field-Programmable Gate Arrays, ACM, 2014, ISBN 978-1-4503-2671-1

[8] L. Kekely, V. Puš, P. Benáček and J. Kořenek: Trade-offs and Progressive Adoption of FPGA Acceleration in Network Traffic Monitoring, International Conference on Field Programmable Logic and Applications, IEEE, 2014

[9] L. Kekely, V. Puš, and J. Kořenek: Software Defined Monitoring of Application Protocols, The 33rd Annual IEEE International Conference on Computer Communications, IEEE, 2014, ISBN 978-1-4799-3360-0

Case Study: Comparison of various approaches in Fault-Tolerant and Attack-Resistant system design

Filip Štěpánek
Informatics, 1-st class, full-time study
Supervisor: Petr Fišer, Martin Novotný

Faculty of Information Technology, Czech Technical University in Prague
Thákurova 9, 160 00 Prague, Czech Republic

filip.stepanek@fit.cvut.cz

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

Keywords. Fault tolerance, attack resistance, FPGA, system design, system optimization

1 Introduction

Fault tolerant systems find many application areas like traffic control systems, where any kind of fault (more or less of accidental nature) could result in system unavailability or failure that could lead to an accident, damage to the property, or loss of lives. These systems emphasize their correct behaviour in environments where the probability of fault generation is high and where the possible faults would result in undesirable system failure. To counter this possibility, fault tolerant systems implement features to increase their safety and reliability [1].

The attack resistant systems on the contrary protect their content from targeted attacks by human individuals who observe the system behaviour for the purpose of finding weaknesses in the system to take advantage of or who try to read the processed data from memories or buses to find secrets that could be later misused to their advantage.

Therefore each of these systems implement their own measures to operate in a given desired environment according to their specification. Fault tolerant systems with some kind of attack resistant features are rare, but examples can be seen in form of digital storage media (CDs/DVDs/Blue-Ray discs) that use ECC (Error-Correction-Codes) to eliminate faults due to the scratches on the surface or due to some other form of possible data corruption of physical origin. Some of the digital storage media may be

copy-protected, thus they implement some form of DRM (Digital rights management) to protect their content. But it is apparent that even though the digital storage media may implement both features of fault tolerant and attack resistant systems, they can hardly be called safety-critical systems (e.g., systems where failure of such a system could lead to disastrous consequences).

1.1 Should fault tolerant systems be attack resistant?

From the experience in the field of software attacks, the attacker in the 80's was a professional who completely understood the computer systems he was attacking. This kind of people were rare and it can be said that their motivation was self-education and not committing the criminal activity. Nowadays the number of attacks is high and the education or the necessary skills of the attackers are lower than before. Among the factors influencing this phenomena are all kinds of tools, tutorials or security weaknesses publicly available online, so all kinds of script kiddies are able to "play" hackers.

Although this involves mostly software systems, the question is "When will embedded systems become a target of malicious attacks?" The truth is, that to attack the embedded system, the attacker needs some kind of equipment – ordinary PC is often not enough for this kind of job. But there is special-purpose hardware available like the COPACOBANA (Cost-Optimized Parallel COde Breaker) [2] and as shown in [3], the prices of attacks on embedded systems can be kept low.

2 System design

When developing an embedded system, the designers have various resources at their disposal like allocated time and effort to produce a device of desired quality and functionality. The produced device is often optimized to suit its specifications and operation environment [4], [5], [6], [7].

To show some dependencies and difficulties in achieving both the fault tolerant and attack resistant system, a triangle illustration is proposed as shown in Figure 1. The proposal is based on the optimisation of the implementation, where the design is being optimised for higher speed (Time), smaller area (Area) or lower power consumption (Power).

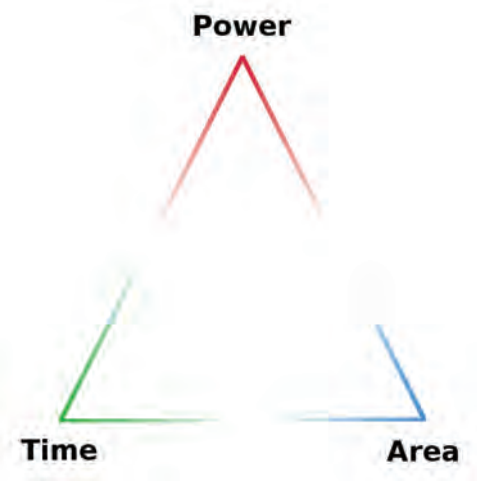


Figure 1: Optimization of the design implementation using three equally important vertices (Area, Time, Power).

As mentioned, the emphasized parameters (vertices) often depend on the specification of the system and its deployment. For example the time-critical systems emphasize the speed of the operation, the low

power devices minimize the power consumption and the area-emphasized systems require the design to be as small as possible in order to fit the given area-constrains.

Needless to say that in ordinary consumer (low-cost) products often no one implements fault tolerant or attack resistant measures as fault tolerance is not required and attack resistance would cost more time and effort to implement (see examples like hacking the baby monitors [8] or even sensitive military equipment [9]). But there are also products that require attack resistance and developers spend time to implement it – examples of such systems include payment cards or other devices that deal with someone's money or privacy. Other examples represent counter-piracy measures or devices that strongly fight tempering like the Xbox gaming platform [10].

2.1 Fault tolerant design

Fault tolerant design aims at ensuring the correct behaviour of the system in unreliable environment. The fault tolerant systems must be able to detect incorrect behaviour (and if desired to correct it). Such a detection is carried out via monitoring and observing the system state during its operation (a feature that is not very welcome in the field of attack resistant systems, as it may reveal information to the attacker). To increase the system reliability (e.g., its fault tolerance), the system implements some sort of redundancy (e.g., area/time/information redundancy).

The area redundancy can be in form of module replication (e.g., TMR – triple modular redundancy) that masks the possible faults using majority function. Time redundancy does not duplicate any physical modules, but instead duplicates the operation (i.e., the device sends the same information many times or does the calculation repeatedly to ensure the correct result). The last is the so-called information redundancy – this type of redundancy adds some other information (information bits) to the data to enable checking its consistency. The information redundancy can be seen in form of ECC (error correction codes) like parity checking, linear codes and cyclic codes.

From the system design point of view the fault tolerance is mostly paid by the area (size – physical redundancy) of the device or time (computation time – time redundancy). In the triangle illustration in Figure 2 it should be regarded as the cost of the fault tolerance of the device. In other words – in case a low-latency system is desired, it cannot implement time redundancy and in case of a system with strict area requirements (e.g., minimal area overhead), the system cannot implement physical redundancy. The power consumption does not seem to be a "hot topic" in the field of fault tolerant systems as the minimal power consumption is not emphasized in systems where fault tolerance is priority. For example it is reasonable to assume, that the implementation of TMR would add some FT features but on the other hand would increase the power consumption of the device. Therefore optimising/minimising the power consumption of the FT device does not seem to be reasonable.

2.2 Attack resistant design

Attack resistant design protects the system from a malicious tempering by criminal individuals. Purpose of the attack resistant systems is to hide the processed or stored data. To achieve this goal the attack resistant systems implement some kind of cryptographic scheme (e.g., encryption/decryption algorithms) so the stored data or the eavesdropped communication is not decoded by the attacker. These systems must implement the attack resistance on multiple levels, e.g., on the lowest levels the designers implement temper-resistant packages, memories, etc. And on the higher levels (software implementation levels) they implement encryption/decryption algorithms to secure the processed data (communication) while making the device itself temper-resistant to prevent reverse engineering or physical attacks. Some of the attacks use the so-called side channels like exploiting the data dependency on the power trace to break

Figure 2: Comparison of system design approaches (vertices of interest) in the field of fault tolerant and attack resistant systems. On the left the triangle illustration of the fault tolerant system shows the need to implement redundancy by using area and time resources of the system to add some fault tolerant properties. The triangle illustration on the right on the contrary points that in the field of attack resistant systems the power consumption is discussed a lot as it can reveal secret information to the attacker.

the encryption. To counter this threat, the designers use special techniques to mask/hide the processed data from the power trace [11].

Therefore, the power consumption is quite often discussed feature of the attack resistant systems, as high variances in the power consumption make the power analysis attacks more feasible. The triangle illustration in Figure 2 shows that unlike the FT systems, the AR systems pay a lot of attention to optimizing the power trace produced by the device. Although this does not mean they want to make it minimal, the designers need to optimize it in order to make the power attacks impractical.

2.3 Example of fault tolerant and attack resistant system

Figure 3 describes the proposed architecture of the fault tolerant and attack resistant system by an example. In this case there is an automated train control system (a proposed project for the Prague subway), where the control data are sent through wireless channel. The communication must be encrypted to ensure the system security. Of course the protocol itself must be designed in a way to make eavesdropping of the communication and later replay attacks useless. Also the device must be secured against physical attacks (temper resistant), as the module might be acquired by some unauthorized means (e.g., stolen from the depot, schematics or documentation might leak outside, etc.). Another problem might be in the operational expectancy of the device. To operate in such a system like public transport, every module (even the encryption/decryption module) must fulfil the requirements of local safety specifications and regulations [12]. Also as is common in the field of fault tolerant systems, the module might be in use for a long time (decades) without changes to its settings or design, which places high demands on strong key management.

3 Evaluation of the attack-resistance using the DPA

The evaluation of the attack resistance of the system uses a set of known attacks, that is calculated using the necessary time and cost of the equipment (tools, lab, computational power) required for the successful attack. In other words – the higher the price of the attack, the more secure the system is. For the purpose of the PhD thesis, the DPA (differential power analysis) will be used to evaluate the attack resistance

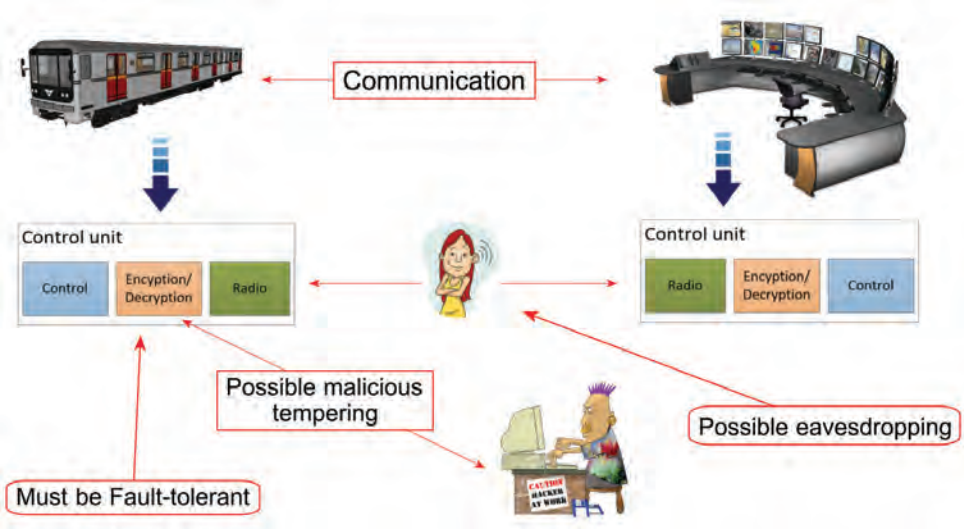


Figure 3: Illustration of usage of proposed encryption/decryption unit for the Prague subway.

of various fault tolerant implementations, as it is regarded as a low-cost and easy to implement attack, that exploits the dependency of the power trace on the processed data. Furthermore, it is a non-invasive technique that does not place high demands on the necessary equipment and the skill of the attacker [13]. Modern cyphers like AES (advanced encryption standard) will be implemented, that have been already proven to be resistant to the exhaustive key search and regarded as strong cypher [14]. The weak point of such algorithms is their own implementation, from which the attacker can reveal secret information like the encryption key. Power consumption of the cyphers implemented on the Evariste II FPGA evaluation board will be exploited using the reference cypher design to calculate the nominal number of traces that are needed to execute a successful attack. After that the cypher will be modified so that it is implemented using some of the fault tolerant "best practices". Among those TMR, error-correction-codes or some other forms of redundancy like multiple repetition of the calculation can be the mentioned. Although it is not expected, that this set of redundancies would change much in the form of attack resistance of the cypher, the aim is to get some basic results that could be later used in combination with attack-resistant "best practices" like hiding or masking the power trace.

3.1 Goals & Proposed results

The goal of the PhD thesis is first to summarize the attack resistance of different fault tolerant implementations, so that the cryptographic schemes can be included into the fault tolerant systems without jeopardizing the assessment of the system fault tolerance and still maintain its level of attack resistance. According to the results of the DPA measurement that are planned for the near future, next course of research might be chosen, like enhancing the current methods or selecting a different form of attack that would be used during the assessment.

4 Conclusion

This paper points at some basic approaches in fault tolerant and attack resistant system design with respect to the common system design. In case the system intends to implement some features of fault tolerance or attack resistance, it must be taken into account at the beginning of the development process. From the thoughts presented in this paper the consideration of the power consumption during the design of the fault tolerant systems may increase their attack resistance. But in order to have fully attack resistant

and fault tolerant system at the same time, the cryptographic scheme must be implemented using fault tolerant "best practices", which may degrade the security properties of the whole design.

The aim of the future research will be to evaluate the described practices and to preserve the fault tolerance of the system by adding some cryptographic features to increase its attack resistance with the focus against the power analysis attacks.

Acknowledgement

This work has been partially supported by grant no. SGS14/105/OHK3/1T/18.

References

[1] E. Dubrova, *Fault-Tolerant Design*. Springer, 2013.

[2] S. Kumar, C. Paar, J. Pelzl, G. Pfeiffer, and M. Schimmler, "Breaking ciphers with copacobana – a cost-optimized parallel code breaker," in *IN WORKSHOP ON CRYPTOGRAPHIC HARDWARE AND EMBEDDED SYSTEMS – CHES 2006,YOKOHAMA*. Springer Verlag, 2006, pp. 101–118.

[3] F. Stepanek, J. Bucek, and M. Novotny, "Differential power analysis under constrained budget: Low cost education of hackers," in *Digital System Design (DSD), 2013 Euromicro Conference on*, Sept 2013, pp. 645–648.

[4] S. Hassoun and T. Sasao, *Logic Synthesis and Verification*, ser. The Springer International Series in Engineering and Computer Science. Springer US, 2002.

[5] J. Wakerly, *Digital Design*. Prentice Hall PTR, 2005.

[6] D. Gajski, *Principles of digital design*. Prentice Hall, 1997.

[7] Z. Salcic and A. Smailagic, *Digital Systems Design and Prototyping: Using Field Programmable Logic and Hardware Description Languages*. Springer, 2000.

[8] B. Schneier. (2013, Aug.) Hacking consumer devices. [Online]. Available: https://www.schneier.com/blog/archives/2013/08/hacking_consume.html

[9] S. McGlaun. (2012, Oct.) Report: Pentagon fails to encrypt drone transmissions. [Online]. Available: http://www.tgdaily.com/security-brief/67192-report-pentagon-fails-to-encrypt-drone-transmissions

[10] A. Huang, *Hacking the Xbox: an introduction to reverse engineering*, ser. No Starch Press Series. No Starch Press, 2003.

[11] S. Mangard, E. Oswald, and T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, ser. Advances in information security. Springer, 2008.

[12] "European Standards EN 50129:2003 - Railway applications: Communication, signalling and processing systems: Safety-related electronic systems for signalling."

[13] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology — CRYPTO' 99*, ser. Lecture Notes in Computer Science, M. Wiener, Ed. Springer Berlin Heidelberg, 1999, vol. 1666, pp. 388–397.

[14] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer, 2010.

Hybridní architektura pro správu knihy s neomezenou hloubkou

Milan Dvořák
Výpočetní technika a informatika, 2. ročník, prezenční studium
Školitel: Jan Kořenek

Fakulta informačních technologií, Vysoké učení technické v Brně
Božetěchova 1/2, 612 66 Brno

idvorakmilan@fit.vutbr.cz

Abstrakt. Silná konkurence mezi účastníky trhu na elektronických burzách si vyžaduje neustálé snižování latence systémů používaných pro obchodování na burze. Poslední snahy vedou k realizaci celého systému na čipu FPGA, čímž dojde k odstranění latence přenosu dat po systémové sběrnici. Některé důležité podčásti systémů však nebyly zatím pomocí FPGA akcelerovány. Příkladem takového problému je správa knihy s neomezenou hloubkou, která se používá na významných akciových burzách. Vzhledem k paměťovým a výpočetním nárokům tohoto problému jsme navrhli novou hybridní hardwarovo-softwarovou architekturu, která na základě zpráv přicházejících z burzy vytváří aktuální knihu s nejlepšími cenami. V hardwaru je udržováno pouze nejlepších N cenových hladin, zbytek je uložen v operační paměti počítače. To umožňuje spravovat polovinu všech akcií (4 000 instrumetů) pomocí jednoho FPGA čipu. Latence aktualizace cenových hladin v hardwaru je pouhých 27 ns. Propustnost hardwarové jednotky je 75 miliónů zpráv za vteřinu, což je 140 krát více než přenosová rychlost dat z burzy.

Klíčová slova. obchodování, burza, FGPA, HW-SW codesign, High Frequency Trading

1 Úvod

Finančním trhům dnes dominuje elektronické obchodování, kdy jednotliví účastníci trhu komunikují s burzou pomocí zasílání zpráv přes počítačovou síť. V hojné míře se používají techniky algoritmického a vysokofrekvenčního obchodování (*High Frequency Trading*, HFT). Obchodník se nezaměřuje na realizaci konkrétních obchodů, ale nastavuje parametry algoritmu, který pak řeší samotné obchodování.

HFT obchodníci využívají nejnovější síťové technologie, aby dosáhli výhody oproti zbytku trhu. I mezi obchodníky však panuje silná konkurence a navzájem se předhánějí v dosažení co nejnižší latence jejich systémů, což je pro ně klíčovým faktorem pro dosažení zisku. Z toho důvodu je věnováno velké usilí v komerční i akademické sféře pro urychlení systémů pro obchodování na burze.

Při zrychlování těchto systémů byla nejprve snaha snížit latenci přenosu dat ze síťového rozhraní do procesoru pomocí speciálních akceleračních karet [1] [2]. Dalšího snižování latence bylo dosaženo akcelerací dekódování zpráv z burzy [3] [4]. Nejnovější snahou v oblasti akcelerace obchodních systémů je realizace celého systému na čipu FPGA [5]. Tím je odstraněna latence přenosu paketů po systémové sběrnici a je dosaženo nejnižších možných latencí. Ne všechny části obchodního systému se však podařilo pomocí FPGA akcelerovat. Lockwood [5] např. neřeší správu knihy, která je však zásadní při zpracování toku dat z burzy. V [6] je sice navržena architektura pro správu agregované knihy s omezenou hloubkou,

ovšem některé významné a zejména akciové burzy používají tzv. knihu s neomezenou hloubkou (viz sekce 2), která zatím nebyla akcelerována pomocí technologie FPGA.

Tento příspěvek představuje hybridní hardwaro-softwarovou architekturu, která umožňuje správu knihy s neomezenou hloubkou. V hardwaru je udržováno pouze nejlepších N cenových hladin, které je možné aktualizovat s latencí pouhých 27 ns. Software obsahuje kompletní obraz všech hladin a v případě potřeby doplňuje chybějící data do hardwaru. Dále je analyzován kompromis mezi počtem cenových hladin uložených v hardwaru, rizikem podtečení a počtem zpráv přenášených po systémové sběrnici. Výsledná architektura byla syntetizována do technologie Virtex-7 a dosahuje frekvence 150 MHz. S využitím dvou modulů QDR SRAM o celkové kapacitě 144 Mibit je možné ukládat obraz burzy až pro 4 tisíce finančních instrumentů, což představuje polovinu celé burzy NASDAQ.

2 Definice problému

Finanční burza je instituce, která umožňuje obchodovat různé finanční instrumenty, např. akcie, derivátové instrumenty nebo komodity. Aktuální cena (kurz) obchodovaných instrumentů se obvykle určuje pomocí průběžné oboustranné aukce mezi nabídkovou (prodejní) a poptávkovou (nákupní) stranou.

Obchodní entity zasílají na burzu své aktuální požadavky pomocí obchodních příkazů. Příkladem takového obchodního příkazu může být *kup 50 akcií firmy Apple za 91 dolarů*. Burza se příchozí požadavky snaží nejdříve spárovat, tzn. najít odpovídající nákupní a prodejní příkaz a provést transakci. Pokud ovšem není možné najít vhodnou protistranu, obchodní příkaz zůstane uložený v tzv. knize.

Kniha obsahuje všechny neprovedené obchodní příkazy pro registrované finanční instrumenty. O aktuálním stavu knihy musí burza informovat své uživatele. V základním režimu burza jednoduše přeposílá informace o jednotlivých obchodních příkazech uživatelům. Pokud tedy obchodník zadá nový požadavek, který se nespáruje, burza mu přiřadí unikátní identifikátor a pošle zprávu typu `ADD` všem uživatelům. Tato zpráva vyjadřuje přidání nového příkazu do knihy a obvykle obsahuje identifikátor příkazu, identifikátor instrumentu, požadovanou cenu, množství a příznak, zda se jedná o nákup či prodej.

V případě, že se obchodník rozhodne změnit svůj existující příkaz, generuje burza zprávu typu `MODIFY`. Tato zpráva obvykle obsahuje identifikátor příkazu, změněnou cenu a změněné množství. Tato zpráva nemusí obsahovat ani identifikátor instrumentu ani původní hodnoty ceny a množství, jelikož tyto informace byly zaslány předchozí zprávou typu `ADD`.

Poslední používaný typ zprávy je `DELETE`. Tato zpráva vzniká, když uživatel zruší svůj příkaz, nebo pokud je tento příkaz spárován a proveden. Zprávy typu `DELETE` již mohou obsahovat pouze identifikátor příkazu, protože ostatní informace jsou známy z předchozích zpráv `ADD` a `MODIFY`.

Pro obchodníky na burze není podstatná informace o jednotlivých příkazech v knize. Obchodní algoritmy obvykle pracují s hodnotami nejlepších cen, na kterých se dané instrumenty obchodují. Systém, který zpracovává příchozí zprávy z burzy, tedy musí z informací o obchodních příkazech vytvořit agregovanou informaci o nejlepších cenách. Principem tohoto zpracování je sdružit příkazy se stejnou cenou, akumulovat jejich požadovaná množství a výsledné cenové hladiny následně seřadit. Tím vzniká agregovaná kniha, která je popsána např. v [6]. Počet cenových hladin v tomto případě je teoreticky neomezený, protože jednotlivé ceny zadávají samotní uživatelé. Tato kniha se proto někdy označuje jako *kniha s neomezenou hloubkou*.

Vzhledem k vynechávání položek ve zprávách `MODIFY` a `DELETE` je nutné ukládat informace pro všechny příkazy. Pro každý příkaz musíme uložit jeho identifikátor (64 bitů), cenu (32 bitů), množství (32 bitů), identifikátor instrumentu (15 bitů) a příznak nákup/prodej (1 bit). Celkem tedy 144 bitů pro každý příkaz.

Agregovaná informace cenových hladin obsahuje cenu (32 bitů), akumulované množství (32 bitů) a počet akumulovaných příkazů (16 bitů), což je celkem 80 bitů.

Celkové paměťové nároky problému správy knihy s neomezenou hloubkou závisí na počtu příkazů, které uživatelé během dne zadají, a na počtu cenových hladin, které tak vzniknou. Jednou z největších

a nejfrekventovanějších burz, které používají tento typ poskytování informace o stavu trhu, je akciová burza NASDAQ. V následující kapitole je tedy popsána analýza datového toku z této burzy.

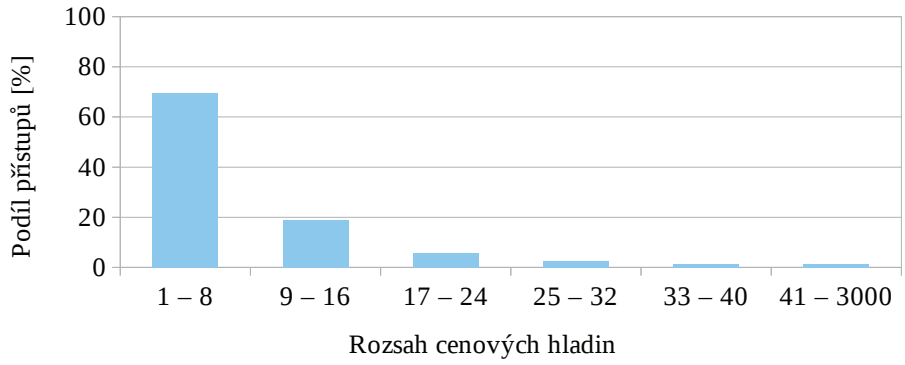
3 Analýza

Pro analýzu paměťových nároků správy knihy s neomezenou hloubkou byl použit celodenní záznam datového toku NASDAQ ITCH4 ze dne 3. 10. 2013. Na této burze se obchoduje necelých 8 000 akcií (instrumentů). Maximální počet příkazů v knize během dne byl více než 1,5 miliónu. Při 144 bitech na jeden příkaz pak potřebujeme minimálně 206 Mibitů pro uložení všech příkazů v knize. Tento objem dat není možné uložit do paměti na čipu, je ovšem možné využít externí statickou paměť.

Tyto příkazy pak vytvářely téměř 350 tisíc cenových hladin na nákupní i prodejní straně, celkově tedy 700 tisíc cenových hladin. Pro 80 bitů na cenovou hladinu pak dostáváme 54 Mibitů. Tento objem dat rovněž není možné uložit do paměti na čipu ani u poslední generace technologie FPGA. Externí paměť pro tento typ dat nepřipadá v úvahu, jelikož cenové hladiny je nutné uchovávat jako seřazený seznam podle hodnoty ceny. Maximální počet hladin na jeden symbol je přitom téměř 3 000. I při využití stromové struktury, která dosahuje logaritmických časů pro vložení položky, by nalezení pozice nové cenové hladiny trvalo nepřípustně dlouho.

Z analýzy celkových paměťových nároků a délky seznamu cenových hladin vyplývá, že nejsme schopni řešit celý problém správy knihy s neomezenou hloubkou na čipu FPGA. Nabízí se možnost na čipu uchovávat a aktualizovat pouze několik nejlepších cenových hladin. Tuto myšlenku podporuje typické chování obchodníka na burze, který svá rozhodnutí vykonává podle několika nejlepších cenových hladin v daném čase. Pro podpoření této myšlenky jsme provedli analýzu lokálnosti přístupů do seznamu cenových hladin.

Pro celodenní záznam z burzy jsme ukládali do histogramu cenovou hladinu, kam přistupují jednotlivé operace `ADD`, `MODIFY` a `DELETE`. Charakter přístupů pro jednotlivé operace byl podobný, stejně tak se podobal histogram přístupů pro nákupní a prodejní stranu. Na obrázku 1 je akumulovaný histogram pro všechny operace na nákupní i prodejní straně.



Obrázek 1: Histogram rozložení přístupů na jednotlivé cenové hladiny

Z histogramu vyplývá, že přístupy k jednotlivým cenovým hladinám vykazují silnou lokalitu. Přes 94 % všech přístupů bylo k prvním 24 hladinám, pro 32 hladin to bylo již 97 % přístupů. K hladinám 41 až 3 000 vede pouhých 1,5 % přístupů. Nutno ovšem poznamenat, že histogram nezohledňuje posun cenových hladin v čase. Jednotlivé hladiny jsou totiž během dne přidávány či naopak odmazávány. Je tedy klidně možné, že aktuálně první záznam v seznamu mohl být ještě před několika málo mikrosekundami v tabulce zanořený mnohem hlouběji.

Lokalita přístupů k cenovým hladinám tedy podporuje myšlenku uchovávat na čipu pouze několik nejlepších cenových hladin. Vzhledem k dynamické povaze této datové struktury je ovšem nutné řešit případné podtečení, kdy se na horní pozice v tabulce dostávají záznamy, které byly před časem mimo několik nejlepších cenových hladin.

4 Architektura

Na základě analýzy operací na burze v předchozí sekci navrhujeme ukládat v hardwaru pouze nejčastěji přistupované cenové hladiny a zbytek udržovat v operační paměti počítače, kde správu knihy zajišťuje software. Čip FPGA slouží jako hardwarová cache. Poskytuje nejlepší cenové hladiny obchodnímu algoritmu s co nejnižší latencí. Zprávy z burzy jsou použity pro rychlou aktualizaci těchto hladin. Software zpracovává všechny zprávy a udržuje kompletní obraz knihy. Při odstranění některých cenových hladin je tak software schopen detekovat podtečení v hardwaru a dodat chybějící informaci zasláním speciální zprávy po systémové sběrnici.

Problém správy knihy s neomezenou hloubkou lze rozdělit na tři podproblémy. První fází je převod identifikátoru instrumentu na interní zkrácenou adresu. Pro řešení tohoto podproblému lze použít architekturu popsanou v [6]. Výstupem této jednotky jsou pak zprávy z burzy obohacené o adresu instrumentu.

Druhou fází je tabulka všech příkazů z burzy. Jedná se o dynamickou tabulku, jelikož příkazy během dne vznikají a zanikají. Vzhledem k velkému množství příkazů na burze je nutné použití hašovacích funkcí, aby byla zajištěna nízká latence a vysoká propustnost. Navrhujeme proto použít kukaččí hašování [7], které se vyznačuje rychlým vyhledáním položky a efektivním využitím paměti [8] [9].

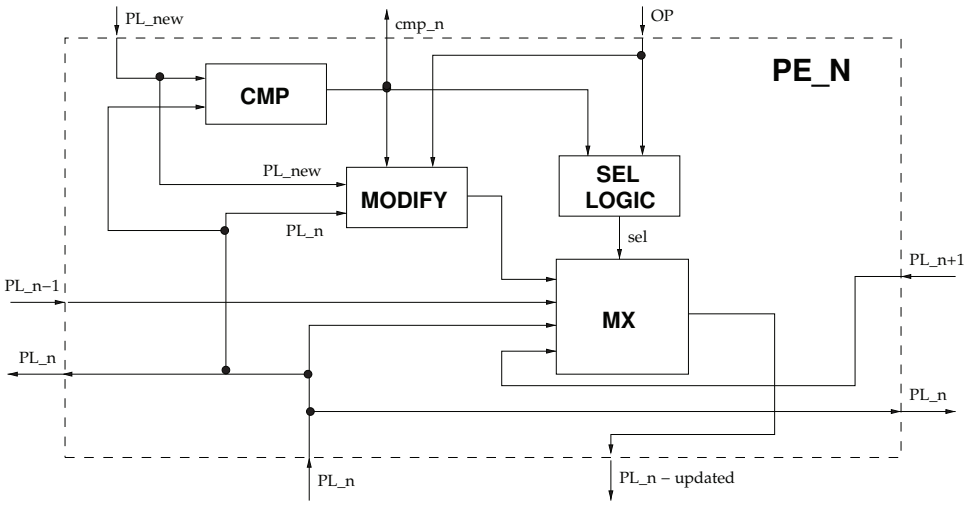
Komponenta s tabulkou příkazů tedy zpracovává zprávy `ADD`, `MODIFY` a `DELETE`. V závislosti na typu zprávy je přidán nový příkaz, nebo smazán či upraven existující příkaz. Informaci o jednotlivých příkazech musíme převést na údaje o cenových hladinách, jak bylo diskutováno v sekci 2. Každá zpráva z burzy generuje aktualizaci pro tabulku cenových hladin. Zpráva `ADD` vede k navýšení množství u dané cenové hladiny. Velikost tohoto navýšení je dána právě množstvím v nově přidaném příkazu. Zpráva `DELETE` naopak vede ke snížení množství u dané cenové hladiny. Zpráva `MODIFY` může způsobit navýšení i snížení množství. Výsledek závisí na tom, jak byl příkaz zprávou upraven.

Poslední komponentou architektury je tabulka cenových hladin, která tyto hladiny uchovává a aktualizuje na základě zpráv z tabulky příkazů. Pro každý instrument je vyhrazena paměť pro uložení N cenových hladin. Parametr N je konfigurovatelný a jeho význam je blíže diskutován v sekci 5. S příchodem zprávy z tabulky příkazů se nejdříve vyčte záznam pro daný instrument. Adresa byla vypočítaná již v rámci tabulky instrumentů, navíc byl k adrese přidán bit s příznakem nákup/prodej. Cenové hladiny jsou tedy uloženy pro nákupní i prodejní stranu zvlášť.

Aktualizace z tabulky příkazů mohou způsobit jednu z následujících operací v tabulce cenových hladin:

- Upravení cenové hladiny, pokud se daná hladina v tabulce již nachází. Množství u příkazu je přičteno nebo odečteno od hodnoty uložené v tabulce.
- Vložení nové cenové hladiny, pokud se navyšovaná cena v tabulce ještě nenachází. Toto vyžaduje posunutí nižších hladin o jednu pozici dolů.
- Odstranění cenové hladiny, pokud u aktualizované hladiny dojde ke snížení množství na nula. Toto vyžaduje odsunutí nižších hladin o jednu pozici nahoru.

Aktualizační operace jsou realizovány paralelně pomocí procesních elementů (`PE`) u každé cenové hladiny. V následujícím textu budeme označovat cenové hladiny jako PL_i a odpovídající elementy jako PE_i pro $1 \leq i \leq N$. Každý element PE_i má 4 datové vstupy, jsou to PL_{i-1}, PL_i, PL_{i+1} a nová cenová hladina PL_{new}, která je vytvořena ze vstupní zprávy. Dále má každý element jeden řídící vstup



Obrázek 2: Architektura procesního elementu

OP, který značí typ aktualizační operace, a jeden řídící výstup cmp_i, což je výsledek porovnání mezi současnou (PL_i) a novou (PL_{new}) cenovou hladinou.

Podrobné schéma procesního elementu je na obrázku 2. Blok `CMP` porovnává vstupní cenovou hladinu PL_i s novou hladinou PL_{new} a vytváří signál $\texttt{cmp}_i$. Blok `MODIFY` realizuje zvýšení nebo snížení množství u cenové hladiny, pokud se současná a nová cena rovnají, jinak tento blok pouze přepošle novou cenovou hladinu. Výsledek porovnání a typ aktualizační operace OP jsou použity také v bloku `SEL_LOGIC` pro výpočet signálu `SEL` u výstupního multiplexoru `MX`. Typ aktualizační operace určuje směr posunutí, výsledek porovnání určuje, jestli je daná hladina pod nebo nad aktualizovanou hladinou a tedy zda se má posouvat. Multiplexor jednoduše vybere jeden ze svých vstupů a tím realizuje požadovanou aktualizační operaci.

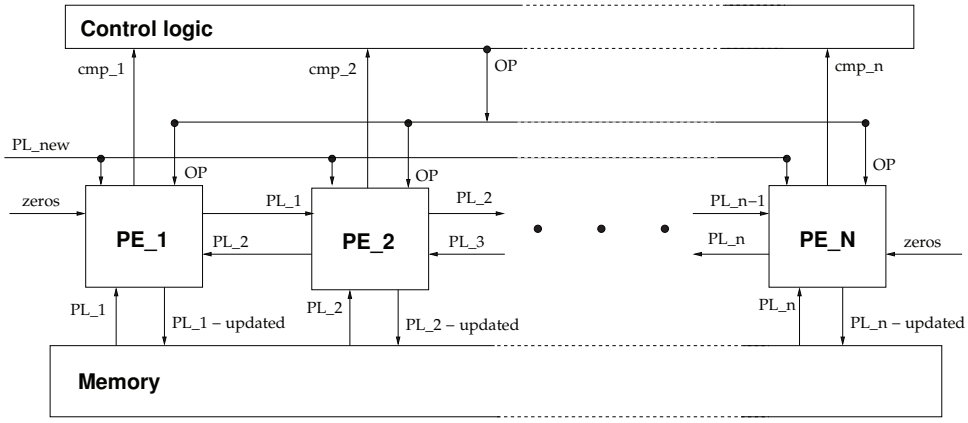
Propojení mezi jednotlivými procesními elementy je zobrazeno na obrázku 3. Každý element načte odpovídající cenovou hladinu z paměti a vyčtenou hodnotu pošle oběma svý sousedům (vstupy PL_{i-1} a PL_{i+1}). Vstup s novou cenovou hladinou je sdílený všemi elementy, které ji porovnají se svou cenovou hladinou PL_i. Výsledky všech porovnání $\texttt{cmp}_i$ jsou zpracovány jednotkou `control logic`, která určí typ aktualizační operace `OP` (změna, vložení nebo smazání hladiny). Jednotlivé elementy použijí typ operace k vybrání výstupní cenové hladiny, která je pak zapsána zpět do paměti. Nejlepší cenové hladiny jsou také přeposlány do obchodního algoritmu (není v obrázku zakresleno).

5 Výsledky

Hardwarovou architekturu popsanou v předchozí sekci jsme implementovali v jazyce VHDL. Jako testovací platformu jsme použili kartu COMBO-80G, která je osazená čipem Virtex-7 XC7VX690T a dvěma paměťovými moduly QDR-II+ SRAM o velikosti 72 Mibit.

Naši implementaci jsme vysyntetizovali pomocí nástroje Xilinx Vivado verze 2013.4. Maximální dosažitelná frekvence je 165.5 MHz, pro reálný obvod bylo použito 150 MHz. Pro kombinační část obvodu, která zajišťuje paralelní aktualizaci všech cenových hladin, jsme nastavili omezující podmínku povolující zpracování jedné aktualizace ve dvou hodinových taktech. Propustnost jednotky je tedy 75 miliónů aktualizačních zpráv za vteřinu, což je 140 krát více než přenosová rychlost analyzovaných dat z burzy. Zpoždění jednotky je pak 4 takty, kromě 2 taktů pro samotnou aktualizaci je potřeba takt na vyčtení záznamu z paměti a takt na zápis výsledku, celkem tedy 27 ns.

Vzhledem k omezenému množství externí statické paměti (144 Mibitů) není možné na této kartě



Obrázek 3: Architektura bloku aktualizace cenových hladin

uložit obchodní příkazy pro všechny obchodované instrumenty. Bylo proto nutné pro zpracování burzy NASDAQ použít 2 karty, přičemž každá měla přidělenou polovinu (4 000) instrumentů.

Dále bylo potřeba řešit omezené množství paměti na čipu. Objem spotřebované paměti je ovlivněn dvěma parametry, jednak počtem instrumentů a dále pak počtem uchovávaných cenových hladin N. Závislost spotřeby zdrojů na těchto dvou parametrech ukazuje tabulka 1.

Počet hladin	**4096 instrumentů**			**8192 instrumentů**		
	Registry	LUT	BRAM	Registry	LUT	BRAM
8	740 (0 %)	5551 (1 %)	242 (16 %)	783 (0 %)	5600 (1 %)	483 (32 %)
16	844 (0 %)	8441 (1 %)	482 (32 %)	862 (0 %)	10646 (2 %)	963 (65 %)
24	680 (0 %)	11951 (2 %)	722 (49 %)	680 (0 %)	13393 (2 %)	1443 (98 %)
32	806 (0 %)	15310 (3 %)	962 (65 %)	911 (0 %)	15411 (3 %)	1923 (130 %)

Tabulka 1: Porovnání spotřeby zdrojů pro různé počty symbolů a cenových hladin

Z tabulky vidíme, že počet obsazených registrů a LUT je velmi nízký i pro 8192 instrumentů a 32 cenových hladin. Objem zabrané paměti na čipu roste lineárně jak s počtem instrumentů, tak s počtem cenových hladin. Pro 4096 instrumentů můžeme uložit až 32 hladin, pro 8192 instrumentů je to jen 16.

Kromě vyhodnocení hardwarové architektury bylo nutné analyzovat synchronizaci se softwarem. Použili jsme stejný záznam z burzy jako v sekci 2. Zaznamenávali jsme počty zpráv do softwaru a ze softwaru. Dále jsme pak pro různé počty cenových hladin v hardwaru sledovali nejhlubší podtečení (minimální počet platných hladin) a počet, kolikrát byl počet hladin nižší jak 5. Hodnota 5 byla zvolena proto, že tento počet hladin často poskytují jiné burzy, které podporují agregovanou knihu.

Naměřené výsledky jsou v tabulce 2. Počet synchronizačních zpráv, které generuje software, i počet zpráv odesílaných z hardwaru klesá s počtem cenových hladin. To je způsobeno tím, že se vzrůstajícím počtem hladin roste počet symbolů, které lze celé uchovávat v hardwaru a není nutná synchronizace. Se vzrůstajícím počtem hladin také přirozeně klesá riziko podtečení. Pouhých 8 hladin je nedostačujících, dochází k častým podtečením až na 0 platných hladin. I pro $N = 16$ občas docházelo k podtečení pod sledovanou hodnotu 5. V případě 24 a 32 hladin již k podtečení nedocházelo, v hardwaru bylo vždy k dispozici alespoň 50 % z uchovávaného počtu hladin.

Z této analýzy tedy vyplývá, že větší počet cenových hladin v hardwaru je výhodný jak z hlediska snížení rizika podtečení, tak z hlediska vytížení systémové sběrnice přenosem zpráv. Rozhodujícím faktorem je tak množství paměti na čipu. Uživatel se může sám rozhodnout, jaký počet cenových hladin vyžaduje ukládat v hardwaru a na základě toho případně snížit počet podporovaných symbolů.

Počet hladin	Zprávy z HW do SW	Zprávy ze SW	Nejnižší hladina	Přesažení hranice
8	6184321	887270	0	42 487
16	5624632	327581	4	88
24	5449269	152218	13	0
32	5360302	63251	21	0

Tabulka 2: Analýza vlivu počtu cenových hladin na riziko podtečení a objem přenášených zpráv

6 Závěr

V tomto příspěvku byl představen problém správy knihy s neomezenou hloubkou v aplikacích pro nízkolatenční obchodování na burze. Tuto úlohu je potřeba akcelerovat pomocí FPGA čipů. Navrhli jsme tedy hybridní architekturu, která umožňuje ukládání horních hladin knihy v hardwaru a doplňování spodních hladin ze softwaru přenosem dat po systémové sběrnici. Ukázali jsme také, jaký vliv má počet hladin v hardwaru na vytížení sběrnice a riziko podtečení. Pokud je nám známo, jedná se o první publikované řešení tohoto problému v FPGA. Latence aktualizace cenových hladin je pouhých 27 ns a maximální možná propustnost je 75 miliónů zpráv za vteřinu.

Výsledky prezentované v tomto příspěvku vznikly v rámci řešení dizertační práce na téma hardwarové architektury s nízkou latencí, které by měly být využitelné zejména v oblasti algoritmického obchodování na burze. Předchozí architekturu problému správy knihy jsem zobecnil pro knihu s neomezenou hloubkou. Tyto problémy nebyly dosud v hardwaru řešeny. Dalším pokračováním této práce by mělo být vylepšení stávající architektury (optimalizace kukaččího hašování pro tabulku příkazů, detailní specifikace a analýza synchronizace se softwarem, ...) a zobecnění pro další případy užití. Praktické nasazení totiž typicky vyžaduje vstupy z více burz a tedy více instancí správy knihy na čipu, což výrazně komplikuje přístupy k paměťovým rozhraním.

Reference

[1] Morris, G. W.; Thomas, D. B.; Luk, W.: FPGA Accelerated Low-Latency Market Data Feed Processing. In *Symposium on High-Performance Interconnects*, ročník 0, 2009, s. 83–89.

[2] Subramoni, H.; Petrini, F.; Agarwal, V.; aj.: Streaming, low-latency communication in on-line trading systems. In *2010 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW)*, 2010, s. 1–8.

[3] Leber, C.; Geib, B.; Litz, H.: High Frequency Trading Acceleration Using FPGAs. In *2011 International Conference on Field Programmable Logic and Applications (FPL)*, 2011, s. 317–322.

[4] Pottathuparambil, R.; Coyne, J.; Allred, J.; aj.: Low-Latency FPGA Based Financial Data Feed Handler. In *IEEE 19th International Symposium on Field-Programmable Custom Computing Machines*, 2011, s. 93–96.

[5] Lockwood, J. W., aj.: A Low-Latency Library in FPGA Hardware for High-Frequency Trading (HFT). In *IEEE 20th Annual Symposium on High-Performance Interconnects*, 2012, s. 9–16.

[6] Dvořák, M.; Kořenek, J.: Low Latency Book Handling in FPGA for High Frequency Trading. In *IEEE 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, 2014, s. 175-178.

[7] Pagh, R.; Rodler, F. F.: Cuckoo hashing. In *Journal of Algorithms*, 2001, str. 2004.

[8] Kirsch, A.; Mitzenmacher, M.; Wieder, U.: More robust hashing: Cuckoo hashing with a stash. In *Proceedings of the 16th Annual European Symposium on Algorithms (ESA)*, 2008, s. 611–622.

[9] Kekely, L.; Žádník M.; Matoušek, J.; Kořenek, J.: Fast Lookup for Dynamic Packet Filtering in FPGA. In *IEEE 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems*, 2014.

PARAMETRIZOVANÝ VÝBER KRITICKÝCH CIEST V DIGITÁLNYCH SYSTÉMOCH

Ing. Miroslav Siebert
Aplikovaná informatika, 2. ročník, denné štúdium
Školiteľ: doc. RNDr. Elena Gramatová, PhD.

Fakulta informatiky a informačných technológií
Slovenská technická univerzita
Ilkovičova 2, 842 16 Bratislava, Slovenská republika

miroslav.siebert@stuba.sk

Abstrakt: Poruchy oneskorení na ceste v digitálnom obvode sú testované nad množinou vybraných kritických ciest. Ich výber je na základe statickej časovej analýzy (*STA*), dynamickej časovej analýzy (*DTA*), prípadne iných metód. Na oneskorenie šírenia signálu však vplývajú viaceré parametre ako sú pokles napájacieho napätia, prechody medzi silikónovými vrstvami pri 3D integrovaných obvodoch, časté prepínanie vstupov, typ šírenej hrany a iné, ktoré môžu zvýšiť kritičnosť cesty. Vplyv jednotlivých parametrov na oneskorenie je známy. Avšak vzájomný vplyv týchto parametrov a tým aj kritičnosť cesty pri ich hľadaní nie je dosiaľ publikovaný. Príspevok prezentuje návrh novej metódy hľadania kritických ciest v digitálnych systémoch na základe viacerých parametrov s nastaviteľnou váhou.

Kľúčové slová: digitálne obvody, poruchy oneskorení, poruchy oneskorení na ceste, kritické cesty, kritičnosť cesty.

1 Úvod

V testovaní digitálnych obvodov bol doposiaľ definovaných veľký počet modelov porúch oneskorení a metód generovania testov. Doteraz boli vyvinuté a sú v značnej miere používané tri základné modely porúch oneskorení: poruchy na prepojeniach, poruchy oneskorení na členoch a poruchy oneskorení na cestách.

Model poruchy oneskorení na cestách je najzložitejší z týchto modelov, nakoľko ich poruchy tvoria súčet oneskorení od vstupu obvodu po jeho výstup. Je schopný detekovať aj malé distribuované oneskorenia od vstupov (alebo výstupov preklápacích obvodov) po výstupy (alebo vstupy preklápacích obvodov) v obvode. V zložitých digitálnych obvodoch existuje veľký počet ciest, ktorý exponenciálne rastie s počtom logických členov. Z toho dôvodu nie je možné otestovať všetky cesty a volí sa iba určitá množina ciest, ktoré sa nazývajú kritické cesty. Na výber kritických ciest sa používa veľa algoritmov a sú založené na viacerých rôznych kritériách. V súčasnosti sa definujú kritické cesty zo statickej časovej analýzy (*STA - Static Time Analysis*), ktorá predpokladá informácie o časovaní navrhnutého obvodu priamo z výroby. Tieto algoritmy spájajú výhody globálneho prístupu na vyššej úrovni a priestorového prístupu na základe presného rozmiestnenia ciest a logických členov priamo na čipe. Tým prispievajú k vyššej kvalite testu a menšieho počtu ciest. Testovacie vektory pre poruchy oneskorenia na týchto cestách sú vygenerované algoritmami automatických generátorov testov (*ATPG - Automatic Test Pattern Generation*).

Niektoré z kritických ciest sú označené ako netestovateľné, nakoľko pre ne neexistuje dvojica testovacích vektorov, ktorá by zabezpečila nábežnú, alebo dobežnú zmenu šíreného signálu na danej ceste. Tieto poruchy môžu byť počas prevádzky maskované, ale ich vzájomnou akumuláciou môže nastať prekročenie akceptovateľnej miery oneskorenia. Ako riešenie tohto problému bola navrhnutá metóda návrhu pre testovateľnosť (*DFT - Design For Testability*) zmeny netestovateľných ciest na testovateľné pridaním jedného logického člena [1], alebo multiplexora [2], v mieste, kde vzniká netestovateľnosť danej cesty . Miesto sa nachádza mimo netestovateľnej kritickej cesty na vstupe niektorého z jej logických členov. Pridanie nového logického člena, alebo multiplexora, však nie je možné na cesty, ktoré sú už kritické, alebo by sa pridaním nového člena kritickými stali, nakoľko by sa do danej cesty vnieslo ďalšie oneskorenie vkladaného člena. Touto metódou je možné v niektorých obvodoch zvýšiť pokrytie až na 100 % a úplne tak odstrániť netestovateľné cesty.

Nasledujúca kapitola opisuje parametre, ktoré majú vplyv na kritičnosť cesty. Architektúra navrhovaného systému je v kapitole 3, Experimentálne výsledky v kapitole 4, ciele dizertačnej práce v kapitola 5. Záveru je venovaná kapitola 6.

2 Parametre vplývajúce na oneskorenie

Z analýzy problematiky kritických ciest a ich výberu možno konštatovať, že kvalitu testu a výber kritickej cesty ovplyvňujú viaceré parametre. Sú to najmä:

- **Robustnosť testu** ─ robustný test je najvhodnejším typom testu, nakoľko porucha je detekovateľná aj v prípade prítomnosti inej poruchy v obvode, ktorá nie je maskovaná. Kombinácia ATPG založeného na časových informáciách a robustného generovania testu môže významne zvýšiť kvalitu testu [3].
- **Zmeny logických hodnôt** (*MIS - multiple input switching)* ─ časté zmeny logických hodnôt na vstupoch logického člena mimo cesty (*off-path*) môžu spôsobiť zvýšenie oneskorenia šírenia zmeny logickej hodnoty až o 36 %. Robustnosť testu v tomto prípade nemá vplyv na zvýšenie oneskorenia [4].
- **Typ šírenej hrany** ─ oneskorenie pri šírení nábežnej a dobežnej hrany signálu je rôzne. Táto asymetria sa zmenou technológie CMOS zo 65 nm na 40 nm zvýšila z 22 % až na 51 % [5].
- **Nedefinované hodnoty** ─ parameter, ktorý definuje koľko bitov z testovacieho vektora môže nadobúdať nedefinovanú hodnotu - X z 5-hodnotovej logiky, pričom $x \in \{0, 1\}$. Čím viacej nedefinovaných logických hodnôt X sa v testovacom vektore nachádza, tým menej je cesta považovaná za kritickú, nakoľko je možné veľkú časť jej hodnôt pomocou kompresie upraviť tak, aby sa eliminovali vplyvy na ostatné parametre.
- **Použiteľnosť cesty vo funkčnom režime** ─ určuje nakoľko nastávajú zmeny logických hodnôt vo funkčnom režime obvodu na danej ceste. Určiť hodnotu pre tento parameter je možné napríklad pomocou simulácie funkčného režimu obvodu. Voľba tohto parametru je dôležitá, aby sa netestovali zmeny logických hodnôt na cestách, ktoré v reálnej prevádzke obvodu nikdy nenastanú, alebo nastanú len minimálne či v špeciálnych prípadoch. V [6] sa uvádza, že cesty, u ktorých scitlivenie vo funkčnom režime nikdy nenastane nie je potrebné zahrnúť do testu.
- **Šum napájacieho zdroja** ─ v obvodoch s veľkou mierou integrácie (*VLSI*) je reálna hodnota napájacieho napätia v jednotlivých obvodoch často nižšia ako špecifikovaná, čo je spôsobené častým preklápaním logických hodnôt. Pokles napätia *I.R* spôsobený parazitnými odpormi a zmena prúdu *i* v čase *t* (*di/dt*) spôsobenou parazitnými indukciami spolu zo zapúzdrením prvkov obvodu v spoločne napájanej doméne sú hlavné faktory vzniku šumu napájacieho napätia [7]. Nadmerné zmeny logických hodnôt vyskytujúce sa najmä počas testovania posúvaním SCAN reťazca, spôsobujú zvýšenie teploty obvodu, šum napájacieho napätia a to vedie k zvýšeniu oneskorenia na jednotlivých logických členoch a následne k zlyhaniu pri testovaní (*overtest*). Oneskorenie na ceste sa šumom napájacieho zdroja môže zvýšiť až o 10 % [8]. Autori v [8]

uvádzajú metódu na výpočet vplyvu tohto šumu na oneskorenie na cestách, ktorý môže byť použitý a zohľadnený už v etape výberu kritických ciest.

- **3D integrované obvody** ─ môže nastať neúplné vyplnene TSV priechodu čo má za následok vznik defektu. Tento defekt môže viesť k slabému prerušeniu (*weak open*), alebo silnému prerušeniu (*strong open*). Pri slabom prerušení je spojenia naďalej funkčné, ale nastáva nárast odporu prepojenia, čo vedie k zvýšeniu oneskorenia na danom prepojení [9]. Rovnako aj vertikálne prepojenia jednotlivých vrstiev TSV môžu byť zdrojom nových porúch a tým aj nových porúch oneskorenia [10]. Pri výbere kritických ciest je preto potrebné zohľadniť, či v prípade 3D integrovaného obvodu daná cesta obsahuje aj TSV prepojenie, čím ju možno označiť za viac kritickú. Autori v [9] navrhujú metodiku pre testovanie TSV na základe parametrov ako sú veľkosť samotného TSV, elektrických parametrov ako napríklad šum napájacieho zdroja a pravdepodobnosti detekcie poruchy.
- **iné** ako napríklad plocha obvodu, spotreba počas testovania a pod.

Z uvedeného vyplýva, že kvalita testovania porúch oneskorení závisí od kvality výberu kritických ciest, ktoré ovplyvňujú viaceré hore uvedené parametre. Každá z doteraz publikovaných literatúr sa zaoberá iba vplyvom jedného z týchto parametrov na oneskorenie šírenia signálu a nedáva zistené poznatky do kontextu s ostatnými parametrami, ktoré ovplyvňujú samotné časovanie obvodu počas testovania a tým aj kritičnosť cesty. Preto nestačí vyhľadať kritické cesty iba podľa ich fyzickej dĺžky, alebo časových parametrov, ale je potrebné sa na ich správny výber pozrieť komplexnejšie, čo je vedeckým cieľom tejto práce. Rovnako je potrebné uvažovať vzájomné vzťahy medzi jednotlivými parametrami, nakoľko niektoré vychádzajú z rovnakého princípu (ako napríklad časté zmeny logických hodnôt a pokles napájacieho napätia z počtu preklopení logických hodnôt) a ich vzájomná miera vplyvu na oneskorenie nebude dvojnásobná. Určenie miery vplyvu súčasného pôsobenia viacerých parametrov je rovnako jedným z vedeckých cieľov projektu.

3 Architektúra systému

Architektúra navrhovaného systému PaCGen je znázornená na obrázku 1. Základom sú vybrané kritické cesty známymi metódami - časovania statickej časovej analýzy (*STA*) a rezerva do hodinového signálu (*slack*). U týchto ciest sa následne overí či sú cesty testovateľné t. j. či existujú také testovacie vektory, ktoré na danej ceste dokážu prešíriť nábežnú a dobežnú hranu signálu. Ak niektoré z ciest sú netestovateľné, blok novej DFT metódy ich upraví na testovateľné pridaním nového logického člena.

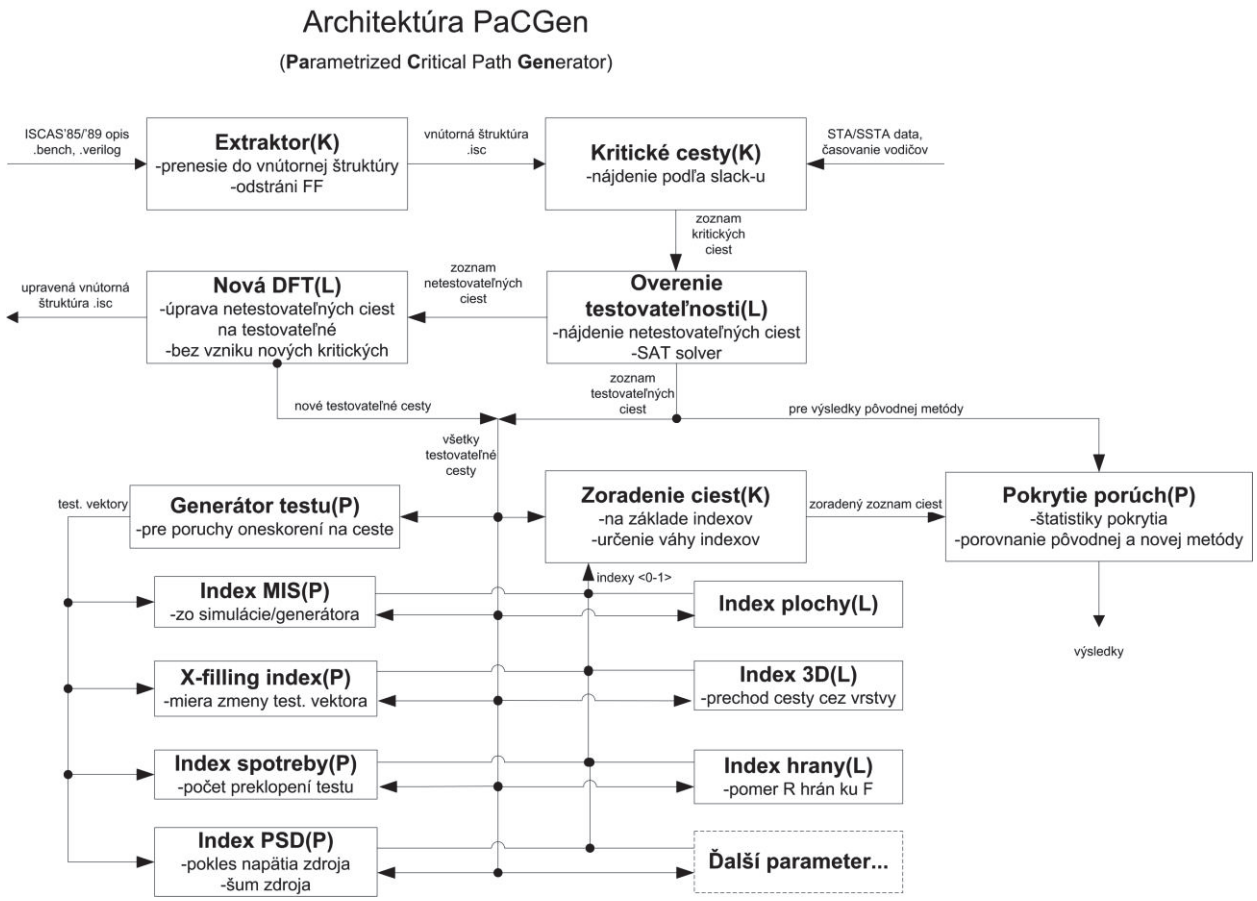
Pre množinu testovateľných ciest sa následne vygenerujú testovacie vektory, na základe ktorých sa príslušnými blokmi vypočítajú hodnoty indexov jednotlivých parametrov uvedených v predchádzajúcej kapitole. Na tento účel bol definovaný vzťah pre kritičnosť c_p, ktorá sa počíta pre každú cestu:

$$c_p = \left(1 - \frac{s_p}{t}\right).\prod_{j=1}^{k}\left[1 - w_j\left(1 - i_{jp}\right)\right],$$

kde s_p predstavuje rezervu do hodinového signálu, t reprezentuje dĺžku časového intervalu hodín, k je počet uvažovaných parametrov vplývajúcich na oneskorenie, w_j váha konkrétneho parametra j a i_{jp} reprezentuje vypočítaný index parametra j vplývajúceho na cestu p. Kritičnosť dosahuje hodnoty <0;1>. Pre váhy všetkých parametrov platí vzťah, kde I je maximálna hodnota vplyvu parametrov:

$$\sum_{j=1}^{k} w_j < I\,.$$

Odporúčaná hodnota I = 0,2, získaná z experimentálnych výsledkov publikovaných k výskumu maximálneho vplyvu jednotlivých parametrov na oneskorenie šírenia signálu. Z množiny takto zoradených ciest zoradených podľa kritičnosti zvolíme obmedzenú množinu ciest, ktorá bude zvolená pre test a porovnáme kvalitu testu s nezoradenou množinou rovnakej veľkosti.



Obrázok 1: Architektúra systému PaCGen.

4 Experimentálne výsledky

V tejto časti sú uvedené výsledky implementácie systému PaCGen nad testovacími obvodmi ISCAS-89. Použité boli dáta STA syntézou v programe Cadence Encounter RTL Compiler s použitím 45nm NanGate FreePDK45 Generic Open Cell knižnice, CMOS technológie, typické podmienky výrobného procesu (*corner*), typické napájacie napätie 1,2 V, teplota 25^0C, nominálne prahové napätie a leakage.

Tabuľka 1 znázorňuje percentá pokrytia porúch oneskorení na prepojeniach nad vybranými obvodmi ISCAS-89. Druhý stĺpec znázorňuje pokrytie bez zoradenia kritických ciest navrhnutou metódou. V treťom stĺpci sú výsledky pokrytia po aplikovaní navrhnutej metódy zoradenia ciest na základe vplyvu viacerých parametrov. V obidvoch prípadoch bolo zvolených 80 % najkritickejších ciest, v prípade zložitejších obvodov len 20 %, z dôvodu simulácie obmedzenej veľkosti pamäti testera. Týmto obmedzením je demonštrované, že navrhnutou metódou je možné vybrať kvalitnejšiu množinu ciest pre test na poruchy oneskorení na ceste.

Tabuľka 2 znázorňuje percentá pokrytia porúch oneskorení na prepojeniach nad vybranými obvodmi ISCAS-89 aj s použitím novej DFT metódy z [1]. Druhý stĺpec znázorňuje pokrytie bez zoradenia ciest a bez aplikovania DFT metódy. Tretí stĺpec už zobrazuje pokrytie po zoradení ciest a aplikovaní DFT metódy. Počet pridaných nových logických členov na zabezpečenie testovateľnosti netestovateľných ciest znázorňuje štvrtý stĺpec a v piatom je počet ciest, ktoré sa stali testovateľnými z netestovateľných po aplikovaní tejto DFT metódy. Z uvedených výsledkov vyplýva, že navrhnutá

metóda aj s použitím novej DFT metódy je vhodnejšia pre zložitejšie obvody s vyšším počtom hradiel, nakoľko pri menších a jednoduchších obvodoch bol prínos záporný. To je spôsobené tým, že pridaním nových logických členov vznikli nové cesty obvodu.

Obvod	**Pokrytie [%]**	
	Nezoradené	***Zoradené***
s27	84.21	86.84
s298	77.66	77.87
s344	72.84	73.20
s420	82.44	82.74
s641	52.22	53.43
s713	25.72	27.07
s820	47.56	47.82
s1196	33.32	33.81

Tabuľka 1: Výsledky pokrytia porúch oneskorení

Obvod	**Pokrytie [%]**		**DFT technika**	
	Nezoradené	***Zoradené s DFT***	***Počet pridaných členov***	***Počet zmenených ciest***
s820	47.56	51.4	66	1920
s832	46.29	51	79	21147
s953	31.33	33.68	16	71
s1196	33.32	43.6	2	6
s1196a	33.32	44.32	6	14
s1196b	33.32	43.6	81	2160
s1238	17.8	33.53	80	2162
s1238a	18.04	33.61	81	2169
s1488	23	24,91	83	1812
s641	52,22	55,27	45	192

Tabuľka 2: Výsledky pokrytia porúch oneskorení s metódou DFT

5 Ciele dizertačnej práce

Z analýzy zabezpečenia testovateľnosti porúch oneskorení synchrónnych sekvenčných obvodov vyplynulo, že neexistuje komplexné riešenie resp. metodika výberu kritických ciest a testovania porúch oneskorení na ceste týchto obvodov. Na základe toho ciele dizertačnej práce sú:

- Špecifikácia parametrov vplývajúcich na výber kritických ciest v obvode, ako napríklad MIS, šum napájacieho zdroja, pokles napájacieho napätia, použiteľnosť cesty vo funkčnom režime, robustnosť cesty, typ šírenej hrany, nedefinované hodnoty a iné.
- Návrh novej metódy pre výber kritických ciest na základe zvolených parametrov s cieľom zvýšiť pokryte porúch oneskorení. Metóda by mala byť čo najuniverzálnejšia, škálovateľná a

flexibilná vzhľadom na výber, alebo použitie parametrov pre nájdenie kritických ciest v kombinačnom obvode.
- Formalizácia výberu kritických ciest v obvode podľa zvolených parametrov s využitím váhovania významu týchto parametrov a návrh vhodných váh jednotlivých parametrov.
- Implementácia navrhnutej metódy a jej overenie nad experimentálnymi obvodmi.
- Návrh metódy pre zefektívnenie testovania porúch oneskorení na existujúcich netestovateľných kritických cestách pomocou zmeny štruktúry testovaného obvodu.

6 Záver

V tomto príspevku boli predstavené motivácia, ciele a priebežné výsledky dizertačnej práce, ktoré sa zamerali na návrh novej metódy zabezpečenia testovateľnosti porúch oneskorení v synchrónnych sekvenčných obvodoch výberom kritických ciest. Jednotlivé parametre majú nastaviteľnú váhu vplyvu na kritičnosť cesty. Bola navrhnutá a implementovaná architektúra systému PaCGen s experimentálnymi výsledkami nad testovacími obvodmi ISCAS'89.

Súčasná práca je venovaná hľadaniu optimálnych váh vplyvu jednotlivých parametrov a návrhu metodiky výpočtu miery zlepšenia testu pre model porúch oneskorení na ceste uvedenou metódou vzhľadom na pravdepodobnosť výskytu poruchy oneskorenia.

Poďakovanie

Táto práca bola čiastočne podporená projektom (VEGA 1/1008/12) a COST Action IC 1103 MEDIAN.

Literatúra

[1] Siebert, M, Gramatova, E : Delay fault coverage increasing in digital circuits, in *Proc. of the Euromicro Conference on Digital System Design (DSD)*, 2013, pp. 475-478.

[2] Pomeranz, I, Reddy, S, M : Design-for-Testability for Improved Path Delay Fault Coverage of Critical Paths, in *Proc. of the 21st International Conference on VLSI Design*, 2008, pp. 175-180

[3] Eggersgluss, S, Yilmaz, M, Chakrabarty, K : Robust Timing-Aware Test Generation Using Pseudo-Boolean Optimization, in *Proc. of the 21st Asian Test Symposium (ATS)*, 2012, pp. 290-295.

[4] Wu, S. H, Chakravarty, S, Wang, L : Impact of Multiple Input Switching on Delay Test under Process Variation, in *Proc. of the 28th IEEE VLSI Test Symposium*, 2010, pp. 87-92.

[5] Wu, S. H, Chakravarty, S, Tetelbaum, A, Wang, L : Refining Delay Test Methodology Using Knowledge of Asymmetric Transition Delay, in *Proc. of the 17th Asian Test Symposium (ATS)*, 2008, pp. 137-142.

[6] Pomeranz, I : On the Detection of Path Delay faults by Functional Broadside Tests, in *Proc. of the 17th IEEE European Test Symposium (ETS)*, 2012, pp. 1-6.

[7] Rao, K. S, Robucci, R, Patel, Ch : Scalable Dynamic Technique for Accurately Predicting Power-Supply Noise and Path Delay, in *Proc. of the 31st VLSI Test Symposium (VTS)*, 2013, pp. 1-6.

[8] Rao, K. S, Sathyanarayana, Ch, Kallianpur, A, Robucci, R, Patel, Ch : Estimating Power Supply Noise and Its Impact on Path Delay, in *Proc. of the 30th VLSI Test Symposium*, 2012, pp. 276-281.

[9] Metzler, A, et al. : Computing Detection Probability of Delay Defects in Signal Line TSVs, in *Proc. of the 18th IEEE European Test Symposium*, 2013, pp. 1-6.

[10]Panth, S, Lim, S. K : Transition Delay Fault Testing of 3D ICs with IR-Drop Study, in *Proc. of the 30th VLSI Test Symposium*, 2012, pp. 270-275.

Detekcia sieťových anomálií a bezpečnostných incidentov s využitím DNS dát

Michal Kováčik
Výpočetní technika a informatika, 2. ročník, prezenční studium
Školitel: Jan Kořenek

Fakulta informačních technologií, Vysoké učení technické v Brně
Božetěchova 1/2, 612 66 Brno

ikovacik@fit.vutbr.cz

Abstrakt. Služba DNS je kritická pre normálne fungovanie Internetu a taktiež množstva dostupných služieb. Väčšina komunikácie na Internete totiž využíva v istej fáze práve DNS. Okrem jej základnej úlohy sa často stáva terčom zneužitia pri množstve rôznych škodlivých aktivít. Táto práca sa zaoberá nežiaducimi aktivitami spájajúcimi sa so službou DNS a jej zneužitím, ktoré sú priblížené spolu s mojim vlastným prístupom k ich detekcii. Najvýznamnejšou časťou práce je kapitola o dizertačnej práci, ktorá špecifikuje vytýčené ciele, približuje spôsob ich dosiahnutia a súčasný stav.

Kľúčové slová. detekce anomálií, bezpečnostní incidenty, DNS útoky, monitorování provozu

1 Úvod

Požiadavky na správu a bezpečnosť počítačových sietí neustále rastú spolu s ich rozvojom. Vysoká dôležitosť sa kladie hlavne dostupnosti služieb a diskrétnosti prenášaných informácií. Rozvíjajú sa však aj útoky a ich počet má stúpajúcu tendenciu. Tento narastajúci trend potvrdzujú aj spoločnosti ako napríklad NSFOCUS[1] alebo Symantec[2] zaoberajúce sa internetovou bezpečnosťou vo svojich výročných správach. Sila a počet útokov na DNS alebo zneužívajúcich službu DNS sa za posledné roky pravidelne takmer zdvojnásoboval, čo dokazuje stúpajúcu popularitu zahrnutia tejto služby do útokov. Monitorovanie sietí za účelom detekcie a zamedzenia sieťových anomálií si vyžaduje stále viac pozornosti.

Služba DNS (Domain Name System)[3] je z pohľadu štruktúry hierarchický systém doménových mien. Hlavnou funkciou služby je preklad doménových mien na IP adresy a opačne, vykonávaný rezolúciou. V skutočnosti služba pracuje s množstvom rôznych dotazov a je možné ju považovať za distribuovanú databázu sieťových informácií, ktorej uzlami sú menné servery. Protokol pracuje na jednoduchom princípe dotazu a odpovede a komunikácia sa vyznačuje symetrickosťou. To znamená že by mala existovať odpoveď na každý zaslaný dotaz, čo však v praxi kvôli protokolu UDP nie je možné zaručiť. Protokol DNS samotný nepoužíva šifrovanie a jeho autentifikácia pomocou zdrojovej IP adresy, portu a transakčného ID je veľmi jednoduchá.

Dôležitosť DNS je zrejmá aj útočníkom, ktorí protokol používajú za nedovolenými účelmi na škodlivé aktivity, prípadne zneužívajú vlastnosti DNS. Bezpečnostné opatrenia v množstve sietí bývajú k DNS

[1] spoločnosť zaoberajúca sa medzinárodnou webovou a sieťovou bezpečnosťou <http://www.nsfocus.com.au/>
[2] spoločnosť poskytujúca bezpečnostné sieťové riešenia <http://www.symantec.com/>
[3] <https://www.ietf.org/rfc/rfc1034.txt>, <https://www.ietf.org/rfc/rfc1035.txt>

prevádzke veľmi benevolentné, čo je obrovskou výhodou pre útočníkov, pre ktorých môže byť DNS prístupovou cestou aj do sietí s vysokým zabezpečením, ktoré sú konfigurované veľmi prísne voči ostatným službám. Útočníci využívajú rôzne techniky ako napríklad častá zmena doménového mena pre vyhnutie sa blokovaniu prístupu, podvrhnutie odpovede na dotaz klienta, zneužitie protokolu na posielanie odlišného typu dát a podobne. DNS komunikácia prebieha tiež medzi stanicami v botnetom.

Nasledujúca kapitola 2 sa venuje problematike zdrojových dát. Kapitola 3 sa venuje konkrétnym DNS anomáliám a ich detekcii. V ďalšej kapitole 4 sa nachádza formulácia cieľa mojej dizertačnej práce, spolu so spôsobmi jeho dosiahnutia. Záverečná kapitola 5 je súhrom tohto príspevku.

2 Monitoring a zdrojové dáta

Dôležitým faktorom pre voľbu detekčnej metódy je typ dostupných zdrojových dát. Na základe dostupného typu zdrojových dát je potom možné odhadovať presnosť a rýchlosť detekčnej metódy. V dnešnej dobe je veľmi populárnym riešením používanie tokových dát (NetFlow[4]). Tento spôsob monitorovania sa pre DNS, ktorý je aplikačným protokolom, javí pri niektorých typoch anomálií ako nedostatočný. Pri DNS sú vo väčšine prípadov veľmi dôležité dáta z položiek aplikačnej vrstvy, ktoré flow dáta neobsahujú. Najideálnejším riešením by samozrejme bolo zaznamenávanie celých paketov (Deep packet inspection), no analýza takýchto paketov by vyžadovala obrovské nároky na výpočtový výkon a rovnako obrovský priestor pre ukladanie zaznamenaných dát. Dôležitou požiadavkou pri monitoringu DNS je však aj efektivita monitorovania a spracovania prevádzky. Nutnosťou je teda hľadanie kompromisu medzi monitorovaním tokov a kompletných paketov.

Na základe možností, ktoré ponúka protokol IPFIX[5] (Internet Protocol Flow Information eXport), by práve jeho použitie malo byť strednou cestou zahŕňajúcou efektívny monitoring ako aj možnosti analýzy vybraných položiek aplikačných protokolov. Zdrojom IPFIX dát, ktorý používam sú dáta z DNS pluginu [5] pre FlowMon Exportér od spoločnosti INVEA[6], ktorý som vyvíjal. Týmto spôsobom mám k dispozícií vybrané položky z aplikačnej vrstvy paketov DNS prevádzky.

3 DNS anomálie a detekčné metódy

Anomálie DNS je možné rozdeliť do kategórií podľa niekoľkých faktorov. V tejto kapitole sa zameriam iba na vybrané typy anomálií, niekoľko vybraných detekčných metód a vlastný prístup k nim v rámci mojej dizertačnej práce. Úplné rozdelenie, detailný popis jednotlivých anomálií a metód detekcie, a popis viacerých typov anomálií som zhrnul v tézach [6]. Ďalšie informácie som čerpal z [9].

3.1 DNS Amplification

Je najpopulárnejším z útokov, ktorý službu DNS zneužíva. Útok sa skladá z dvoch hlavných častí. Prvou je spoofing[7] zdrojovej IP adresy, druhou je vygenerovanie dotazu, ktorý spôsobí čo najväčšiu odpoveď. Vzhľadom k tomu, že sa pri tomto útoku generuje obrovské množstvo dotazov a zneužívané DNS servery odpovedajú mnohonásobne väčšími odpoveďami, je možné tento útok detekovať už pomocou tokových dát vo forme NetFlow.

Na detekciu útoku je možné použiť mnoho zaujímavých metód, ako príklad vyberiem metódu založenú na NetFlow dátach [1], ktorá funguje pomocou jednoduchých prahov. Metoda vyniká jednoduchosťou a rýchlosťou, jej presnosť však nie je ideálna, pretože generuje priveľa falošných poplachov. Na základe

[4] definovaný v <http://www.ietf.org/rfc/rfc3954.txt>
[5] definovaný v <http://www.ietf.org/rfc/rfc5101.txt>
[6] viac na <https://www.invea.com/sk/go/flowmon>
[7] podvrhnutie

tejto metódy som v spolupráci so združením CESNET implementoval vlastnú, ktorá detekuje útok na základe homogenity dotazov a odpovedí, asymetrickej veľkosti dotazov a odpovedí a početnosti dotazov. Pri relatívne zachovanej jednoduchosti bola dosiahnutá oveľa vyššia presnosť detekcie. Metóda je nasadená ako detekčný modul v systéme NEMEA [2]. Z ďalších prístupov k detekcii, ktoré som analyzoval je možné spomenúť detekciu na základe metódy podobnosti a entropie. Metódy sa ukázali ako úspešné a sú schopné detekovať útok, nevýhodou však je nutnosť dodania vhodných referenčných dát.

Ako možnú alternatívu detekcie amplifikačného útoku som skúmal súvislosť s položkami *DNSSEC OK* a *UDP payload size*, ktoré sú súčasťou rozšírenia EDNS0[8]. Obsah týchto položiek však nie je možné priamo spojiť s útokmi, keďže väčšina DNS prevádzky používajúca EDNS0 pracuje s hodnotami položiek, ktoré boli predpokladané v prítomnosti útoku. Pre zlepšenie presnosti detekcie a potvrdenie útoku, je možné použiť mnou publikovaný detektor podvrhnutých adries na sieti [7], čo priblížim v 4.1.

3.2 DNS tunneling

Hlavnou myšlienkou je zapuzdrenie dát do klasickej DNS prevádzky, ktorá nebýva nijako obmedzovaná. Takto je potom možné tunelovať akúkoľvek prevádzku a obchádzať firewaly, či platené prístupové body do siete. Tunelované pakety sa vyznačujú neobvyklou veľkosťou, veľkou dĺžkou doménového mena, veľkým počtom číslic v názve domény, ktorý býva navyše vygenerovaný.

Tunelovanie vzhľadom k prenášaným paketom mení charakter DNS prevádzky a detekcia je teda za istých okolností možná aj z tokových dát. Použiteľné sú napríklad metódy založené na entropii, podobne ako v [4], kde je takáto metóda použitá na analýzu histogramov veľkostí paketov. Okrem toho autori v tomto prístupe sledujú aj frekvenciu nekonformných paketov. Ďalšou je možnosť analýzy tokových dát štatistickými metódami. V tomto prípade je však nutné správne určiť parametre pre detekciu a tiež hraničné hodnoty pre anomálne správanie. Od toho sa potom odvíja celková presnosť metódy. Každá sieť má iné charakteristiky a preto je vždy najskôr nevyhnutné vykonať analýzu sieťovej prevádzky. Vhodnejšia sa javí analýza obsahu paketov pri ktorej množstvo metód zameriava na zmysluplnosť prenášaných dotazov a odpovedí. Najčastejšia je detekcia pomocou frekvenčnej analýzy v rôznych variantách. Zo všetkých spomeniem frekvenčnú analýzu na jednotlivých bigramoch [8].

Pri vlastnej analýze a detekcii tunelovania pomocou DNS som sa zameriaval v prvom rade na netypické typy odpovedí, ktoré sú používané. Často sa pre prenos používajú hlavne typy TXT, SRV alebo napríklad NULL. Ďalšou sledovanou vlastnosťou bola neprimeraná veľkosť paketov. Význačnou je aj dĺžka doménového mena, ktorá býva oproti bežnej prevádzke dvoj- až troj-násobná. Použitím frekvenčnej analýzy doménového mena je detekcia veľmi úspešná, čo je bohužiaľ na úkor rýchlosti detekcie. Generované doménové mená majú na rozdiel od skutočných približne rovnomerné rozloženie znakov, čo nezodpovedá žiadnemu bežnému jazyku.

3.3 Cache poisoning

Jedná sa o podvrhnutie obsahu cache záznamu na serveri za účelom presmerovania. Detekcia je možná aj pomocou štatistickej analýzy DNS, no problémom zostáva generovanie množstva falošných poplachov.

Autori v [4] používajú pre detekciu algoritmus pracujúci s NetFlow, ktorý používa IP adresy zdrojov a cieľov, čísla portov, časy medzi príchodmi jednotlivých paketov a postupnosť udalostí. Algoritmus zaznamenáva prichádzajúce dotazy a odpovede a na základe ich postupnosti a početnosti je schopný generovať poplach pri útoku.

Pri vlastných experimentoch som sa zameral na detekciu pomocou krátkej histórie. Metóda sa zameriava na pokusy o uhádnutie transakčného ID a používam v nej transakčné ID dotazu, znenie dotazu, zdrojovú a cieľovú IP adresu a zdrojový port. Unikátne kombinácie dotazov sa zaznamenávajú a uchovávajú. Po príchode zodpovedajúcej odpovede je dotaz odstránený z histórie. Pokiaľ sa líši v transakčnom ID,

[8] Extension mechanisms for DNS <http://www.ietf.org/rfc/rfc2671.txt>

môže sa jednať o narodeninový útok, ktorým je cache poisoning sprevádzaný. Upozornenie sa však hlási až po obdržaní viac ako jedného paketu s rôznym ID, aby sa predchádzalo falošným poplachom. Problémom metódy je efektívne ukladanie histórie v prípade, že je počet dotazov väčší ako počet odpovedí, v tomto prípade môže nekontrolovane rásť množstvo záznamov pre porovnanie. Taktiež má algoritmus problém s niektorými anomálnymi prejavmi v DNS prevádzke, ktoré ale nesúvisia s cache poisoning.

3.4 Škodlivé domény

So škodlivými doménami sa spája používanie techniky fast-flux, ktorá dovoľuje zneužiť vlastnosti DNS na sťaženie zablokovania domén. Pre tento typ anomálneho správania obsahuje NetFlow nedostatočnú informáciu pre detekciu a jedinou možnosťou je v tomto prípade použitie formátu zdrojových dát obsahujúceho aj vybrané položky z aplikačnej vrstvy. Okrem úplných paketových dát sa ideálne ponúka IPFIX obohatený o aplikačné dáta, ktorý by obsahoval napríklad kľúčové položky ako TTL, dotazované doménové mená a podobne.

Autori v [10] sa zameriavajú na domény, na ktoré chodí abnormálny alebo koncentrovaný počet dotazov a na detekciu dotazov na neexistujúce doménové mená (NXDOMAIN). Detekcia odpovedí NXDOMAIN sa pritom javí ako pomerne úspešná. Okrem toho existuje viacero prác, ktorých výsledkom je reputačný systém na základe pasívnej analýzy DNS prevádzky. Jedným z nich je aj [3], kde autori extrahujú z DNS prevádzky 15 význačných príznakov na ktoré sa zameriavajú. Vhodným doplnkom každej metódy na detekciu domén je kontrola voči Blacklistom.

Pri vlastných experimentoch som sa zameral na niekoľko spôsobov určenia škodlivých domén. Analyzované domény predspracovávam rozdelením na jednotlivé úrovne domén a vynechaním častí kratších ako štyri znaky. Takto rozdelené doménové meno je podrobené frekvenčnej analýze. Navyše sa experimentálne snažím pracovať s analýzou skladby slov, ktorá pozostáva z niekoľkých častí. Prvou časťou je analýza dĺžky časti doménového mena, ktorá má hraničnú hodnotu priradenú na základe priemernej dĺžky doménového mena v normálnej prevádzke. Druhou časťou je detektor počtu samohlások, ktorý porovnáva počet samohlások voči počtu písmen. V tretej časti sa sleduje počet opakujúcich sa písmen v názve domény voči jej dĺžke. Posledná štvrtá časť analyzuje počet číslic v doménovom mene.

4 Ciele dizertačnej práce

Moja dizertačná práca sa zameriava na pokrytie nedostatkov existujúcich metód a tým o dosiahnutie lepších výsledkov v oblasti detekcie. Jednotlivé metódy pracujú s rôznym typom vstupných dát, prípadne využívajú iba podmnožinu dostupných informácií. Rôzne vstupné dáta často vedú k rôznym stupňom efektivity a presnosti pri detekcii. Z toho dôvodu v rámci mojej práce, využívam spoločne zdrojové dáta vo formáte NetFlow (tokové), IPFIX (obohatené o aplikačnú vrstvu) a plné paketové dáta. Pritom sa snažím nájsť čo najlepšiu rovnováhu v ich súčinnosti pre potreby posilnenia bezpečnosti počítačových sietí. Rovnako sa v rámci práce snažím o čo najlepšiu efektivitu detekčných metód a ich univerzálnosť. Cieľ mojej dizertačnej práce som formuloval vo vlastných tézach [6] a jeho znenie je:

S využitím kombinácie a korelácie zdrojových DNS dát s kompletným obsahom paketov (Deep packet inspection) a NetFlow/IPFIX dát (IP Flow monitoring) zefektívniť detekciu anomálií a bezpečnostných incidentov v DNS dátach s ohľadom na jej rýchlosť a presnosť.

Hlavný cieľ, ktorý som formuloval je možné rozčleniť na niekoľko menších cieľov, esenciálnych pre jeho dosiahnutie:

1. Analýza dostupných zdrojových DNS dát pomocou rôznych variant korelácie.

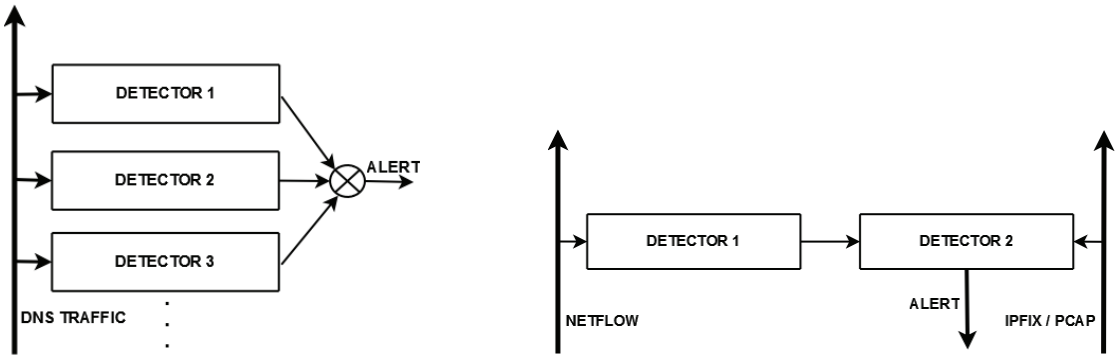
2. Určenie kľúčových metrík potrebných pre detekciu jednotlivých typov anomálií.
3. Návrh optimalizovaných detekčných metód.
4. Návrh vhodného spôsobu kombinácie výsledkov z jednotlivých detekčných metód.
5. Implementácia systému realizujúceho vybrané metódy.
6. Experimentálne vyhodnotenie dosiahnutých výsledkov.

4.1 Spôsob riešenia

Hlavnou myšlienkou práce je využitie rôznych typov dát spolu s DNS dátami za účelom vytvorenia sady detektorov pre rôzne typy anomálií, ktorých efektivita a presnosť bude vyššia než pri obyčajných detektoroch. Pre každú anomáliu môže koexistovať niekoľko detektorov, ktoré navzájom spolupracujú. Možné spôsoby spolupráce sú načrtnuté na Obrázku 1.

Ukážka vľavo na obrázku predstavuje spoluprácu na princípe potvrdenia incidentu a teda spresnenia detekcie. Ako príklad môžem uviesť mnou implementované riešenie dvoch detektorov. Prvý detektor sa snaží odhaliť útok DNS Amplification. Druhý detektor zachytáva v sieti IP spoofing. Koreláciou výsledkov týchto dvoch detektorov sa potvrdí existencia anomálie. V tomto prípade oba detektory pracujú s tokovými dátami.

Ukážka vpravo predstavuje spoluprácu na rozdielnej úrovni zdrojových dát. Jednoduchý detektor avizuje druhému detektoru udalosť, na základe ktorej druhý detektor extrahuje a využije informácie z aplikačnej vrstvy. Konkrétny príklad znova uvediem z vlastnej práce. Prvý, jednoduchý detektor monitoruje a zaznamenáva priebeh SMTP prevádzky. Na jej základe druhý detektor zachytávajúci DNS dáta obohatené o položky aplikačnej vrstvy vo formáte IPFIX dohľadá v prevádzke prípadnú existenciu reverzného dotazovania sa na zdroj SMTP prevádzky a výsledok tohto dotazovania. V prípade negatívnej odpovede je možné zdroj pokladať za škodlivý, kvôli distibúcií nevyžiadanej pošty vo forme spamu.



Obrázok 1: Ukážka spolupráce niekoľkých detektorov.

Koreláciu dát je nutné vykonať z rôznych pohľadov - dáta z rôznych zdrojov, dáta rôznych typov či úrovne. Zaujímavá môže byť aj korelácia na rôznych časovo merateľných intervaloch a na základe rôznych množín. Získané poznatky z korelačných experimentov sú dôležité z pohľadu súvislostí jednotlivých skupín dát, a rovnako aj z pohľadu vhodnosti použitia určitej detekčnej metódy. Na základe dôkladnej analýzy je potom potrebné určiť konkrétne položky dát, ktoré sú pre detekciu daného incidentu nevyhnutné alebo prospešné. Tento krok vedie k návrhu optimalizovaných detekčných metód.

Výsledky jednotlivých detektorov alebo ich častí bude potrebné vhodne kombinovať. Je preto nevyhnutné navrhnúť hierarchiu jednotlivých ukazovateľov a ich podiely na výslednej detekcii. Výsledky niektorých detektorov by napríklad mali byť zohľadnené pri rozhodovaní iných.

5 Záver

Hlavnou úlohou tohto príspevku bolo predstaviť ciele mojej dizertačnej práce a načrtnúť spôsoby ich dosiahnutia. Venoval som sa problematike vhodnosti zdrojových dát a dospel som k záveru, že najlepším riešením je využívanie IPFIX s tokovými dátami obohatenými o položky aplikačnej vrstvy a kombinovanie viacerých typov dát. Ďalej som popísal vlastný prístup k vybraným anomáliám, vybral zaujímavé metódy ich detekcie a priblížil získané poznatky. V kapitole o dizertačnej práci som potom poskytol návrh riešenia spolupráce viacerých detektorov, respektíve detekcie pomocou viacerých typov dát. Momentálne sa venujem optimalizácií a zlepšovaniu algoritmov detekcie, pričom sa snažím nachádzať súvislosti a vyvodiť návaznosti jednotlivých typov dát a výsledkov detektorov.

Poďakovanie

Táto práca bola podporená projektom IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

Reference

[1] *Detecting Reflection Attacks in DNS Flows*, ročník 19, University of Twente, 2013.

[2] Bartoš, V.; Žádník, M.; Čejka, T.: Nemea: Framework for stream-wise analysis of network traffic. URL <http://www.cesnet.cz/wp-content/uploads/2014/02/trapnemea.pdf>

[3] Bilge, L.; Kirda, E.; Kruegel, C.; aj.: EXPOSURE : Finding malicious domains using passive DNS analysis. In *NDSS 2011, 18th Annual Network and Distributed System Security Symposium, 6-9 February 2011, San Diego, CA, USA*, 2011.

[4] Karasaridis, A.; Meier-Hellstern, K.; Hoeflin, D.: NIS04-2: Detection of DNS Anomalies using Flow Data Analysis. In *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, 2006, ISSN 1930-529X, s. 1–6.

[5] Kováčik, M.: Liberouter: DNS plugin [online]. [cit. 2014-06-24]. URL <https://www.liberouter.org/technologies/dns-plugin/>

[6] Kováčik, M.: *Detekce síťových anomálií a bezpečnostních incidentů s využitím DNS dat*. Pojednání k tématu disertační práce, Fakulta informačních technologií VUT v Brně, Brno, CZ, 2014.

[7] Kováčik, M.; Kajan, M.; Žádník, M.: Detecting IP-spoofing by modelling history of IP address entry points. In *Emerging Management Mechanisms for the Future Internet*, *Lecture Notes in Computer Science 7943*, ročník 2013, Springer Verlag, 2013, ISBN 978-3-642-38997-9, ISSN 0302-9743, s. 73–83.

[8] Qi, C.; Chen, X.; Xu, C.; aj.: A Bigram based Real Time DNS Tunnel Detection Approach. *Procedia Computer Science*, ročník 17, 2013: s. 852 – 860, ISSN 1877-0509.

[9] Roolvink, S.: Detecting attacks involving DNS servers : A netflow data based approach. 2008. URL <http://essay.utwente.nl/58497/>

[10] Villamarin-Salomon, R.; Brustoloni, J. C.: Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic. In *Consumer Communications and Networking Conference, 2008. CCNC 2008. 5th IEEE*, 2008, s. 476–481.

ENERGETICKY-AUTONÓMNY BIOMONITOROVACÍ SYSTÉM

Gabriel Nagy
Mikroelektronika, 2. ročník, denná prezenčná forma štúdia
Školiteľ: Viera Stopjaková

Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave
Ilkovičova 3, 812 19 Bratislava

gabriel.nagy@stuba.sk

Abstrakt. Tento príspevok sa zaoberá návrhom základného konceptu energeticky autonómneho biomonitorovacieho systému, so zameraním najmä na využite ľudského tela ako jedného z možných zdrojov energie pre bezdrôtové biosenzory umiestnené na tele, ale aj pre ostatné biomonitorovacie zariadenia. Ako primárny zdroj energie je uvažovaný rozdiel teplôt medzi ľudským telom a okolitým prostredím. Dôležitými faktormi pre energetický zdroj na báza teplotného rozdielu je jeho umiestnenie na tele a samotné klimatické prostredie, v ktorom sa bude pohybovať monitorovaná osoba. V rámci tejto fázy štúdia bol vykonaný aj návrh napäťového meniča, ktorý je dôležitou časťou tzv. *energy harvesting* bloku.

Kľúčové slová. Získavanie energie, alternatívne zdroje energie, nízko-energetický návrh

1 Úvod

Zdravotná starostlivosť patrí dlhodobo medzi hlavné priority každej vyspelej spoločnosti. V prípade dlhodobých ochorení si kvalitná starostlivosť zväčša vyžaduje pravidelné návštevy zdravotných zariadení. Možným riešením ako minimalizovať osobné návštevy u lekára a zabezpečiť domácu starostlivosť je nepretržité monitorovanie pacienta pomocou biomonitorovacích systémov. Tieto zariadenia sú kvôli mobilite pacienta zvyčajne bezdrôtové a podľa možnosti minatúrne. Kominukácia s mobilnými telefónmi prostredníctvom bezdrôtových technológií tak umožňuje zasielanie monitorovaných údajov priamo do zdravotných stredísk [1]. Aspekty, ktoré treba pri návrhu takéhoto systému zvážiť sú: mobilita pacienta, hmotnosť zariadenia, výdrž batérií, potreba zásahu pacienta do činnosti a údržby zariadenia, či umiestnenie snímačov a zariadenia. Možnosti napájania biomonitorovacích zariadení sú ovplyvnené najmä ich umiestením. V prípade implantovaných biosenzorov je výmena batérií takmer vylúčená. Preto je nevyhnutné, aby implantované elektronické zariadenia mali minimálnu spotrebu elektrickej energie, prípadne boli aspoň čiastočne energeticky-autonómne [2]. Aj pri biosenzoroch umiestnených na tele je dôležitá minimálna spotreba energie a minimálna potreba interakcie pacienta so zariadením. Potrebu batérií ako aj ich výmenu je možné úplne eliminovať, ak bude monitorovací systém napájaný z okolitého prostredia.

Vzhľadom na cieľovú aplikáciu sa v našej práci zameriavame primárne na využiteľnosť energie z ľudského tela, teda z rozdielu teplôt medzi telom a jeho okolím. Energeticky-autonómny biomonitorovací systém by pozostával z troch hlavných častí. Výkonová časť slúži na získavanie a transformáciu energie. Druhá časť je tzv. výpočtová a zabezpečuje snímanie parametrov, predspracovanie nameraných dát a bezdrôtovú komunikáciu. Poslednou časťou je zásobník energie (napr. batéria), ktorá nepretržite dodáva energiu všetkým častiam systému [3].

V sekcii 2 je uvedené povrchové rozloženie teploty na ľudskom tele a je analyzovaná použiteľnosť teplotného rozdielu ako zdroja energie pre monitorovacie zariadenie. V sekcii 3 je uvedený principiálny návrh výkonovej časti energetického meniča. Súčasne sú tu predstavené výsledky simulácií ako aj reálne údaje v súvislosti s možnou budúcou prototypovou výrobou systému. Sekcia 4 predstavuje rámcové ciele dizertačnej práce a ich doterajšie plnenie. Posledná sekcia prináša zhrnutie.

2 Použiteľnosť teplotného rozdielu ako zdroja energie

Na vyhodnotenie rozdielu teplôt medzi ľudským telom a jeho okolím ako možného energetického zdroja pre energy harvesting systém je potrebné poznať reálne hodnoty rozdielu teplôt. Povrchová teplota ľudského tela pri teplote okolia 25 °C a vnútornej telesnej teplote 36,7 °C sa pohybuje v rozsahu od 28,2 °C až po 34,4 °C, a to v závislosti od konkrétneho miesta na tele (viď. Tab. 1 [4]). Na základe údajov z Tab. 1 sa ako najvhodnejšia pozícia pre umiestnenie energetického meniča založeného na rozdiele teplôt medzi ľudským telom a jeho prostredím javí byť brucho monitorovanej osoby. Táto časť tela však za normálnych okolností žiaľ nie je v priamom kontakte s prostredím. Preto je vhodnejšie zamerať sa na tie časti tela, ktoré sú bežne vystavené kontaktu s okolím. Takou časťou je práve ruka, konrétne zápästie, kde by bolo možné umiestniť celé zariadenie vo forme napr. náramku, ktorý nespôsobuje monitorovanej osobe žiadne obmedzenie pri každodennej činnosti.

Poloha	**°C**	**Poloha**	**°C**	**Poloha**	**°C**
Stredová os tela		**Ľavá/Pravá strana tela**			
		- predná strana		**- zadná strana**	
čelo	31,6	krk	32,3	lopatka	33,3
druhé rebro	30,3	horná časť hrudníka	33,7	pás	33,7
štvrté rebro	32,1	dolná časť hrudníka	33,8	pozadie	30,2
koniec hrudného koša	33,2	rebrá	33,4	stehno	31,2
7,5 cm nad pupkom	34,4	pás	33,1	lýtko	28,2
3,0 cm pod pupkom	33,4	stehno	30,9	ruka	32,5
11,5 cm pod pupkom	31,8	holeň	30,4		

Tab. 1: Povrchová teplota vybraných častí ľudského tela [4]

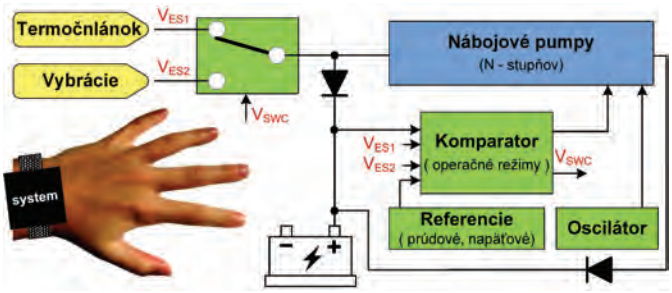
Pri návrhu systému, ktorý získava energiu z prostredia je potrebné poznať či je zvolený zdroj energie stabilný. V tomto prípade ide o premenlivosť teploty okolia v priebehu roka [5] i počas dňa. Počas roku 2013 neprekročila maximálna teplota vzduchu hranicu 25 °C počas približne 80 % dní. Súčasne ani počas jedného dňa nebola najnižšia denná teplota vzduchu viac ako 22 °C. Podrobnejšia analýza priebehu teplôt ako aj použiteľnosti tohto zdroja energie je prezentovaná v [3].

3 Realizácia energy harvesting časti

Na základe získaných informácií a parametrov komerčne dostupných peltierových článkov sa domnievame, je možné zhotoviť bezdrôtové monitorovacie zariadenie s energy harvesting systémom s rozmerom bežných náramkových hodiniek. Navrhnutá bloková schéma výkonovej časti energetického meniča je zobrazená na Obr. 1. Termočlánky uvedené v Tab. 2 by pri teplotnom rozdieli 5 °C mali generovať napätie naprázdno v rozsahu od 20 mV až do 200 mV.

3.1 Riadenie činnosti energetického meniča

Základná požiadavka pre napájanie monitorovacieho systému je jeho schopnosť automaticky sa prispôsobiť aj neoptimálnym podmienkam. Ak slúži rozdiel teplôt ako primárny zdroj energie, takýto prípad



Obr. 1: Výkonová časť energy harvesting systému

	[6]	[7]	[8]	[9]
ΔT (°C)	68	68	67	100
I_{max} (A)	3,30	8,50	3,90	0,37
V_{max} (V)	8,1	8,6	15,4	4,5
R_{ser} (Ω)	1,80	0,85	3,10	12,4
$Rozmer$ (mm x mm)	30x30	30x30	30x30	40x40

Tab. 2: Porovnanie parametrov bežných termočlánkov (podľa technickej dokumentácie)

nastáva práve počas horúcich letných dní v mesiacoch v období jún až august [5]. Najmä vtedy môže poklesnúť rozdiel teplôt pod hranicu 5 °C. Práve pre takéto prípady je systémy vybavený zásobníkom energie (batériou), ktorého kapacita je však obmedzená. Preto je vhodné okrem primárneho zdroja uvažovať aj zdroj sekundárny, kde by bolo možné využiť buď solárnu energiu alebo energiu z pohybu ruky pri rôznych činnostiach. Týmto spôsobom by sme mali byť schopní získavať energiu nepretržite. Na základe týchto úvah je potrebné rozdeliť činnosť energy harvesting systému na niekoľko pracovných režimov, ktorú budú riadené komparátormi. Prehľad pracovných režimov je uvedený v Tab. 3.

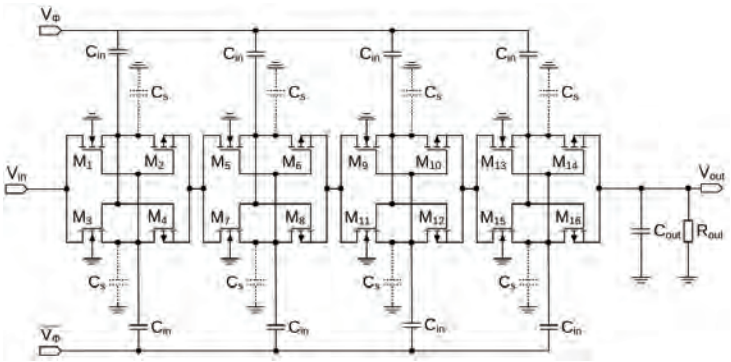
Režim	Podmienka	Činnosť/Stav systému
1	$V_{ES1} < V_{Hranica}$ a súčasne $V_{ES2} < V_{Hranica}$	Systém prejde do stavu spánku.
2a	$V_{ES1} > V_{Hranica}$	Dobíjanie batérie.
2b	$V_{ES1} < V_{Hranica}$ a súčasne $V_{ES2} > V_{Hranica}$	Dobíjanie batérie.
3	$V_{bateria}$ = úplne nabitá	Systém prejde do stavu spánku.

Tab. 3: Režimy činnosti energy harvesting systému

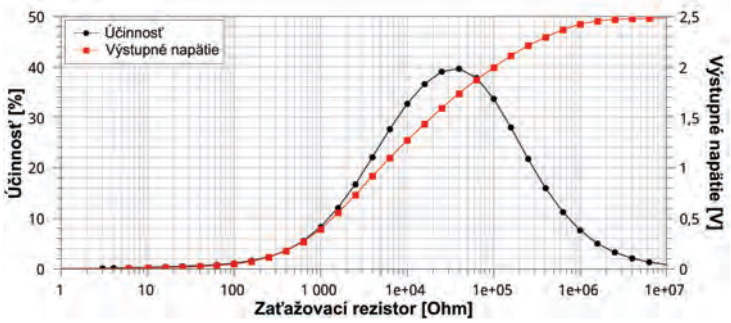
3.2 Návrh nábojových púmp

Počas doterajšieho výskumu sme sa venovali aj návrhu nábojovej pumpy a jej implementácii priamo na čipe, nakoľko táto je základnou súčasťou napäťových meničov. Nábojová pumpa s naprieč spínanými kondenzátormi (Obr. 2), navrhnutá v 90 nm technológii, vykazuje najlepšie parametre [10]. Na základe rovnice (1) je možné odhadnúť napäťové straty v tejto nábojovej pumpe, kde R_{onN} reprezentuje odpor kanála NMOS tranzistora v zapnutom stave a R_{onP} označuje rovnaký parameter pre PMOS tranzistor. C_{in} reprezentuje kapacitu spínaného kondenzátora a C_s predstavuje parazitné kapacity. Tieto štyri uvedené parametre sú uvažované vzhľadom na jeden stupeň nábojovej pumpy a výsledná rovnica uvažuje rovnaké rozmery prvkov vo všetkých stupňoch. Počet stupňov nábojovej pumpy je označený ako N, a stupne sú spínané neprekrývajúcimi sa signálmi o frekvencii f. Napäťové signály sú nasledovne: vstupné napätie je označené ako V_{in}, amplitúda spínacieho signálu je V_Φ a výstupné napätie je označené ako V_{out}. Hlavné dosiahnuté parametre navrhnutej nábojovej pumpy sú zobrazené na Obr. 3.

$$V_{out} \approx V_{in} + N.\left(V_\phi - I_{out}.(R_{onN} + R_{onP}) - \frac{I_{out}}{(C_{in} + C_s).f}\right) \quad (1)$$



Obr. 2: Nábojová pumpa s naprieč spínanými kondenzátormi



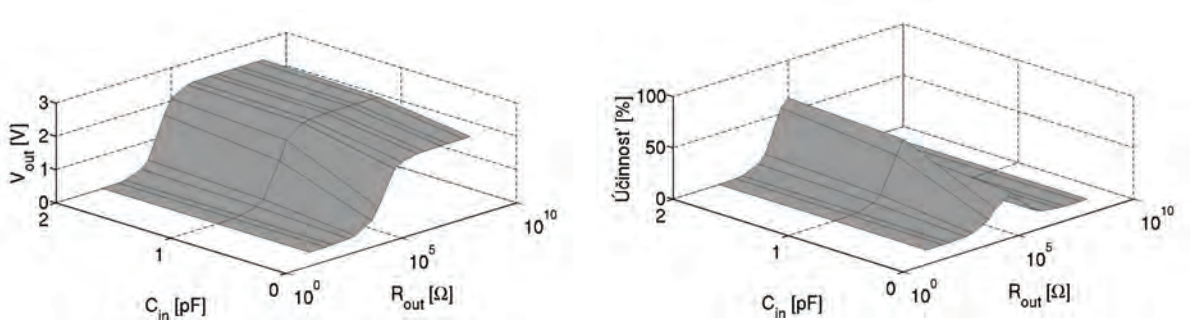
Obr. 3: Závislosť výstupného napätia a účinnosti od zaťažovacieho rezistora

Neoptimalizovaná nábojová pumpa bola navrhnutá v $90nm$ CMOS technológii. Rozmery použitých PMOS tranzistorov boli 100 μm/0,1 μm a pre NMOS tranzistory rozmery sú 50 μm/0,1 μm (šírka kanála/dĺžka kanála). Vstupné napätie (V_{in}) = 500 mV. Hodnota spínaného napätia (V_Φ) bola 500 mV so striedou signálu 1:1. Maximálna dosiahnutá hodnota výstupného napätia je 2,48 V a maximálna dosiahnutá účinnosť je 40 % pri 1,74 V na výstupe. Plošne najnáročnejším prvkom sú spínané kondenzátory. Maximálny výstupný výkon je 217 μW (aj pri spínacej frekvencii iba 1 MHz). Každý z kondenzátor s kapacitou 1 nF zaberá plochu približne 0,5 mm^2. Spolu to predstavuje zrejme neakceptovateľnú plochu 4 mm^2, pričom plocha všetkých použitých MOS tranzistorov nepresahuje 200 μm^2.

Za účelom redukcie plochy čipu bola vykonaná optimalizácia pumpy pre nájdenie kompromisu vzhľadom na plochu, výstupný výkon a samozrejme účinnosť pumpy. Spínacia frekvencia bola 100-násobne zvýšená čo umožnilo výrazne zmenšenie spínaných kondenzátorov. Taktiež boli zmenšené aj rozmery použitých MOS tranzistorov. Na Obr. 4a je zobrazené výstupné napätie a na Obr. 4b účinnosť pumpy pre viaceré hodnoty spínaných kondenzátorov a v závislosti od hodnoty zaťažovacieho rezistora (R_{out}) na výstupe. Hodnota kapacity výstupného kondenzátora bola 10 pF a spínacie kondenzátory mali hodnoty v rozsahu od 0,2 pF do 2 pF. Rozmery použitých PMOS tranzistorov boli 20 μm/0,1 μm a rozmery NMOS tranzistorov boli 10 μm/0,1 μm. Spínacia kapacita 200 fF je už porovnateľná s parazitnými kapacitami, čo výrazne zvyšuje straty, ako dokazujú uvedené závislosti.

3.3 Experimentálne overenie vhodnosti zvoleného zdroja energie

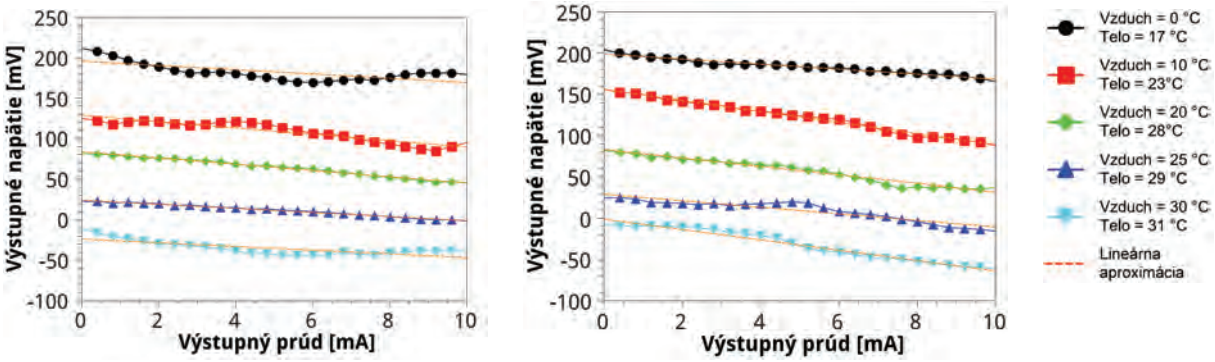
Pre potvrdenie odhadov na základe technickej dokumentácie bolo vykonané experimentálne overenie použiteľnosti najdostupnejších komerčných termočlánkov. Jednalo sa o dva termočlánky uvedené v Tab. 2



(a) Výstupné napätie (b) Účinnosť

Obr. 4: Parametre 4-stupňovej nábojovej pumpy pre rôzne spínacie kondenzátory a záťažový rezistor

([6, 8]). Naprázdno by mali byť schopné dodávať napätie približne 30 mV pri rozdiele teplôt približne 4 °C. Teplota vzduchu počas merania bola 25 °C a povrchová teplota tela na zápästí pod termočlánkom bola 29 °C. Pri klesajúcej teplote vzduchu, klesala aj povrchová teplota tela. Kvôli potlačeniu vplyvu chladiča a jeho tepelnej kapacity sme čakali 2 minúty než sa považovala teplota chladiča za ustálenú. Volt-ampérové charakteristiky uvedených termočlánkov boli odmerané pri teplote vzduchu 30, 25, 20, 10 a 0 °C. Pre tieto hodnoty okolitej teploty boli namerané nasledujúce povrchové teploty na zápästí: 31, 29, 28, 23 a 17 °C. Termočlánky TEC1-07103 a TES1-12704 majú podobné charakteristiky aj keď ich parametre nie sú identické ako ukazujú závislosti na Obr. 5.



(a) TEC1-07103 (b) TES1-12704

Obr. 5: Volt-ampérové charakteristiky termočlánkov pri teplote vzduchu 30, 25, 20, 10 a 0 °C

4 Ciele dizertačnej práce

Na základe doteraz vykonanej analýzy súčasného stavu a potrieb v oblasti energeticky úsporných a energeticky autonómnych integrovaných systémov, ako aj z nej získaných poznatkov a dosiahnutých výsledkov prezentovaných v tomto príspevku, boli ciele nášho výskumu reprezentujúce rámcové tézy dizertačnej práce (a ich doterajšie plnenie) stanovené nasledovne:

- Preskúmať a porovnať alternatívne zdroje energie z hľadiska možnosti ich implementácie priamo na čipe a analyzovať reálnosť zabezpečenia čiastočnej energetickej autonómnosti integrovaných systémov (*splnené*).
- Vyhodnotiť vhodnosť možných zdrojov energie pre bezdrôtové zariadenia umiestnené na ľudskom tele, napr. prenosné monitorovacie a zdravotnícke systémy (*splnené*).
- Navrhnúť a optimalizovať systém na získavanie energie z okolia využívajúci kombináciu viacerých

zdrojov energie (*rozpracované*).

• Vypracovať metodiku pre návrh systému na získavanie energie z alternatívnych zdrojov priamo na čipe s možnosťou čiastočnej automatizácie návrhu (*nezačaté*).

• Implementovať a experimentálne overiť miniatúrny energy harvesting systém pre biomonitorovacie zariadenie (*nezačaté*).

5 Záver

V tomto príspevku bolo uvedené principiálne riešenie výkonovej časti energy harvesting systému. Hlavným zameraním bolo poukázať na dôležité stránky a faktory, ktoré je treba pri návrhu energeticky-autonómneho systému zohľadniť. Takými sú napríklad jeho umiestenie či vlastnosti prostredia.

V rámci doterajšej práce na návrhu nízko-príkonových obvodov, ktoré sú nevyhnutné pre energeticky-autonómne aplikácie vzniklo spolu doteraz 14 publikácií, na ktorých som autorom resp. spoluautorom (2 články v karentovaných a impaktovaných vedeckých časopisoch, 10 príspevkov na medzinárodných konferenciách a sympóziách a 2 príspevky na domácich konferenciách) .

Poďakovanie

Tento príspevok vznikol vďaka podpore v rámci OP Výskum a vývoj pre projekt: Kompetenčné centrum inteligentných technológií pre elektronizáciu a informatizáciu systémov a služieb, ITMS: 26240220072, spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

Literatúra

[1] S. Rajasekaran, P. Kumaran, G. Premnath, and M. Karthik, "Human Health Monitoring Using Wireless Sensor Networks (WSN)," *International Journal of Application or Innovation in Engineering & Management (IJAIEM)*, vol. 2, no. 12, pp. 323–330, 2013.

[2] K. Bazaka and M. V. Jacob, "Implantable devices: issues and challenges," *Electronics*, vol. 2, no. 1, pp. 1–34, 2012.

[3] G. Nagy and V. Stopjaková, "Human body as an energy source for a wireless boimonitoting," *6th Biomedical Engineering Conference of Young Biomedical Engineers and Researchers 2014*, Bratislava, Slovakia, pp. –, 2014.

[4] F. G. Benedict, W. R. Miles, and A. Johnson, "The temperature of the human skin," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 5, no. 6, p. 218, 1919.

[5] SHMU, *Climatological services/Climagrams/Air temperature*. Slovak hydrometeorological institute (Slovenský hydrometeorologický ústav), 2014.

[6] TEC1-07103, *Datasheet*, 2009.

[7] TEC1-07108, *Datasheet*, 2009.

[8] TES1-12704, *Datasheet*, 2005.

[9] TGM-287-1.0-2.5, *Datasheet*, 2010.

[10] G. Nagy and V. Stopjaková, "Analysis and Evaluation of Charge-pumps Realizable in 90nm CMOS Technology," *24th International Conference Radioelektronika 2014*, Bratislava, Slovakia, pp. –, 2014.

SYNCHRONIZATION METHODOLOGY FOR FAULT TOLERANT SYSTEM RECOVERY AFTER ITS FAILURE

Karel Szurman
Computer Science and Engineering, 2-st class, part-time study
Supervisor: Zdeněk Kotásek

Faculty of Information Technology, Brno University of Technology
Bozetechova 1/2, 612 66 Brno

iszurman@fit.vutbr.cz

Abstract. Modern fault tolerant systems implemented into FPGAs integrate very often hardware redundancy together with fault tolerant approaches based on active fault recovery and the system reconfiguration. An integral part of the recovery process in these systems is except of fault-masking behavior and FPGA partial reconfiguration also the synchronization of reconfigured circuit copy with remaining circuits which are during the recovery process still operating. In the paper, basic principles of our synchronization methodic are described together with generic architecture for synchronization in fault tolerant systems. The usage of the generic architecture for synchronization is demonstrated by its implementation into reconfigurable fault tolerant CAN bus control system.

Keywords. Fault tolerant system, FPGA, state synchronization, recovery, partial dynamic reconfiguration, failure.

1 Introduction

An increasing number of safety-critical systems use active fault tolerant techniques. The main reason is the active approach can ensure the system operability while faults are present in the system environment together with its complete recovery in the case when the system failure occurs. Such demands have fault tolerant systems (FTSs) e.g. in space applications where reprogrammable FPGAs being more often used. These FPGAs are based on flash or SRAM technology. The flash-based FPGAs have non-volatile configuration memory and they are more robust against radiation effects (such are Single Event Effects) when comparing to FPGA devices based on SRAM cells. Nevertheless, SRAM FPGAs are not limited in number of programming cycles. Actual trend is to combine both types of FPGA devices [6]. In SRAM FPGAs, hardware redundancy can be easily combined with the reconfiguration process to achieve the correct system functionality. The most used form of hardware redundancy is triple modular redundancy (TMR) due to its fault-masking ability and tolerable overhead. Active FTSs based on TMR architecture (or N-modular redundancy in general) are often implemented as reconfigurable because the fault tolerance of the TMR architecture is ensured only for the class of expected failures and after the failure occurs, it loses fault mitigation ability. Fault detection in a TMR is operating by means of majority voting from copies of protected circuit. When the failure in a one from circuit copies is detected then corresponding TMR module located in FPGA configuration memory is reconfigured through partial dynamic reconfiguration (PDR) process. After the reconfiguration process is finished, its operational state is not up-to-dated and need to be synchronized with the correctly operating circuit copies in TMR architecture before it is incorporated back into the system.

Two main approaches to state synchronization are often used. The representative one is based on sharing of the system state between all redundant copies of the protected circuit. In [3], the soft processors are combined in TMR architecture and their context is shared through the Block RAM memory in FPGA. Then, after the failed soft processor is reconfigured, the interrupt routine is used for its synchronization with others. The main benefit of this method is the recovery process can be performed on the fly and overhead of the synchronization is only the time required to store and restore the processor's state context. The second approach uses the principle, where the synchronization is performed as the copy of the state from operating reference circuit to the reconfigured circuit. In [5], the method based on the principle of roll-forward recovery was used. Through copying of all data registers from the correct circuit copy into the failed copy, the state correction was achieved. In [4], the synchronization for FSM-based system is presented. The method uses the principle of predicting a future state (checkpoint state) to which the system will soon converge and presetting the reconfigured circuit to it.

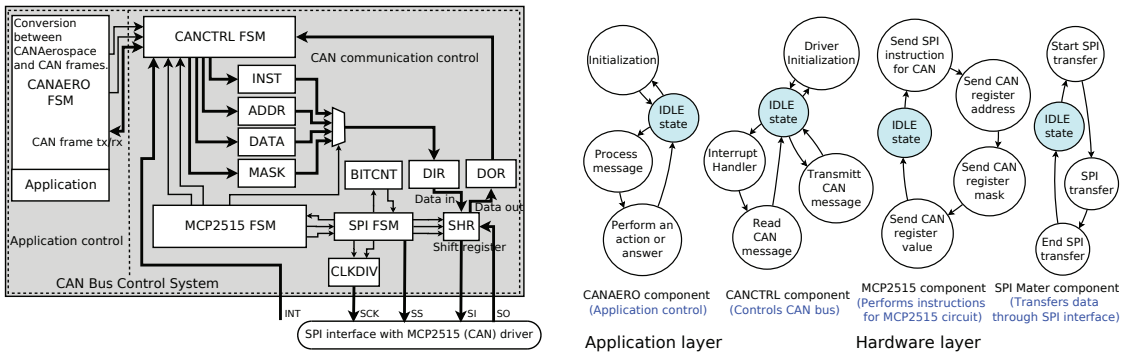
A recovery workflow in active FTSs consists of fault detection, failed circuit reconfiguration and circuit state synchronization phases. My Ph.D thesis is focused on a phase of the state synchronization. The aim is to develop a new methodology for the design and implementation of a suitable synchronization method for specific FTS implemented into SRAM FPGA. In this paper, fundamental considerations related to our synchronization methodology are described. The paper is organized as follows. First, generic architecture for system synchronization by copying of its state is proposed and implementation of its principles into designed reconfigurable fault tolerant CAN bus control system is described. Then, basics of our synchronization methodology together with my previous work and goals of my Ph.D thesis are presented in following chapters as well.

2 Generic Architecture for the State Synchronization Implemented Into Reconfigurable Fault Tolerant CAN Bus Control System

In our research, we concentrated on synchronization methods for FSM-based systems so far. As the first step, we developed specific synchronization methods for reconfigurable FTS including fault tolerant CAN bus control system (FTCAN) and our generic partial dynamic reconfiguration controller (GPDRC), which were described further in [1] and [2]. The aim was to enable recovery of the failed circuit copy through the reconfiguration process and to design suitable method for the synchronization process of circuits after the recovery.

Before implementation of the synchronization into the FTS, it was necessary to analyze all internal states in the core of the CAN bus control system. The architecture of the control system and overview of its main control states is shown in Fig. 1. Because the control system is divided into application and hardware control parts, we decided for two different approaches to synchronize application and hardware layer of the system, the reasons are as follows:

- Hardware layer is controlled by the CANCTRL unit. This unit processes incoming interrupt requests or it performs control communication with the circuit MSP2515 of the CAN controller. Otherwise, the unit is in inactive state and waits for its activation. Because the inactive state is always reached and transmitting of control command sequence to the MCP2515 is relatively fast, the best method for the synchronization of the CANCTRL and lower level components is to wait until the reference circuit is not preset to IDLE (inactive) state and then, the recovered circuit synchronize with it.
- Application layer is controlled by the CANAERO unit. It executes initialization sequence of the MCP2515 circuit, processes messages in CANAerospace application protocol and performs corresponding actions which are encoded in received messages. The unit contains data registers with values which are acquired during the unit operation and mathematical calculations. The data context of this unit depends on its previous actions, therefore its synchronization is based on copying the state from reference to synchronized unit.



(a) Architecture of the CAN bus control system

(b) States of FSMs used in the control system

Figure 1: Architecture of the CAN bus control system and overview of its states within control FSMs.

2.1 Generic Architecture

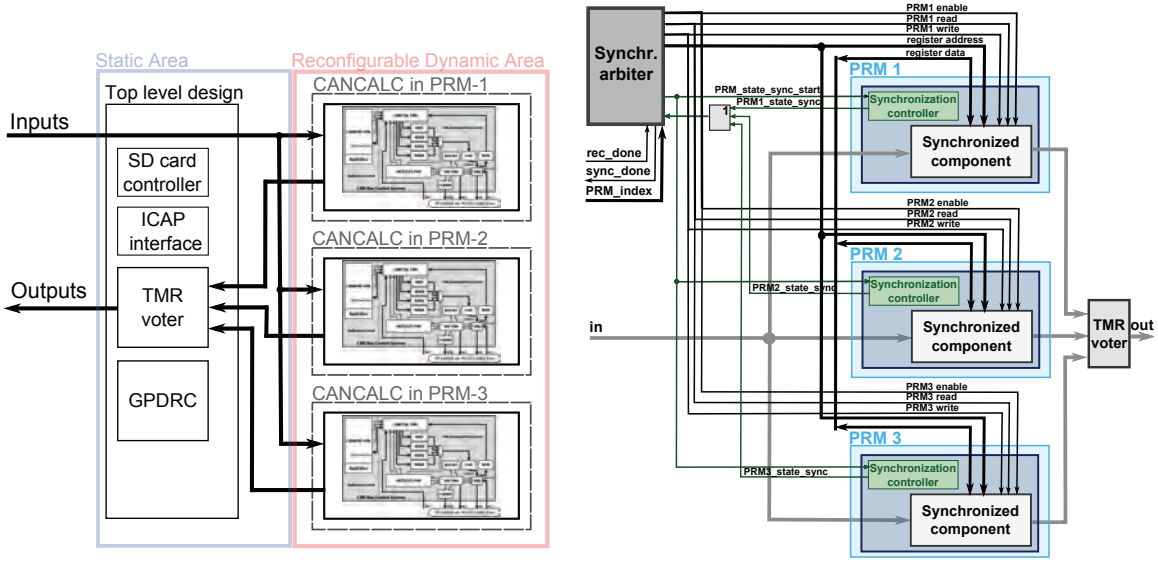
On the basis of results gained from analysis of architecture designed FTS we decided that synchronization process of the reconfigured circuit copy in the TMR architecture has to be controlled on two levels. From the outside, the synchronization should be controlled on the level of individual circuits, and in the inside on the level of circuits for synchronization of their internal components. We designed generic architecture for synchronization with arbiter and controllers. The architecture consists of the following parts:

- **Synchronization arbiter** - it is a hardware unit which is responsible for controlling the complete synchronization process from the highest architecture level. After the reconfiguration of the failed circuit is finished the arbiter is activated. Before the synchronization begins the arbiter identifies the specific roles of all redundant circuits within the TMR. It indicates which circuit is synchronized, which circuit is used as reference for copying of its state into synchronized one and which circuit is paused during this process. Then, the synchronization procedure is started and the arbiter communicates with synchronization controllers and synchronously controls all phases of the synchronization process until the reconfigured circuit is fully synchronized with other circuits. Finally the arbiter switches all circuits into operational state.
- **Synchronization controller** - it is a hardware unit which is implemented into each PRM with replicated circuit in the TMR. Its role is to control the synchronization of internal components and subsystems of the circuit. The controller communicates with the arbiter during the synchronization process. It addresses individual components of the circuit and their internals for the synchronization. According to the role of the circuit (reference, synchronized or paused) during the synchronization, the controller can executes a) the transmission of a state information in the reference circuit, b) the reception of state information and its saving into internal registers or c) suspension of the units which have no role in synchronization process until it is not finished. After the controller finishes its function it alerts the arbiter which will perform another steps of synchronization.
- **Synchronization bus** - it consists of wire interconnections for transferring control and data signals between all redundant circuit copies. Its complexity depends on requirements for the synchronization process, especially the speed of the synchronization or the implementation area overhead.
- **Synchronization interface** - it is a communication interface used for data transfers from or into the circuit components during the synchronization process. This interface is implemented for all data registers which hold the state information in the replicated circuit of TMR architecture.

Implementation of some previously mentioned functionalities can be merged or placed in other components than is declared above. For some less complex systems requirements for the synchronization can be reduced.

2.2 Recovery Process for Reconfigurable Fault Tolerant CAN Bus Control System

The architecture of our reconfigurable fault tolerant CAN bus control system is shown in Fig. 2. Redundant copies of the CAN control system are placed in reconfigurable dynamic area into separated partial reconfigurable modules (PRMs). Into the static non-reconfigurable area the units responsible for the control of recovery process are programmed. The static area includes the GPDRC, bitstream storage controller with interface to SD card, the ICAP interface and the synchronization arbiter for the control of the synchronization process. In the static area, TMR voter is also located which besides the fault-masking also identifies the failed circuit copy when a fault is detected. This information is passed into the GPDRC which starts the reconfiguration of a PRM where the failure was localized. After the reconfiguration of certain circuit copy is finished, the synchronization arbiter is activated and the synchronization procedure to return the TMR into full operation is performed.



(a) TMR architecture and FTCAN layout in FPGA

(b) Parallel synchronization method for FTCAN

Figure 2: Reconfigurable FTCAN and its synchronization through parallel bus.

As was mentioned in the beginning of the chapter 2, two different approaches to synchronize application and hardware layer of the system were used. The hardware layer is synchronized by the state synchronization procedure. The principle of the procedure is based on waiting until the reference circuit reaches the specified state and then this state is preset to the reconfigured circuit. This mechanism is implemented on the level of CANCTRL unit where the synchronization controller is placed. When the start of this phase is indicated by PRM state sync start signal, the controller starts waiting for the IDLE state in the reference circuit. When it is reached, the synchronization controller indicates it by PRM state sync signal to the synchronization arbiter. The arbiter stops execution in all units of the synchronized system and then it enters the second phase - the data context synchronization of the application layer.

Synchronization process of the second phase was implemented in serial and parallel versions. The difference between these variants is in the size of the bus and control logic. Obviously, the serial implementation is slower and simpler, but the principle is the same. Therefore only parallel version is described here. The parallel version of the synchronization uses two parallel buses for data transfers between registers in reference and synchronized circuits. Individual registers are addressed through the address bus. The scheme with implemented synchronization into FTCAN design is shown in Fig. 2b. The principle of the synchronization lies on sequential addressing of each register through the address bus and enabling PRM write or PRM read signals for circuits which are active during the synchronization

process. Reference circuit transfers the content of its addressed registers to the data bus byte after byte while the synchronized circuit reads these data from the bus and stores them into its internal registers.

3 Synchronization Methodology

Based on the experience we gained during the development and the implementation of the synchronization method for reconfigurable fault tolerant CAN bus control system we determined the set of essential questions which must be considered and then satisfied in a certain way by designed synchronization method implemented into an FTS in general. The essential questions are as follows:

1. The state in which the synchronization of the recovered unit is performed.
2. The definition of the system context (i.e. the set of data) which will be used for the synchronization.
3. The problem of the interconnection of redundant components which will be needed for the synchronization procedure execution and its control.

3.1 Parameters of Synchronization Methods

The synchronization method development and its implementation are closely combined with the architecture of the FT system and its complexity, requirements on its real-time behavior, with principles of performing its function and the type of volume of the synchronized context. This fact is apparent from essential questions which were declared. All these aspects must be taken into account when the method of system synchronization after fault occurrence is developed. Thus, the principles of synchronization and its specific implementation have a strong impact on the FT system and its parameters. These are:

The dynamic parameters reflect the impact of the synchronization on the operation of the system and its function. From among them, the following dynamic parameters can be mentioned:

- **The impact on the function of the system** - it says whether during the synchronization the system requires to be stopped or the synchronization can be completed while the system is running. Based on this criterion, the synchronization methods can be divided into function blocking and function non-blocking methods.
- **The time needed to perform the synchronization** - it is closely combined with other parameters, the requirements on the synchronization and the volume of data which needs to be synchronized.

The static parameters have an indirect impact on system features and have a close relation to the principles of the algorithm used to implement the synchronization procedure. The static parameters are as follows:

- **The area demands on the implementation** - it reflects the overhead costs of FPGA sources needed to implement the synchronization method.
- **The power demands** - it determines the power needed to be delivered to the synchronization system.
- **The reliability of the synchronization implementation** - it is related to the reliability of the synchronization system. Apart from dynamic and static parameters which can be affected by the selected synchronization method, other aspects must be taken into account. They have a close relation to synchronization implementation into FTS and the individual steps of the method.

4 Conclusions and Future Research

In this paper, considerations for the synchronization of FSM-based systems were summarized. On the basis of results from experiments and new knowledge which I gained so far my future research will be based. It will be focused on design and implementation of synchronization methods for active FTSs

with more complex architectures and systems based on soft-processors. Proposed generic architecture for synchronization will be used. Possibly, new architecture for synchronization will be designed due to specific demands, architecture and behavior of these systems. Then, according to new results from research I will be able to compare requirements for synchronization methods in simple and complex FTSs using various components with different types of data and control flow.

5 Ph.D Thesis Goals

The synchronization method must be devised in the way which will allow to identify and implement the best possible principles of synchronization for the given architecture, real time requirements and price (overheads). Therefore the main aim of my Ph.D thesis is to propose specific methodology to design and implement synchronization procedure for target FTS. Basic goals of my Ph.D thesis are as follows:

1. To combine fault tolerant system with partial reconfiguration controller to enable the ability of active recovery of a part of the system where the failure was detected.
2. To propose the methodology for the synchronization of a simple and FSM-based FTSs.
3. To propose the methodology for synchronization of complex FTSs and systems based on soft-processors (such as Xilinx PicoBlaze, Xilinx MicroBlaze and LEON3). This part of the methodology should consider specifics of soft-processors and also granularity of components in a FTS.
4. To develop part of methodology on the assessing of a designed synchronization method on the basis of its parameters and requirements. Therefore, several critical parameters of the system where the synchronization is implemented were defined with the goal to allow and simplify the classification of synchronization methods.

Acknowledgments

This work was supported by National COST project LD12036-"Methodologies for Fault Tolerant Systems Design Development, Implementation and Verification"; project Centre of excellence, IT4Innovations (ED1.1.00/02.0070); Project No. MSM 0021630528-"Security-Oriented Research in Information Technology" and project FIT-S-14-2297.

References

[1] K. Szurman, J. Kastil, M. Straka and Z. Kotasek, "Fault Tolerant CAN Bus Control System Implemented into FPGA," In the IEEE International Symposium on Design and Diagnostics of Electronic Circuits and Systems, Karlovy Vary, CZ, 2013, pp. 289–292, ISBN 978-1-4673-1185-4.

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

[3] S. Tanoue, T. Ishida, Y. Ichinomiya, M. Amagasaki, M. Kuga and T. Sueyoshi, "A novel states recovery technique for the TMR softcore processor," In the International Conference on Field Programmable Logic and Applications, Praha, CZ, 2009, pp. 543–546, ISSN 1946-1488.

[4] J.R. Azambuja, F. Sousa, L. Rosa and F.L. Kastensmidt, "Evaluating large grain TMR and selective partial reconfiguration for soft error mitigation in SRAM-based FPGAs," In the 15th IEEE International On-Line Testing Symposium, Lisbon, PT, 2009, pp. 101–106, ISBN 978-1-4244-4596-7.

[5] S.-Y. Yu and E.J. McCluskey, "On-line testing and recovery in TMR systems for real-time applications," Test Conference, 2001. Proceedings. International, Baltimore, MD, 2009, pp. 240–249, ISBN 0-7803-7169-0.

[6] I. Kuon, R. Tessier and J. Rose, "FPGA Architecture: Survey and Challenges," Foundations and Trends in Electronic Design Automation, Vol. 2 No. 2, 2008 , pp. 135–253, ISSN 1551-3939.

Optimalizace synchronizační kominukace v DFS

Jindřich Skupa
Inženýrská informatika, 2. ročník, prezenční
Školitel: prof. Ing. Jiří Šafařík CSc.

Fakulta aplikovaných věd, Západočeská univerzita
Univerzitní 22, 306 14 Plzeň

skupaj@kiv.zcu.cz

Abstrakt. Článek rozebírá problematiku distribuovaných systémů a komunikace v distribuovaných systémech. Obsahem je úvod do problematiky distribuovaných souborových systémů, rozbor dostupných řešení replikace dat a představení experimentálního distribuovaného souborového systému KIVFS. Následně jsou popsány aktuálně používané algoritmy s jejich negativním dopadem na propustnost systému a návrhy na jejich optimalizaci.

Klíčová slova. distribuované souborové systémy, synchronizace, distribuované transakce, směrování zpráv, kivfs

1 Úvod

Aktuálním trendem v oblasti informačních technologií je zpracování velkého množství dat a jejich ukládání. Množství dat je generováno především multimédii ve vysoké kvalitě a obsahem generovaným stále rostoucím počtem uživatelů. Množství zpracovávaných a ukládaných dat průběžně narůstá, proto je problém uchovávat data na jednom serveru, který lze omezeně škálovat(přidáním zdrojů). Data je proto vhodné ukládat do distribuovaných souborových systémů, které jsou snadno škálovatelné, jak v oblasti výpočetního výkonu, tak v oblasti úložného prostoru. Spolu s množstvím dat roste i počet klientů, kteří k těmto datům přistupují, největší nárůst je v oblasti mobilních zařízení. Mobilní zařízení mají ovšem problémy s připojením, z praxe je známa nestabilní kvalita přenosu, časté výpadky, vysoká latence, nízká přenosová rychlost a datové limity. V dalších částech článku bude řešena problematika přístupu k datům s ohledem na kvalitu přenosových linek.

2 Distribuované souborové systémy

Distribuované souborové systémy (DFS) jsou navržené pro ukládání dat. Nabízí vzdálený přístup k souborům, které mohou být fyzicky rozložené na více serverech, tuto skutečnost překrývají a data nabízí transparentně jako jeden zdroj - adresářový strom. V rámci uzlů distribuovaného systému mohou probíhat replikace dat, migrace a zálohování. Distribuovaný souborový systém se obyčejně skládá z úložiště samotných dat, databáze metadat s jejichž pomocí řídí práci s daty (nalezení, čtení, změnu, replikaci a migraci dat). Distribuovaný souborový systém dále implementuje techniky autentizace a autorizace uživatelů, např. LDAP a Kerberos [5].

Základní vlastnosti DFS Definice distribuovaných souborových systémů specifikuje některé vlastnosti, především transparentnost systému vůči okolí (pro uživatele není rozdíl mezi lokálními a vzdálenými daty, neznají skutečné umístění). Následující výčet obsahuje běžně požadované vlastnosti DFS: transparentnost (klienti přistupují k datům skrze DFS jako k jednomu celku), škálovatelnost (navýšení zdrojů pro obsluhu většího množství uživatelů a dat), heterogenita (spolupráce na různých SW i HW platformách), bezpečnost (přístup k datům, zabezpečení před ztrátou a poškozením), replikace (data jsou dostupná ve více kopiích), migrace (data je možné v rámci systému přesouvat).

Pokročilé vlastnosti DFS Další žádané vlastnosti, které nejsou u současných implementací DFS běžné nebo dostupné jsou například: online replikace dat (data jsou bez prodlevy replikována), multimaster read/write replikace metadat (možnost zapisovat na jakoukoli dostupnou repliku).

3 Synchronizace replik a DFS

Jak bylo již zmíněno v předchozím oddíle, většina DFS je složena ze dvou základních částí - úložištěm metadat (umístění fyzických dat, seznam a stav replik, přístupová oprávnění, informace o souborech...) a fyzickým úložištěm dat. Úložiště metadat bývá pro celý strom DFS společné nebo rozdělené podle logických svazků v rámci DFS, přístup k jednotlivým částem je však transparentní, přes jediného prostředníka. Tato část je kritická pro další funkci souborového systému (přístup k obsahu souborů, řízení replikace fyzických dat atp.) a je třeba zajistit její replikaci, konzistenci a synchronní stav. Replikaci je možné provádět pomocí následujících modelů a odpovídajících algoritmů:

- **Transakční replikace** (transactional replication) - změna je synchronně propagována na všechny servery jako jedna distribuovaná transakce
- **Replikace pomocí shody** (state machine replication) - změna je propagována na servery na základě shody majority (Paxos algoritmy [1], [2]), majorita má vzájemně data v konzistentním stavu
- **Virtuální synchronnost** (virtual synchrony)- změna je propagována asynchronně pomocí uspořádané fronty zpráv, přístup je vždy možný jen na aktuální repliky

Každý z modelů má své spefické vlastnosti v oblasti komunikační režie, konzistence dat, spolehlivosti a latence. Transakční zpracování nabízí silný model konzistence (lze označit i za striktní), nicméně selhání jednoho z uzlů, vysoká latence nebo neznámý stav může zapříčinit zastavení systému. Aktulálně se využívají algoritmy 2PC[3] a 3PC[4], které jsou relativně jednoduché na reálnou implementaci. Replikace s využitím shody vyžaduje pouze nadpoloviční počet serverů, které se na dané operaci shodnou, vyžaduje ovšem trvalé úložiště pro logy (dopředné, zpětné) a může vést k nekonzistentním stavům jednotlivých replik (menšina serverů, která se nepodílela na shodě musí provést obnovu - provedení chybějících operací). Hlavní výhodou je, že Paxos algoritmy uvažují asynchronní komunikační linky s variabilním zpožděním a možnost selhání libovolného uzlu. Implementace těchto algoritmů je však v reálném světě poměrně náročná. Virtuální synchronnost představuje model, který představuje asynchronní zpracovávání seřazených zpráv - dva servery vidí zprávy přijaté ve stejném pořadí, ale neprovádní změny synchronně. Tento model nabízí vyšší výkon, ale není příliš odolný vůči chybám.

Propagaci replik lze klasifikovat do dvou skupin také podle způsobu propagace změn: **synchronní** - konzistence replik je důležitější než výsledný výkon, data nejsou potvrzena dokud nedojde k synchronnímu zápisu na zvolených replikách, dochází tak k akumulaci možné latence, **asynchronní** - okamžitý výkon je důležitější než konzistence, data jsou nejprve zapsána na primární repliku a následně je zaslána zpráva o změne dotčeným replikám.

Dále lze replikaci dělit podle rozdělení rolí jednotlivých replik. Jednou z nich je, že replikace dat je řízena jedním hlavní/centrálním uzlem (master, leader), který přijímá požadavky na změny a ty propaguje

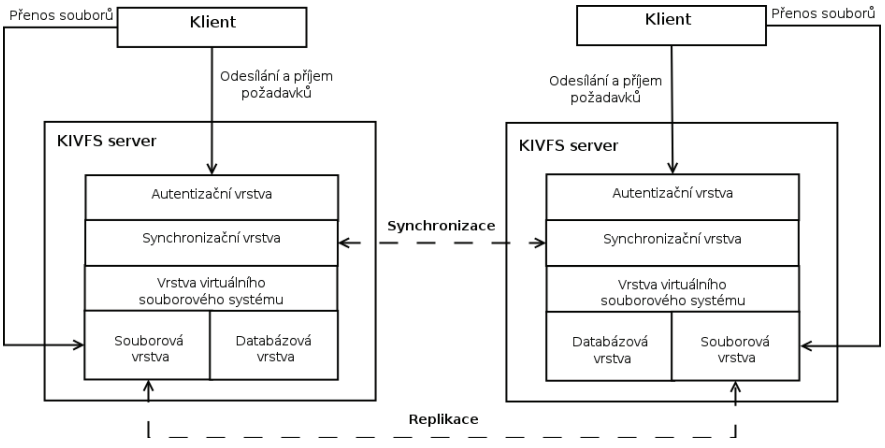
do podřízených replik. Tento způsob je použitelný u všech předchozích modelů. Zde není nutné řešit algoritmy absolutní uspořádání, protože to je určeno hlavním uzlem (řeší vzájmné uspořádání paralelních požadavků), tím je zajištěná vyšší rychlost zápisu, protože odpadá část komunikační režie. Nevýhodou tohoto přístupu (master-slave) je potřeba řešení situace, kdy dojde k selhání hlavního serveru. Tento stav je třeba nejprve detekovat (např. časový limit nečinnosti) a následně zvolit nový hlavní server. Během této doby je činnost systému pozastavena.

Dalším možným přístupem je rovnost všech replik (MultiMaster replikace), kdy každá replika může iniciovat změny. Aby mohla být změna povedena je třeba nejprve vyjednat její absolutní pořadí a následně schválit její zápis. Toto řešení nevyžaduje vyhledání hlavního uzlu, detekci jeho selhání a volbu nového, není třeba řešit situace kdy je hlavní uzel slabým článkem systému. Na druhé straně vyžaduje větší množství komunikace mezi uzly, než předešlé přístupy.

4 KIVFS

Na Katedře informatiky a výpočetní techniky Fakulty aplikovaných věd Západočeské univerzity (KIV) byl navržen a vytvořen experimentální distribuovaný souborový systém KIVFS[6][7] za účelem implementace a zkoumání pokročilých vlastností distribovaných souborových systémů (multimaster replikace metadat, online replikace dat, směrování požadavků).

KIVFS je implementováno jako skupina služeb (vrstev), které poskytují specifické služby souborového systému. K implementaci pomocí samostatných služeb bylo přistoupeno z důvodu snadnější implementace, kdy každá služba je při svém vývoji nezávislá na ostatních, což dává možnost vyměnit implementace jednotlivých služeb a porovnat jejich funkce, kvalitativní a výkonnostní parametry.



Obrázek 1: Schéma KIVFS

Jednotlivé servery mezi sebou komunikují vlastním protokolem nad TCP nebo lokálně přes UNIX sokety. Struktura KIVFS je naznačena na obr. 1, kde jsou zobrazené jednotlivé vrstvy a komunikace. Implementace je kompletně v uživatelském prostoru, jednotlivé služby jsou tak závislé pouze na použitých knihovnách. Pro autentizaci používá Kerberos[5], přenost je šifrován pomocí SSL a data jsou ukládána v relačních databázích.

4.1 Klíčové vlastnosti KIVFS

V současné době jsou implementovány následující funkčnosti: **online replikace** - data jsou ihned po dokončení jejich nahrání replikována, **multimaster read/write repliky** - na každou repliku je možné zapisovat, čtení probíhá z libovolné repliky s aktuální verzí dat[7], **statistiky klientských přístupů** -

KIVFS poskytuje statistiky přístupů, ty pak slouží pro pokročilé strategie klientského cacheování [8], **směrování dat** - data jsou směrována mezi servery za účelem nejrychlejšího doručení[9].

KIVFS využívá distribuované transakce (s potvrzováním majoritou) a logické časové značky, aby bylo zajištěno, že stav metadat bude na všech uzlech ve shodném stavu. Požadovaný model konzistence metadat je striktně konzistentní. Je třeba zaručit, že operace budou provedeny na všech aktivních uzlech ve stejný čas. Tyto požadavky vznikly jako základní v předchozích pracích[7],[6] vzhledem k požadovným vlastnostem pro KIVFS.

5 Řešená problematika

Při výkonnostních testech KIVFS byl identifikován problém v oblasti nutné komunikace mezi servery pro synchronizaci požadavků a koordinaci distribuovaných transakcí, kdy byla rychlost synchronizace a její režie limitující pro dosažení lepších výsledků v porovnání s OpenAFS.

Pro každou synchronní operaci v KIVFS je zapotřebí nejprve přiřadit požadavku logickou časovou značku, následně ho podle ní zařadit do fronty pro synchronní provedení dané operace jako atomické transakce (vzhledem ke klientské aplikaci). Časovou značku aktuálně určují skalární logické hodiny. Získání časové značky probíhá ve třech krocích, v prvním pošle server, který požadavek obsluhuje, zprávu ostatním serverům s žádostí o přidělení časové značky a její návrh, v druhém kroku ostatní servery pošlou zpět vlastní návrh. V posledním kroce obsluhující server vybere většinovou časovou značku a oznámí ji ostatním, ti si požadavek zařadí do svých lokálních front. Pro každý požadavek je tedy nutné odeslat/přijmout $3N$ zpráv (N je počet serverů).

Následně je požadavek synchronně proveden na všech serverech, to je implementováno jako distribuované třífázové provedení[4]. V případě třífázového provádění koordinátor transakce nejprve osloví ostatní servery s požadavkem na provedení a čeká na jejich odpovědi, v případě kladných odpovědí pošle pokyn k přípravě provedení a opět čeká na odpovědi, jako poslední krok vyšle pokyn ke skutečnému provedení. To představuje příjem/odeslaní $6N$ zpráv ve 3 krocích.

Každý příchozí požadavek tedy vyžaduje celkem $9N$ odeslaných/přijatých zpráv posílaných v 5 krocích. V každém cyklu se vyčkává na všechny odpovědi. Nejpomalejší nebo nejméně spolehlivá linka pak určuje výslednou nejvyšší rychlost synchronizace a distribuovaného provádění.

Řešením ztráty výkonu v případě, že jeden ze serverů bude na výrazně horší lince než ostatní, je tento server ze skupiny vyřadit a ponechat ho pouze v režimu permanentní obnovy dat, kdy není součástí hlasování. Touto akcí je možné, že vyřadíme server, který by mohl využít jinou alternativní linku. Dále klienty v blízkosti vyřazeného serveru donutíme se připojit k vzdálenejším serverům také přes toto nekvalitní spojení, což povede k nižší efektivitě přenosu dat a objektivnímu zpomalení DFS z klientů. Je třeba navrhnout lepší řešení, které by tento problém řešilo s minimálními dopady na klienty a DFS.

6 Optimalizace synchronizační režie

V následující části článku budou navrženy možné optimalizace v přenosu synchronizačních požadavků za účelem minimalizace zpoždění způsobeného nutnou komunikací mezi servery při získávání shody nad prováděnými operacemi.

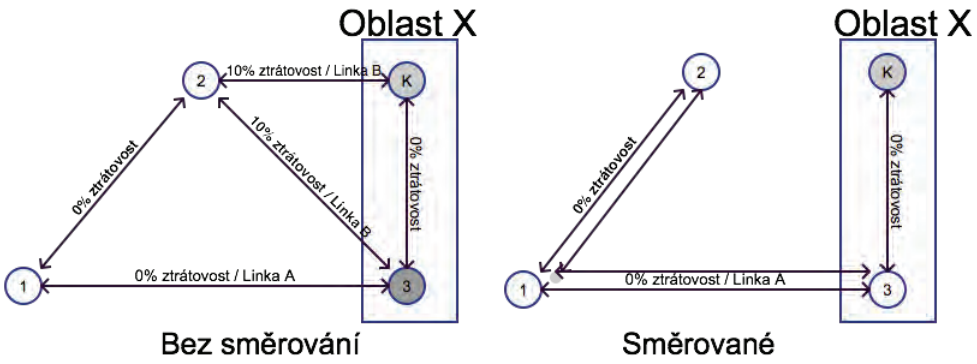
6.1 Směrování synchronizačních požadavků

Vzhledem k tomu, že v rámci KIVFS jsou směrovány datové přenosy[9], se nabízí možnost směrovat i synchronizační přenosy. Princip směrování synchronizačních požadavků je naznačen na diagramu obr.2.

Předpokádaná situace je následující, v oblasti X je přítomen jeden KIVFS server, dvě různé síťové linky A, B a klienti. Linka B má ztrátovost paketů 10%, tato linka je však použita pro komunikaci

mezi servery 2 a 3 (předpokládáme, že není možnost ovlivnit směrovací tabulky), linka A má ztrátovost packetů 0% a je využívána pro spojení mezi servery 1 a 3 (privátní spoj). Standardní chování by bylo, že rychlost/odezva linky B bude limitní pro rychlost synchronizace nebo že bude server 3 vyřazen. První možnost - zpomalení synchronizace má negativní dopad na rychlost celého systému. Druhá možnost může vést k tomu, že klienti budou využívat jiný server než 3, v nejhorším případě přes linku B, což povede k znekvalitnění služby pro klienty a ztrátu výhody v tom, že je server 3 umístěn v lokální síti.

Optimálním řešením je linku B nahradit směrováním požadavků přes server 1. Výsledkem bude, že linka B nebude mít vliv na rychlost synchronizace, veškerá komunikace bude probíhat po spolehlivých linkách. Režie spojená s předáváním požadavků přes server 1 je zanedbatelná. Implementace a ověření je součástí dalších prací.



Obrázek 2: KIVFS směrování

6.2 Redukce komunikujících serverů

Další možností je redukovat počet serverů, které se na synchronizaci podílí, to představuje rozdělení KIVFS serverů do skupin, například podle podstromů v rámci adresářové struktury, zde bude komplikované řešit například přesun souborů mezi těmito stromy, nelze ani zaručit pevnost těchto skupin, vzhledem k tomu, že data mohou být dynamicky přesouvána mezi servery.

6.3 Sdružování požadavků

Dalším přístupem je provádět synchronizaci a provádění požadavků po více operacích zároveň - logicky je shlukovat a místo synchronizace a provádění každého zvlášť je provádět hromadně. Zde je třeba řešit jak skupiny požadavků volit, jak určovat pořadí požadavků v rámci skupiny a jak dlouho čekat na sestavení skupiny, aby tím nebyl ovlivněn výkon (latence).

6.4 Redukce počtu zpráv

Mezi servery je předpokládána komunikace každý s každým. Zde je možné využít grafových algoritmů a analogii s distribuovaným multicastem. V nejjednodušší variantě lze vycházet ze záplavového algoritmu (zpráva bude zaslána ”nejbližším”serverům a ty zprávu přepošlou dál). Dalším krokem je vyhledávání super-serverů, které mají kvalitní linky mezi sebou (WAN) a zároveň na své sousedy (LAN). Super-servery budou prostředníkem v komunikaci, zprávy mezi nimi se pak budou předávat pouze jednou a jejich distribuce koncovým uzlům bude prováděna cílové síti.

7 Aktuální stav

Během mé práce byla dle zadání[7] vytvořena synchronizační vrstva KIVFS, která slouží k přidělování logických časových značek a transakčnímu zpracování požadavků. Synchronizační vrstva dále sleduje stavy jednotlivých linek, které jsou předávány mezi servery a následně jsou z nich přepočítávány optimální trasy pro směrování provozu datových požadavků. Vrstva také zajišťuje obnovu uzlu po výpadku. Současné práce se věnují optimalizaci síťové komunikace (úpravy protokolu KIVFS a ladění výkonu nad TCP) a implementaci funkcí uvedených v předchozích kapitolách. Předmětem disertační práce bude návrh optimalizačních rozšíření algoritmů pro zvýšení propustnosti systémů využívající absolutního řazení a distribuovaných transakcí, zvolenou metrikou je celková propustnost systému a latence při zpracování klientských požadavků.

8 Závěr

V článku byl uveden experimentální distribuovaný souborový systém KIVFS, na kterém byla představena problematika synchronizace a provádění požadavků v distribuovaném systému. Byly navrženy možnosti optimalizace komunikace nutné pro synchronizaci a provádění, které jsou postupně implementovány. Aktuálně na KIVFS probíhá implementace směrování synchronizačních požadavků z měřených metrik (latence, rychlost), výsledky ověření budou publikovány v dalších pracích a porovnány se souborovým systémem OpenAFS. Následně budou implementovány a v praxi ověřeny další představené postupy.

Reference

[1] LAMPORT, L. *Paxos made simple* ACM SIGACT News 32, 4 (Dec. 2001), 18–25.

[2] CHANDRA, T. D., GRIESEMER, R., REDSTONE, J. *Paxos Made Live: An Engineering Perspective* Proceedings of the Twenty-sixth Annual ACM Symposium on Principles of Distributed Computing, PODC '07, 2007, ISBN 978-1-59593-616-5

[3] BERNSTEIN, P. A., HADZILACOS, V., GOODMAN, N. *Concurrency Control and Recovery in Database Systems* Kapitola 7, Addison Wesley Publishing Company, 1987, 370 s. ISBN 0-201-10715-5

[4] SKEEN, D., STONEBRAKER, M. *A Formal Model of Crash Recovery in a Distributed System* IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, svazek SE-9, číslo 3, květen 1983

[5] NEUMAN, C., *The Kerberos Network Authentication Service*, RFC 4120, Network Working Group, červenec 2005 Dostupné z URL: http://www.ietf.org/rfc/rfc4120.txt

[6] JUNÁK, M., MATĚJKA, L., PEŠIČKA, L., PIVNIČKA, M., SKUPA, J., STREJC, R., STEINER, V., ŠAFAŘÍK, J. *KIV-DFS-Experimental Distributed File System* In Informatics 2009. Košice: Technical University, 2009. s. 45-50. ISBN: 978-80-8086-126-1

[7] MATĚJKA, L., ŠAFAŘÍK, J., PEŠIČKA, L. *Distributed file system with online multi-master replicas.* In: Engineering of Computer Based Systems (ECBS-EERC), 2011 2nd Eastern European Regional Conference on the. IEEE, 2011. p. 13-18.

[8] BŽOCH, P., et al. *Design and Implementation of a Caching Algorithm Applicable to Mobile Clients*. Informatica, 2012, 36.4: 369-378.

[9] SKUPA, J. *KIVFS - Synchronization and requests routing.* Plzeň, 2012. Diplomová práce. Západočeská univerzita v Plzni. Fakulta aplikovaných věd. Katedra informatiky a výpočetní techniky.

CONTRIBUTION TO THE LOW-POWER DESIGN

Dominik Macko
Applied Informatics, 3-rd class, full-time study
Supervisor: Pavel Čičák, Consultant: Katarína Jelemenská

Faculty of Informatics and Information Technologies
Slovak University of Technology
Ilkovičova 2, 84216 Bratislava

dominik.macko@stuba.sk

Abstract. Power consumption is one of the key constraints in system on chip (SoC) design process. Very powerful and widely accepted method for applying power-reduction techniques is adoption of power-management strategy. Power management is commonly specified at the register-transfer level (RTL) or lower level. We have proposed an extension of low-power design flow to the system level integrating power management into the system-level specification. Based on this specification the standard-based specification along with the power-management control logic will then be automatically generated during the high-level synthesis process. The generated power management along with the functional model can be verified and analyzed at more-mature RTL. This paper briefly describes this extension and summarizes the experimental results for evaluation of several aspects.

Keywords. Low power, Power consumption, Power management, Power reduction.

1 Introduction

The ever increasing demand for portable systems-on-chips (SoCs) has added the power consumption to the traditional constraints (e.g. area, performance, or cost) [1]. To address the increasing power-reduction requirements, there have been many techniques developed, such as clock and power gating, or voltage and frequency scaling (summarized in [2]). Some of these techniques are straightforward, others are difficult to adopt. To help the design teams to use the advanced power-reduction techniques, the standard for design and verification of low-power integrated circuit (IEEE Std 1801-2013 [3]; known as UPF) was developed. The UPF-based low-power design flow involves the application of the power-reduction techniques in an RTL (Register-Transfer Level) or lower level digital-system model. These techniques impact all aspects of integrated-circuits development (i.e. design, implementation, and verification) increasing thus the ever-growing complexity of current digital-systems designs even more. To increase the system development efficiency, the system design should start at more abstract - so called electronic system level (ESL) [4].

The combination of existing approaches can close the gap in the state-of-the-art of the low-power design process. The standard-based abstract system-level power management can be inspired by [5]. If the concepts of power management are the same at both ESL and RTL, the equivalence is verified much easier. Fully standardized (UPF) power management at the RTL assures the compatibility with the existing EDA (Electronic Design Automation) tools. The use of high-level synthesis in the power architecture analysis process (similarly to [6]) provides better trade-off (performance, power, and area)

and more accurate analysis. Such an approach exploiting more automation can reduce the possibility of human errors. The RTL implementation stage is achieved more quickly, and therefore more mature RTL verification process can start earlier. The abstract power management is easier to understand, thus even the designers not familiar with the power-management techniques can design for low power.

The achievements in the finite-state machines designs [7, 8] can be applied to the power-state machines (PSMs) of the power-management units (PMUs). The PSM architecture needs to be modified in order to manage its own power consumption. Such a modification can save most of its leakage power and significant portion of dynamic power.

We have proposed an extension of UPF-based low-power design flow in which the power-management specification is integrated into the system-level functional specification model [9]. The abstract specification itself would have no benefit without the means to analyze the impact on power consumption of selected power architecture. Therefore, we have augmented the proposed specification by the synthesis algorithm to more-detailed standard UPF power management [2, 10], that is supported by current EDA tools for power analysis. UPF standard assumes that the control signals for the power-management elements (e.g. power switches, level shifters, or isolation cells) are generated by the functional model. Thus, we need to generate the functional RTL description of PMU besides the UPF power management itself. To be consistent the PMU for low-power design should be power-efficient as well. Therefore, we have proposed the novel PMU architecture utilizing the power-management techniques inside its own PSMs [11] and conducted experiments to determine its power-efficiency [12]. This paper briefly describes our methodology and reports on the experimental evaluation of its benefits.

1.1 Dissertation goals

The research is targeted to functional verification of system-level digital systems design with an aspect of power consumption, specifically to the following goals.

- Identification of power-reduction techniques applicable at the system level of abstraction, their evaluation and selection of potential techniques for integration.
- Development of a method for integration of selected power-reduction techniques into the system-level model specification.
- Development of power-aware hybrid verification method based on modified equivalence checking and property checking.

2 Proposed low-power design flow

The key idea of the proposed novel methodology [2] lies in an extension of the current low-power design flow in a way that will enable to utilize the advantages of system-level modeling (e.g. shorter specifications, subsystems intercommunications, or faster and simpler verification). The proposed extension, illustrated in Figure 1, keeps the current low-power design flow steps intact enabling thus to use the traditional design/verification methods and tools at the lower levels.

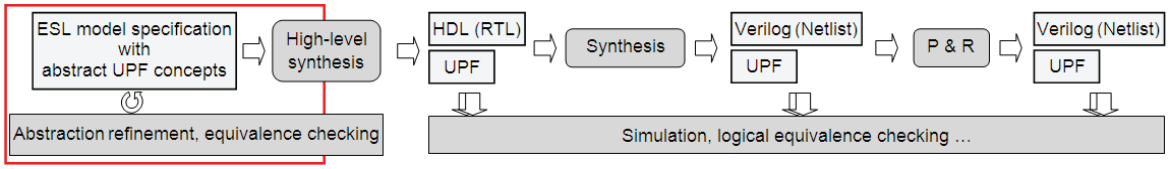


Figure 1: Low-power design flow.

This methodology starts from crude system specification at the ESL, where the main UPF concepts are integrated into the system functional specification in an abstract form. The specification participates in the abstraction refinement process. When sufficiently refined the high-level synthesis enables to

automatically extract the specified power management and to generate its standard UPF representation along with the RTL functional model (typically in HDL). Then, the low-power design flow continues in a traditional way.

After power management is specified, this specification has to be verified for functional, electrical, and structural correctness and completeness (syntax, semantics, design object, inconsistent or incomplete power intent) – this is usually done through formal verification. The next step is to verify the correct functionality of the system with low-power behavior on top of normal functional behavior [13]. As mentioned in [14], additional low-power design units are often a rich source of errors and must be thoroughly verified for all specified operating modes. One of the advantages of the proposed methodology lies in the automation. Since power-management specification at RTL is automatically generated, we are able to avoid many power-related human errors. The designer does not need to worry about specifying low-level power-management elements, such as power switches or isolation cells. Moreover, the assertion generation concerning the power-management control sequences is automated in a way similar to [15]. In this way, the low-level power-related logic can be verified (both by simulation and formally) based on system-level abstract specification speeding thus up the verification process.

2.1 Power-efficient power-management logic

The power-management logic determines the system power mode and generates the signals controlling the power-management elements. During the power-mode transition, the exact sequence of control signals has to be driven. These control sequences are handled by so called power-state machines (PSM) – application-specific kind of finite-state machine. In PSM, the inputs are the subset of the outputs. It means that the change of the inputs values triggers transitions through several states, each producing different outputs values. This application-specific nature of PSM gives us an opportunity to save the power inside the PSM when the state transition is not needed. When the current power state at the PSM outputs and the target power state at the PSM inputs are the same, the sleep signal is activated, powering the transition logic down. The comparison logic simultaneously generates the clock-gating enable signal for the state logic – it guarantees that the state is not changing when the transition logic is powered-down. The proposed PSM modified architecture is shown in Figure 2.

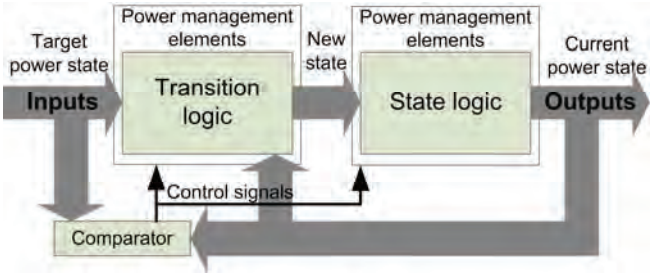


Figure 2: The proposed PSM architecture.

3 Experimental results

For evaluation of suitability of proposed abstract power-management specification we use the complexity comparison between the system-level and standard-based RTL power intent. As a complexity parameter we take the number of characters the designer needs to use for description of the power intent. Note that the system-level power management was not created with intent to use the shortest description, but to abstract from lower-level details, such as power nets and power-management elements (e.g. power switches, level shifters, or isolation cells).

We have generated half a million samples for system-level power management, describing the power intent in HSSL- and SystemC-integrated power-management extensions and synthesized the standard UPF power management specification using two proposed synthesis algorithms. The first one [2] generates the equivalent power intent in UPF, while the second one handles in addition some inconsistencies in specification (e.g. a power state is specified for power domain, but never used) and analyzes the architecture of the system that is driving its decisions (e.g. component connections).

For the generated samples we were scaling the number of power domains from 1 to 10, the number of power modes from 1 to 10, the average number of instances per power domain from 1 to 10, the number of instances interconnections from 1 to 100, and the average number of power states in power domains from 1 to 5.

In Figure 3, the HSSL-integrated power-management specification (*HSSL*) is compared to the generated standard-based power-management specification. *HSSL->UPF1* and *HSSL->UPF2* denote the usage of the first and second algorithm respectively. In the left-hand chart the absolute values of the complexity parameter are shown. In the right-hand chart the relative comparison is given.

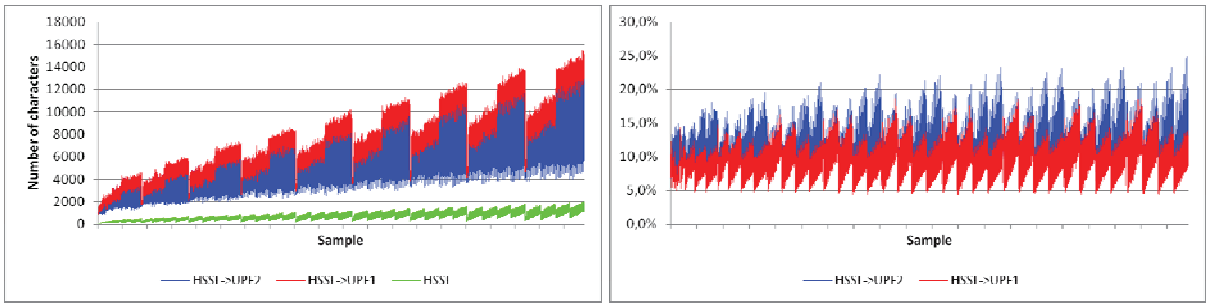


Figure 3: HSSL-UPF power-management specification comparison (absolute – left part; relative – right part).

The result is that such system-level power-management has approximately 10.3 times lower complexity in average compared to the standard UPF specification for the first synthesis algorithm and 8.7 times lower for the second algorithm. The higher benefit (higher difference between specifications) was achieved for samples with a smaller number of instances per power domain and a high number of inter-domain connections.

When evaluating the SystemC-integrated power-management specification, the results are slightly different. For the first synthesis algorithm, in average the 4.4 times lower complexity of system-level power-management specification compared to the standard UPF specification was achieved and 3.7 times lower for the second algorithm. The reason for HSSL to be more efficient in this evaluation is that we integrated the power management into the language itself. For SystemC language, we have used an extension library and traditional C++ modeling style. For example, the assignment of an instance to the power domain is in HSSL incorporated into the instantiation of the component statement, while in SystemC the additional method has to be called for a power domain object with a component instance name as an argument.

In the second group of experiments we have compared our proposed modification of PSM architecture with the original one. Firstly, we have processed the experiments on a single PSM design controlling one power domain. It switches between 5 power states represented by 3 control signals. The PSM input target state can be one of the four power states – one power state is transparent to the power-mode determination module, it is just an intermediate state in order to correctly reach some other state. The second comparison was processed on a slightly bigger PSM design that controls 4 power domains switching between 8 combinations of power states of these power domains (10 control signals). As an input, only 5 combinations are valid – the remaining 3 combinations are internal.

Since the power has also the dynamic contributing element, the experiments consist of four simulation test-cases, each with different set of parameters. These parameters are the clock frequency

and the toggle-rate of the target state (PSM input). The simulation test-cases are summarized in the Table 1. The first column refers to the test-case number and *f(CLK)* stands for the clock frequency. The *simulation time* reports the actual runtime of the simulation test-case. TR_{TS} represents the toggle-rate of the target-state signal and is expressed by the number of target-state changes (toggles) per clock period (clock cycle). The last test-case reflects the situation, when the toggle-rate per clock-period is very low. Moreover, the clock frequency was lowered to 50 kHz in order to simulate more realistic situation (the real PMUs operate commonly at the real-time clocks – typically 32.768 kHz).

#	f(CLK)	Simulation time	TR_{TS} [toggles/T_{CLK}]		Power [nW]					
			PSM1	PSM2	PSM1b	PSM1sm	Saving	PSM2b	PSM2sm	Saving
1	50 MHz	5 µs	0.288	0.352	2676.539	2601.271	3%	4599.603	4176.747	9%
2	50 MHz	5 µs	0.116	0.112	1934.087	1367.942	29%	3074.951	1866.738	39%
3	50 MHz	5 µs	0.04	0.036	1787	1032.114	42%	2629.805	1129.035	57%
4	50 kHz	10 s	0.00001	0.000012	799.697	567.994	29%	1368.202	335.154	76%

Table 1: Simulation test-cases description and power estimations.

In the right part of Table 1, the results of power estimations for described designs are shown and compared. *PSM1b* refers to the first PSM design (controlling one power domain) with the basic architecture (without internal power management). *PSM1sm* refers to the self-managed architecture (with integrated comparator and power-management elements) of the first PSM design. Analogously, *PSM2b* and *PSM2sm* refer to the second PSM design (controlling four power domains). The columns named *Saving* refer to the amount of saved power.

We may notice that the power saving of the modified PSMs scales from 3 to 76 %. The power saving increases with decreasing TR_{TS} parameter and starts to decrease with very rare target-state changes (test-case 4) for the first PSM design (small PSM), when the added components start to consume significant portion of power compared to the rest of PSM (even then there is a power saving of 29 %). The power-reduction in the first test-case is very low and for such case self-management would not be worthy of additional area requirements. This test-case can be considered boundary case, for which the modified PSM architecture is beneficial regarding the power consumption. In Table 1, the TR_{TS} for the first test-case is shown to be approximately 0.3. It means that the target state changes approximately each third clock cycle. The transition logic is activated for one or more clock cycles, depending on the sequence needed to correctly reach the target state from the current power state. Therefore we may assume, that in this case the PSM was active (transition logic powered-up) more than 30 % of the simulation time. It means that the PSM has to be at least 70 % of the time inactive in order the modified PSM architecture is suitable to be used.

4 Conclusion

Our work has produced several contributions. We have extended the low-power design methodology to the system level, abstracting from lower-level details, such as power nets, ports and other power-management elements. Our methodology utilizes high-level synthesis process in order to find trade-off between performance, power, and area of the system. The experiments showed that the abstract power-management specification is approximately 7-times more concise than its RTL standard-based equivalent. Therefore, it is easier to adopt and it is less prone to human errors, reducing the design re-spins. The control signals for power-management elements are generated by PMU. In order the PMU to be power-efficient, we have proposed a modification of its PSMs. The modified architecture, managing its power consumption, reduces the power inside the PSM up to 76 % in the experimental design. Although, the power consumption of PMU is negligible compared to the whole system, in systems where the power consumption is converging towards the sleep-mode power, the PMU becomes the significant consumer of the power.

Besides the mentioned experimentally-proven contributions, we have produced several others. We have determined the existing power-reduction techniques usable at the system-level of abstraction. We have integrated the power-management specification into the HSSL language in a form of language syntax extension and into the SystemC language in a form of extension library. We have proposed the technique of stopping the system-block operation by utilizing the isolation power-management elements. Finally, we have proposed two power-management specification synthesis algorithms for transformations of the system-level abstract power intent into the more-detailed UPF format. The first one produces the equivalent power-management specification, the second one uses optimizing decisions based on the system architecture and power-management consistency checks.

Our further work is focused on automated generation of power-related assertions enhancing the verification process.

Acknowledgment

This work was partially supported by the Slovak Science Grant Agency (VEGA 1/1008/12 and VEGA 1/0616/14), Slovak University of Technology ("ANSNS – Low-power system design automation") and COST Action IC 1103 MEDIAN.

References

[1] S. Bailey, G. Chidolue, and A. Crone, "Low Power Design and Verification Techniques," 2007.

[2] D. Macko, "System-level power management specification," in PAD, 2013, pp. 87-92.

[3] IEEE Standard for Design and Verification of Low Power Integrated Circuits, IEEE Std 1801-2013.

[4] ITRS, Design, 2011, http://www.itrs.net/Links/2012ITRS/Home2012.htm.

[5] O. Mbarek, A. Pegatoquet and M. Auguin, "A Methodology for Power-Aware Transaction-Level Models of Systems-on-Chip Using UPF Standard Concepts," Integrated Circuit and System Design: Power and Timing Modeling, Optimization, and Simulation, LNCS, vol. 6951, pp. 226-236, 2011.

[6] S. Ahuja, High Level Power Estimation and Reduction Techniques for Power Aware Hardware Design, Faculty of the Virginia Polytechnic Institute and State University, 2010. Dissertation thesis.

[7] K. Usami and H. Yoshioka, "A Scheme to Reduce Active Leakage Power by Detecting State Transitions," in MWSCAS, IEEE, 2004, pp. I-493-I-496.

[8] K. Usami and N. Ohkubo, "A Design Approach for Fine-grained Run-Time Power Gating using Locally Extracted Sleep Signals," in ICCD, IEEE, 2006, pp. 155-161.

[9] D. Macko and K. Jelemenská, "Power-Intent Integration into the Digital System Specification Model," in MEDIAN, 2013, pp. 49-52.

[10]D. Macko and K. Jelemenská, "Managing digital-system power at the system level," in AFRICON, IEEE, 2013, pp. 179-183.

[11]D. Macko and K. Jelemenská, "Self-Managing Power Management Unit," in DDECS, IEEE, 2014, pp. 159-162.

[12]D. Macko and K. Jelemenská, "Power-Efficient Power-Management Logic," submitted to PATMOS 2014.

[13]Cadence Design Systems, A practical guide to low power design: User experience with CPF, 2012.

[14]N. Khan, "Cosed-loop verification methodology for low-power SoC design," Special Technology Report - Low Power Design, no. 1, pp. 7-8, September 2008.

[15]A. Hazra, S. Goyal, P. Dasgupta and A. Pal, "Formal Verification of Architectural Power Intent," IEEE Transaction on very large scale integration (VLSI) systems, vol. 21, no. 1, pp. 78-91, January 2013.

ANALÝZA DYNAMICKÝCH VLASTNOSTÍ SMĚROVACÍCH TABULEK

Jiří Matoušek
Výpočetní technika a informatika, 3. ročník, prezenční studium
Školitel: Jan Kořenek

Fakulta informačních technologií, Vysoké učení technické v Brně
Božetěchova 1/2, 612 66 Brno

imatousek@fit.vutbr.cz

Abstrakt. Současné požadavky kladené na směrování paketů v páteřních sítích vynucují akceleraci této operace v hardware s použitím paměťově efektivních technik reprezentace odpovídající sady prefixů. Ačkoliv je dynamická povaha směrovacích informací významným faktorem ovlivňujícím paměťovou reprezentaci sady prefixů, doposud nebyly dynamické vlastnosti směrování v páteřních sítích studovány z pohledu změn ve směrovací tabulce. Tento příspěvek obsahuje chybějící analýzu, která byla provedena jako první krok k návrhu a implementaci systému dynamického přidělování paměti na čipu FPGA pro potřeby reprezentace sady prefixů ze směrovací tabulky. V příspěvku je ukázáno, že směrovací tabulka obsahuje nezanedbatelné množství dlouhodobě stabilních záznamů. Změny ve směrovací tabulce pak připadají především na aktualizace záznamů, které jsou většinou provedeny za méně než 2 hodiny. V rámci analýzy bylo také ukázáno, že záznamy odstraňované ze směrovací tabulky byly v 70 % případů přidány před méně než 24 hodinami.

Klíčová slova. Směrovací tabulka, dynamické vlastnosti směrování, páteřní sítě, LPM.

1 Úvod

Neustále se zvyšující množství dat přenášených přes počítačové sítě má přímý vliv na nárůst podporovaných přenosových rychlostí. Například pro Ethernet je již standardizována přenosová rychlost 100 Gb/s [1]. Počet koncových zařízení připojených k Internetu navíc dosahuje řádu miliard a nadále rychle narůstá. Oba tyto trendy se přitom přímo dotýkají jedné ze základních operací v počítačových sítí – směrování paketů. Směrovací tabulky pro páteřní sítě dosahují velikosti přes 500 tisíc IPv4 a téměř 18 tisíc IPv6 záznamů (viz [2]), na základě kterých je při podpoře přenosové rychlosti 100 Gb/s nutné učinit rozhodnutí o směrování paketu za 6,72 ns. Směrování paketu již tedy není možné provádět v software, ale tato operace musí být v páteřních sítích implementována v hardware.

V rámci své disertační práce na téma *Využití rekonfigurovatelných obvodů v oblasti počítačových sítí* se zabývám použitím technologie FPGA pro implementaci operace vyhledání nejdelšího shodného prefixu (anglicky *longest prefix match*, *LPM*), která představuje výpočetně nejnáročnější součást procesu směrování paketů. Typická implementace operace LPM pro vysokorychlostní sítě je založena na stromové datové struktuře kódující prohledávanou množinu prefixů. Kvůli požadavku na rychlé zpracování je vyhledávání v této stromové struktuře nejčastěji implementováno v hardware ve formě zřetězené linky, jejíž jednotlivé stupně zajišťují vyhledávání na různých hladinách stromové struktury. Pro uložení od-

povídajících hladin stromové struktury je každému stupni zřetězené linky přiřazen samostatný paměťový blok.

Implementace s využitím zřetězené linky zajišťuje dostatečnou rychlost vyhledávání, která je však limitována rychlostí přístupu do paměti sloužící k uložení stromové struktury. V [3] bylo ukázáno, že při vhodném zakódování množiny prefixů je možné k jejímu uložení využít dostupnou distribuovanou paměť na čipu FPGA a dosáhnout tak dostatečné rychlosti vyhledání pro podporu přenosové rychlosti 170 Gb/s.

Vzhledem k dynamické povaze množiny prefixů se však paměťové nároky v jednotlivých stupních zřetězené linky v čase mění. S ohledem na omezené množství dostupné paměti na čipu FPGA se proto jeví jako vhodné zajistit dostatečné množství paměti v jednotlivých stupních zřetězené linky prostřednictvím dynamického přidělování paměti. Tento příspěvek se tudíž zabývá analýzou dynamických vlastností směrovacích tabulek v páteřních sítích, která by následně měla být rozšířena o pohled na vztah změn ve směrovacích tabulkách a paměťových nároků v jednotlivých stupních zřetězené linky. Na základě těchto analýz by pak měla být navržena architektura umožňující dynamické přidělování paměti na čipu FPGA.

Struktura příspěvku je následující. V kapitole 2 jsou představeny příbuzné práce a je identifikován prostor pro provedenou analýzu. Následně jsou v kapitole 3 popsána data, na nichž byla analýza prováděna, a způsob jejich předzpracování. Popis vlastní analýzy a jejích výsledků je obsahem kapitoly 4. Provedená analýza je také v rámci kapitoly 5 zasazena do kontextu dalších cílů disertační práce. Příspěvek a dosažené výsledky jsou shrnuty v kapitole 6.

2 Příbuzné práce

Existuje mnoho prací zabývajících se dynamickou povahou směrování paketů z pohledu koncových síťových zařízení. Komplexní analýzu aktualizací směrovacích informací zasílaných mezi páteřními směrovači však najdeme pouze v [4] a [5].

Původní analýza [4] provedená na datech z roku 1996 je založená na sledování posloupnosti aktualizací zasílaných protokolem BGP pro danou dvojici (prefix, směrovač). Identifikované posloupnosti přidání a odebrání prefixu jsou klasifikovány do tří kategorií: 1) aktualizace spojené se změnou směrování, 2) aktualizace spojené se změnou směrovací politiky a 3) patologické aktualizace směrovacích informací. Z výsledků analýzy vyplývá, že 99 % BGP aktualizací zasílaných mezi páteřními směrovači spadá do kategorie patologických aktualizací. Tato práce také přináší pohled na některé kvantitativní vlastnosti zasílaných aktualizací směrovacích informací: a) množství zasílaných aktualizací závisí na zátěži sítě a kopíruje její denní, týdenní a roční vzory, b) aktualizace pro danou dvojici (prefix, autonomní systém) jsou registrovány převážně s periodou 30 a 60 s a c) 35-100 % dvojic (prefix, autonomní systém) je aktualizováno alespoň jedenkrát za den, přičemž medián je 50 %.

Revize závěrů původní analýzy po 10 letech byla publikována v [5]. V rámci této práce byly uplatněny stejné metody analýzy aktualizací směrovacích informací jako v [4], avšak u některých posloupností přidání a odebrání prefixu došlo k jejich zpřesnění. Díky tomu bylo možné aktualizace směrovacích informací přesněji klasifikovat do výše uvedených 3 kategorií. Základním zjištěním revidované anlýzy je skutečnost, že z pohledu BGP aktualizací je Internet „zaměstnanější“ (legitimní aktualizace spojené se změnou směrování či směrovací politiky představují 84 % všech aktualizací) a „zdravější“ (podíl patologických aktualizací je jen 16 %). Ostatní zjištění revidované analýzy potvrzují výsledky z před desíti let a pouze u periody aktualizací pro danou dvojici (prefix, autonomní systém) se kromě významného podílu periody 30 s objevuje také značné množství aktualizací s periodou větší než 8 hodin.

Obě uvedené práce se zabývají především klasifikací aktualizací směrovacích informací a také některými jejich kvantitativními vlastnostmi. Z pohledu implementace operace LPM je však mnohem podstatnější vliv přijatých BGP aktualizací na samotnou směrovací tabulku, která se nemění s přijetím každé aktualizace. Navíc ne všechny změny směrovací tabulky znamenají změnu prefixu uloženého v tabulce,

a tudíž datové struktury reprezentující množinu prefixů pro potřeby operace LPM. Tento příspěvek se proto zabývá analýzou vlivu aktualizací směrovacích informací na směrovací tabulku, přičemž jejich vliv na reprezentaci prefixové sady pro potřeby operace LPM ponechává k rozpracování do budoucna.

3 Vstupní data a jejich předzpracování

Pro provedenou analýzu byla vybrána data z kolektoru RRC14 umístěného v Palo Alto (USA). Přístup k použitým datům je umožněn prostřednictvím služby RIS (*Routing Information Service*) raw data [6]. Použitá data pocházejí z období 1.10.2013 až 28.2.2014. V tomto období se počet záznamů ve směrovací tabulce zvýšil o 3,2 % z původních 493 219 až na 508 881 záznamů.

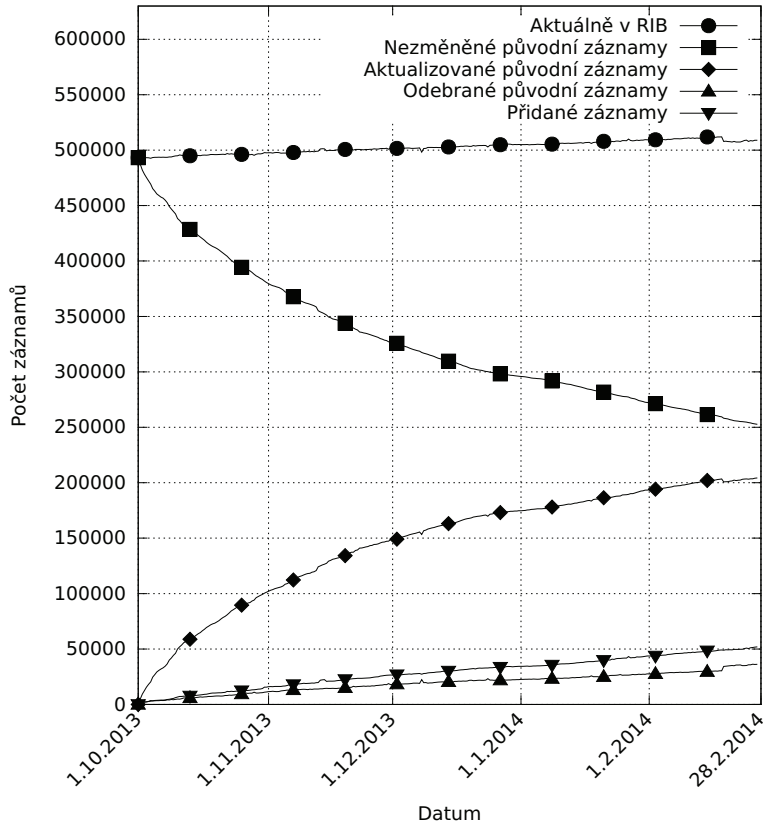
Na kolektoru RRC14 jsou data ze směrovacích tabulek k dispozici ve formě obrazů celých směrovacích tabulek s 8hodinovými rozestupy a také v podobě souhrnu všech BGP zpráv v intervalech po 5 minutách. Pro provedenou analýzu byla použita data v obou formátech. Celkový obraz směrovací tabulky sloužil jako výchozí stav, na který byly postupně aplikovány aktualizace (přidání či odebrání záznamu) obsažené v souhrnu BGP zpráv v jednotlivých 5minutových intervalech. Je však třeba poznamenat, že ne každé přidání či odebrání záznamu předepsané v BGP zprávě se projevilo jako přidání či odebrání záznamu z rekonstruované směrovací tabulky. Identifikátorem záznamu je totiž kromě prefixu IP adresy také identifikace zdroje této informace. Nový záznam byl tedy zaveden pouze v případě, že pro daný prefix nebyla přítomná směrovací informace ze žádného jiného zdroje. Odebrání záznamu pak nastávalo pouze v případě, že už pro daný prefix neexistovala směrovací informace ze žádného dalšího zdroje.

Protože záznamy v obrazu směrovací tabulky obsahují informaci o času přidání a BGP zprávy si s sebou také nesou informaci o času přijetí, je možné výše uvedeným postupem zrekonstruovat stav směrovací tabulky v libovolném okamžiku uvažovaného časového intervalu (s rozlišením na sekundy).

4 Analýza dynamických vlastností směrovacích tabulek

První částí analýzy bylo sledování vývoje směrovací tabulky (anglicky *routing information base*, *RIB*) v celém uvažovaném intervalu. Výsledky sledování jsou zobrazeny v grafu na obrázku 1, ze kterého je patrný nárůst velikosti směrovací tabulky (viz „Aktuálně v RIB“) způsobený větším počtem přidaných záznamů než odebraných původních záznamů. Z pohledu vytyčeného cíle analýzy je však nejpodstatnější informací celkové množství změněných záznamů. Oproti stavu na začátku října 2013 se na konci února 2014 vyskytovalo ve směrovací tabulce přibližně 50 000 nově přidaných záznamů a více než 36 000 původních záznamů bylo odebráno. Největší podíl změn (přes 200 000) však připadá na aktualizace původních záznamů, které reprezentují jednu či více posloupností odebrání a následného navrácení záznamu do směrovací tabulky. Z grafu je také na poklesu počtu nezměněných původních záznamů patrné, že během 5 měsíců došlo k obměně (odebrání či aktualizaci) téměř poloviny původních záznamů směrovací tabulky.

Po zjištění trendů vývoje změn ve směrovací tabulce v průběhu celého sledovaného období byly tyto trendy sledovány v průběhu jednoho dne. Histogramy na obrázku 2a zobrazují počty změn v jednotlivých částech dne v průměru za 5 měsíců sledovaného období. Z uvedeného grafu je patrné větší množství změn záznamů ve směrovací tabulce během pracovních hodin. Uvedené průběhy také potvrzují zjištění z první části analýzy, že největší podíl změn připadá na aktualizace záznamů. Hlavním zjištěním je však skutečnost, že aktualizace (tj. odebrání a následné navrácení záznamu do směrovací tabulky) často proběhne za méně než 2 hodiny. Tuto vlastnost lze využít při implementaci operace LPM, konkrétně při aktualizaci prefixu v datové struktuře. Vzhledem k předpokládanému opětovnému přidání odebíraného prefixu během následujících 2 hodin není nutné jeho skutečné odebrání, ale je možné jej realizovat nastavením příznaku neplatnosti, který bude s opětovným přidáním prefixu vynulován.



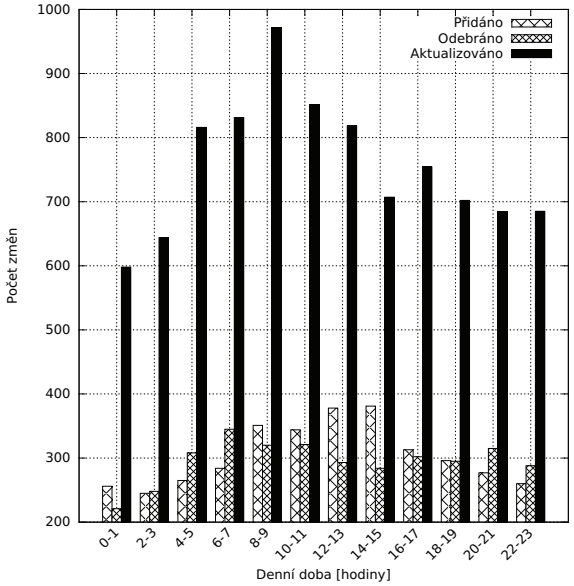
Obrázek 1: Vývoj směrovací tabulky v období 5 měsíců

Poslední část analýzy byla věnována sledování délky výskytu záznamu ve směrovací tabulce. Výsledky tohoto pozorování, zobrazené na obrázku 2b, se vztahují pouze na záznamy, které byly ze směrovací tabulky odstraněny, a tudíž u nich byla známa celá délka jejich výskytu v RIB. Na rozdíl od předcházejících částí analýzy byl každý výskyt opakovaně přidávaného a odebíraného prefixu započítán samostatně a jeden prefix tudíž mohl přispět k výsledkům analýzy několika různými hodnotami délky výskytu v RIB. Graf na obrázku 2b představuje kumulativní funkci délky výskytu záznamu v RIB a udává podíl záznamů, které byly ve směrovací tabulce přítomny uvedenou či kratší dobu. Z grafu lze tedy například vyčíst, že 70 % odstraněných záznamů se ve směrovací tabulce vyskytovalo méně než 1 den. V rámci sledovaného 5měsíčního období byl nejdelší výskyt záznamu 284 dnů a v průměru byly záznamy ze směrovací tabulky odstraňovány za 9 dnů a 4 hodiny. Medián délky výskytu záznamu v RIB byl ale 35 minut a lze tedy konstatovat, že více než polovina odebraných záznamů stráví ve směrovací tabulce méně než 1 hodinu.

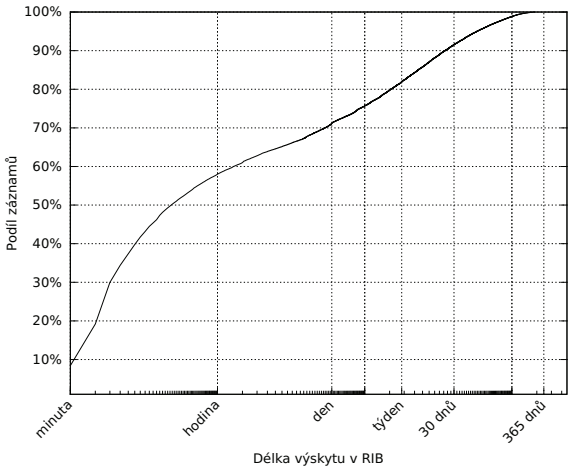
5 Cíle disertační práce

Aktualní situace v páteřních sítích klade vysoké požadavky na implementaci operace směrování paketů. Kvůli podpoře vysokých přenosových rychlostí je nutné akcelerovat směrování paketů v hardware. Velké množství záznamů ve směrovacích tabulkách také vynucuje využití speciálních paměťově efektivních reprezentací sady prefixů ze směrovací tabulky a efektivní nakládání s přidělenou pamětí během provádění aktualizací směrovacích informací.

V rámci své disertační práce se zabývám možnostmi akcelerace operace LPM s využitím technologie FPGA. Požadovanou rychlost zpracování lze poměrně snadno zajistit implementací operace LPM



(a) Změny směrovací tabulky během dne



(b) Délka výskytu záznamů ve směrovací tabulce

Obrázek 2: Vývoj směrovací tabulky v různých časových intervalech

zřetězenou linkou. Problematickým je však dostatečně rychlý přístup do paměti. V první části disertační práce jsem se proto věnoval analýze stávajících LPM algoritmů, především z hlediska jejich paměťové náročnosti při reprezentaci prefixových sad z aktuálních směrovacích tabulek páteřních směrovačů. Na základě této analýzy jsem následně navrhl novou reprezentaci prefixových sad a hardwarovou architekturu pro její zpracování. Tato reprezentace umožňuje uložit kompletní prefixové sady z páteřních směrovacích tabulek v rychlé paměti na FPGA čipu. Navržená hardwarová architektura podporuje propustnost přes 170 Gb/s. Výsledky dosažené v rámci první části disertační práce jsem publikoval v [3].

Dynamickou povahu směrovací tabulky je nutné reflektovat při jejím uložení v paměti pro potřeby operace LPM. Vzhledem k použití zřetězené linky a oddělených paměťových bloků v jejích jednotlivých stupních by statické přidělení rezervní paměti mohlo vést k jejímu neefektivnímu využití. Proto jsem se v druhé části disertační práce nejprve zaměřil na analýzu dynamických vlastností směrovacích tabulek, jejíž popis je obsahem tohoto příspěvku. Provedená analýza navazuje na podobné práce v této oblasti, přičemž posouvá předmět zájmu z aktualizací vyměňovaných mezi směrovači na samotnou směrovací tabulku. Znalosti získané touto analýzou a sada skriptů vytořená při jejím sestavování následně poslouží k bližšímu pohledu na vztah změn ve směrovacích tabulkách a paměťových nároků v jednotlivých stupních zřetězené linky. Aktualizace směrovací tabulky budou transformovány na aktualizace stromové datové struktury reprezentující sadu prefixů a na jednotlivých hladinách datové struktury budou sledovány změny v paměťových nárocích během provádění aktualizací. S touto znalostí se pak budu věnovat návrhu a implementaci systému dynamického přidělování paměti na čipu FPGA, jehož vytvoření je hlavním cílem druhé části disertační práce.

6 Závěr

Zatímco se existující studie dynamických vlastností směrování v páteřních sítích zaměřovaly především na analýzu aktualizací zasílaných mezi páteřními směrovači a jejich klasifikaci do kategorií odpovídajících příčinám těchto změn, žádná práce se nevěnovala vlivu aktualizací na směrovací tabulku z pohledu implementace směrování v páteřním směrovači. Tento příspěvek proto představuje analýzu poskytující chybějící pohled na dynamickou povahu směrování v páteřních sítích.

Provedená analýza ukázala, že obměna poloviny směrovací tabulky trvá více než 5 měsíců a nezanedbatelná část záznamů je tudíž dlouhodobě stabilních. Hlavní příčinou změn ve směrovací tabulce jsou aktualizace stávajících záznamů (jejich odebrání a opětovné přidání), u nichž se při bližším pohledu ukázalo, že velmi často proběhnou za méně než 2 hodiny. Tato vlastnost by mohla být využita při implementaci aktualizací záznamů, například pomocí příznaku platnosti. V rámci analýzy bylo také ukázáno, že se 70 % odstraňovaných záznamů vyskytovalo ve směrovací tabulce méně než 1 den.

Výsledky analýzy představené v tomto příspěvku budou doplněny o studii vztahu změn ve směrovacích tabulkách a paměťových nároků v jednotlivých stupních zřetězené linky implementující operaci LPM. Tyto informace následně poslouží pro návrh a implementaci systému dynamické alokace paměti na čipu FPGA v rámci disertační práce *Využití rekonfigurovatelných obvodů v oblasti počítačových sítí*.

Poděkování

Tato práce byla podpořena Evropským fondem regionálního rozvoje (ERDF) v rámci projektu Centra excelence IT4Innovations (CZ.1.05/1.1.00/02.0070) a dále projektem Architektury paralelních a vestavěných počítačových systémů (FIT-S-14-2297).

Reference

[1] IEEE Computer Society: Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications; Amendment 4: Media Access Control Parameters, Physical Layers, and Management Parameters for 40 Gb/s and 100 Gb/s Operation. IEEE std 802.3ba-2010, June 2010. ISBN 978-0-7381-6322-2.

[2] (2014, Jun.) IPv6 / IPv4 Comparative Statistics. [Online]. Available: `http://bgp.potaroo.net/v6/v6rpt.html`

[3] J. Matoušek, M. Skačan, and J. Kořenek: Memory Efficient IP Lookup in 100 Gbps Networks. In 23rd International Conference on Field Programmable Logic and Applications (FPL'13), Porto: IEEE Circuits and Systems Society, 2013. ISBN 978-1-4799-0004-6.

[4] C. Labovitz, R. G. Malan, and F. Jahanian: Internet Routing Instability. IEEE/ACM Transactions on Networking, vol. 6, no. 5, pp. 515–528, October 1998, ISSN 1063-6692.

[5] J. Li, M. Guidero, Z. Wu, E. Purpus, and T. Ehrenkranz: BGP Routing Dynamics Revisited. ACM SIGCOMM Computer Communication Review, vol. 37, no. 2, pp. 5–16, April 2007, ISSN 0146-4833.

[6] (2014, Jun.) RIS Raw Data – RIPE Network Coordination Centre. [Online]. Available: `http://www.ripe.net/data-tools/stats/ris/ris-raw-data`

Application of Evolutionary Computing for Optimization of Functional Verification *

Marcela Šimková
Computer Science and Engineering, 3rd year, (full-time study)
Supervisor: Zdeněk Kotásek

Faculty of Information Technology, Brno University of Technology
Božetěchova 2, Brno 612 66

isimkova@fit.vutbr.cz

Abstract. This paper introduces a new method for optimization of *coverage-driven verification* (CDV) that is based on evolutionary computing. In comparison to the classical CDV that utilizes random search, using this method, the convergence to the maximum coverage is much faster, fewer transactions are used and no manual effort is required from the user. Moreover, the optimization is targeted to the verification process itself without the dependence on the circuit that is verified.

Keywords. Functional verification, Optimization, Evolutionary algorithms.

1 Introduction

According to The 2012 Wilson Research Group Functional Verification Study [1], several challenges that are a hot topic in the verification field exist nowadays. For example, defining of appropriate metrics to measure the progress in verification, restricting the time needed to discover a next bug and the time to isolate and resolve the bug. Nevertheless, the most important challenges are creating sufficient tests to verify the whole design and managing the verification process. Therefore, new methods that target these issues are in a great demand, despite they are in the centre of interest in many companies.

In this paper, the attention is paid to functional verification as it is the most dominant simulation-based verification approach used in industry. A new method for optimization of the verification process is introduced. It is based on evolutionary computing, in particular the genetic algorithms, and accelerates reaching coverage closure of measurable properties determined by the specification.

The paper is organized as follows. Section 2 introduces functional verification and the process of CDV. In Section 3, evolutionary computing is outlined and the decision why the genetic algorithm was selected for optimizing CDV is explained. The structure of the genetic algorithm adapted for CDV is described. The experimental part is incorporated into Section 4. The optimization is directly applied to functional verification of a selected circuit. Section 5 concludes the paper and clarifies, how the CDV optimization correlates to the topic of the PhD thesis.

2 Coverage-Driven Verification

Functional verification is based on simulation and uses sophisticated testbenches with additional features to increase the efficiency of verification. First of all, it generates a set of constrained-random test vectors

*This work was supported by the EU COST Action IC1103 MEDIAN, the national COST project LD12036, the scholarship programme Brno PhD Talent, and the BUT FIT project FIT-S-14-2297.

called transactions (constraints define their correct form) and compares the behaviour of *Device Under Test* (DUT) with the behaviour specified by a provided reference model. The reference model is prepared according to the specification in SystemVerilog, in C/C++ or other languages that are supported.

Coverage is a very important feature in functional verification and defines the quality of verification tests based on the incorporated attributes/properties of the verified system (also called coverage metrics). In other words, it measures whether enough transactions were produced in order to exercise most of the possible behaviours of the circuit. The list of supported coverage metrics follows, some of them are provided automatically in an HDL simulator, other must be written by hand.

- *Functional coverage* is implemented manually, measures how well input transactions cover the specification of the verified design. It focuses mostly on the semantics. For more precise definition, see Chapter 4 in [3] or Chapter 18 in IEEE SystemVerilog standard [4]. In the example in Listing 1, one statement from the specification is selected: For the input X (integer), at least one negative, one positive and one zero value has to be checked. For this statement, the coverage item in SystemVerilog is constructed according to the standard. This item is automatically loaded by the HDL simulator and its occurrence is recorded during verification.

```
coverpoint input_X {
    bins neg = {[$:-1]};  // Check at least one negative value
    bins zero = {0};      // Check zero value
    bins pos = {[1:$]};   // Check at least one positive value
}
```

Listing 1: An example of functional coverage item in SystemVerilog.

- *Structural coverage* is generated automatically by a simulation tool, measures how well input transactions cover the implementation (the source code) of the verified design. Typical structural coverage metrics are toggle, statement, branch, condition, expression or FSM coverage. For more precise definition, see Chapter 5 in [3] or Chapter 29 in IEEE SystemVerilog standard [4]. In Figure 1, an example of the lines of code that were not covered during verification can be seen. In particular, X_B means that one or more branches were missed, X_T specifies which branch it is, X_S means that one or more statements were missed. The tick mark means that the line of code was properly executed.

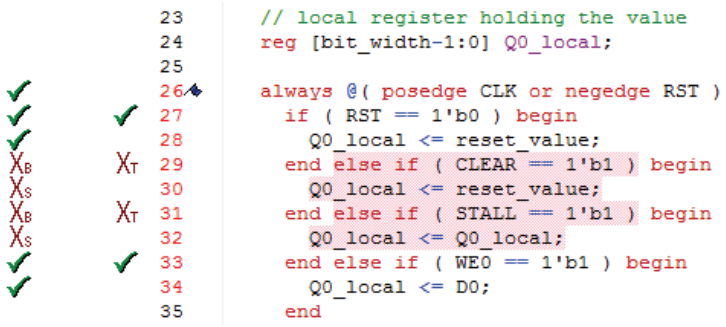


Figure 1: The missed code coverage statements recorded by the ModelSim simulator.

Coverage closure means provoking the occurrence of each of the measurable properties [3]. HDL simulators like ModelSim from Mentor Graphics, Riviera-PRO from Aldec, or Incisive Enterprise Simulator from Cadence offer coverage analysis (they measure all coverage metrics automatically) and produce statistics about which coverage items were hit during verification runs. If there are holes (unexplored areas) in the coverage analysis, the verification effort is directed to the preparation of test scenarios which will cover these holes. One option is to change the constraints of the pseudo-random generator, the second option is to prepare direct tests. This process is called *coverage-driven verification* (CDV).

For complex systems, like processors or controllers, reaching coverage closure represents a daunting task and a clue how to do this is not defined yet [2]. Maybe that is the reason why some verification teams still check coverage holes and prepare direct tests to cover such holes manually [6]. To target this issue, new techniques for automation of reaching coverage closure in CDV have to be developed. The generation of appropriate scenarios can be driven by an intelligent program that controls coverage results and chooses constraints or seed of pseudo-random number generator.

Several solutions already exist, e.g. based on machine learning techniques. In [5], Bayesian networks are applied to CDV problem. In the first step, a training set is used to learn the parameters of a Bayesian network that models the relationship between coverage and generated transactions. In the second step, the Bayesian network is used to provide the most probable transactions that would lead to a given coverage task. In [6], a tool called StressTest is introduced. The StressTest engine uses closed-loop feedback techniques to transform the internal Markov model (used for generating transactions) into one that effectively covers the user-defined points of interest. This approach is targeted to verification of microprocessors and requires an engineering team to provide a template describing interface protocols of the system. The authors in [7] present a method for automated generation of simulation vectors based on the analysis of the HDL description and the path coverage feedback. This method utilizes constraint solving using the word-level SAT. Some of the other related solutions are inbuilt in proprietary industry tools like inFact from Mentor Graphics or VCS from Synopsys. Unfortunately, producers of these tools are usually not willing to reveal the techniques their tools use to achieve the high level of coverage.

3 Reaching Coverage Closure Using Genetic Algorithm

In our approach, we consider to apply evolutionary computing to achieve the maximum coverage automatically in a reasonable time. In CDV, the search space (coverage space) of possible solutions is defined by different coverage metrics that were mentioned in Section 2. In particular, when the coverage metric is defined by one measurable attribute of a circuit, the coverage space represents an n-dimensional region defined by n coverage metrics.

In the following subsection, different search-space algorithms are presented and the reason why the genetic algorithm is suitable for solving the CDV problem is explained.

3.1 Search-Space Algorithms

If the search space of all possible solutions (in our case covering all the properties) is big and random variables appear in the formulation of the problem, stochastic optimization algorithms can be used. The basic ones are random search, simulated annealing, hill climbing, swarm and evolutionary algorithms.

Random search algorithm generates the candidate solutions randomly. Computation ends when a good solution is found or when the limit of iterations is reached. This algorithm is weak for solving real world problems because it lacks strategy and does not exploit the knowledge gained during computation.

Local search algorithms (simulated annealing, hill climbing) are iterative algorithms that start with an arbitrary solution to a problem, then attempt to find a better solution by incrementally changing a single element of the solution (neighbourhood exploration). They are convenient to find a local optimum but it is not guaranteed that the best possible solution will be found.

The evolutionary algorithms employ a population of candidate solutions that are evolved through several generations. The quality of candidate solutions is determined by the *fitness function*. According to the fitness, the best solutions are selected and serve as parents for the next population. Offsprings are created by genetic operators, either mutation or crossover. If the algorithm works well, the average fitness function of the population is rising. It means that the algorithm is exploring a profitable part of the search space. At the same time, genetic operators ensure diversity, so the algorithm is resilient to the problem of local optima. More information about evolutionary algorithms can be found in [8].

While the local search focuses only on the neighbourhood of the good solutions, the random search moves through the whole state space but without exploring perspective areas. From this simple comparison, the evolutionary algorithms seem to be the best choice as they combine all advantages together.

For CDV optimization, we decided to use one of the evolutionary algorithms called the *genetic algorithm* (GA). GA fits best to this problem as it utilizes both genetic operators and its candidate solutions are represented by bit strings of a constant length. In some cases, GA serves just as an optimizer of specific processes and its aim is not to find the best solution but only to preserve and employ the domain knowledge. This is exactly what we need in CDV as we want to optimize the process of functional verification continuously and to utilize the domain knowledge about the reached level of coverage.

3.2 Adopted Genetic Algorithm for Optimization of CDV

According to our knowledge, applying GA for optimization of CDV is innovative as it was not widely used in this domain before. Figure 2 demonstrates how GA adjusted to the process of CDV works and the following text explains how it differs from the basic GA described in Section 3.1.

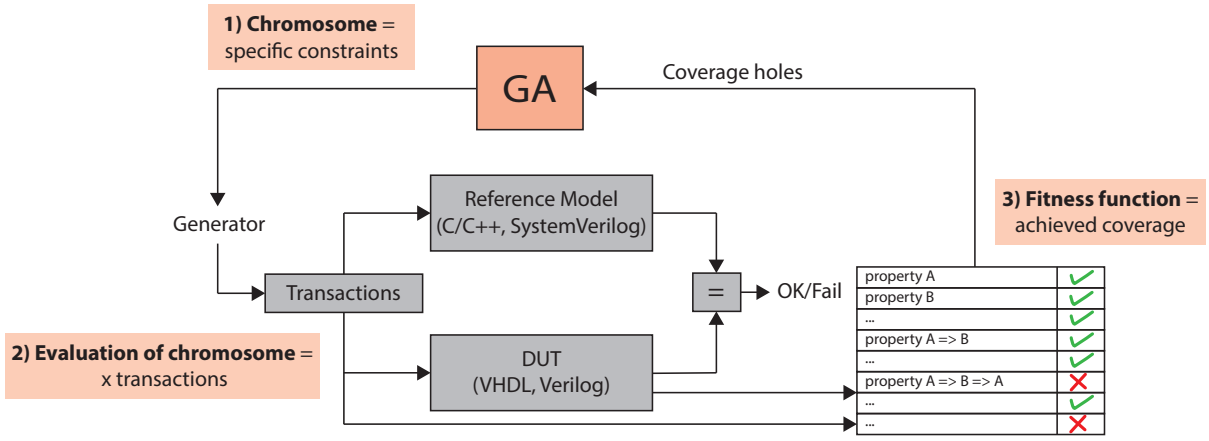


Figure 2: CDV optimized by the adopted genetic algorithm.

In our case, every candidate solution (chromosome) encodes constraints (restrictions) for the pseudo-random number generator (step 1). These constraints are represented by the probabilities of generated values for the input transactions of the verified circuit. According to these constraints, the generator produces a set of input transactions that are applied to the inputs of the verified circuit (step 2). Using these transactions, specific properties are verified and how well it is done, is reflected by the coverage measurement. The coverage status corresponds to the fitness function using which the candidate solution is evaluated (step 3). Similarly, every candidate solution of the GA population is evaluated. The best candidate solutions or their offsprings are propagated to the new population.

A huge advantage of this method is the circuit-independency. It means that optimization focuses on reaching coverage closure of the defined coverage metrics, but only these metrics are dependent on the circuit that is verified. In other words, GA only integrates coverage metrics to the definition of the optimization task, but this task is the same for every verified circuit. Therefore, this method is generally applicable for functional verification of any circuit. It will be provided as an extension of the basic functional verification environment prepared according to the *Universal Verification Methodology* (UVM) [9]. The Figure 3 highlights the components/classes that are added to the UVM environment. GA represents the core of the algorithm. Chromosome sequencer sends candidate solutions to the transaction sequencer that subsequently generates input transactions according to the constraints encoded in these candidate solutions. The structure of transactions is defined in GA transaction class.

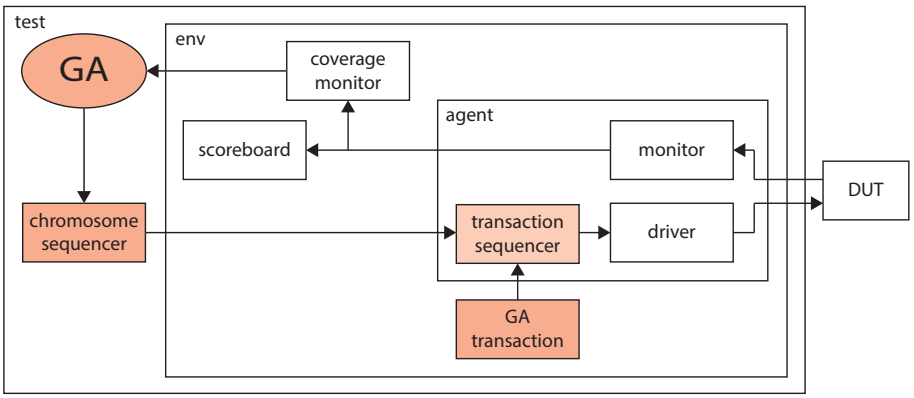
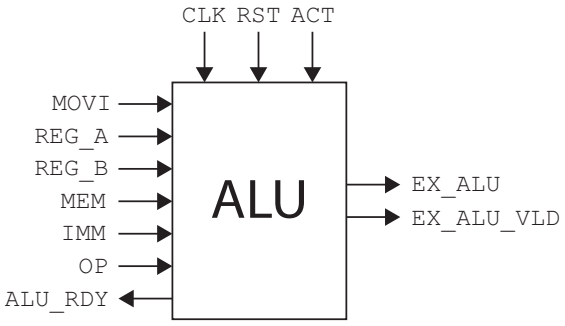


Figure 3: Extension to the UVM verification environment.

4 Case Study

As an evaluation circuit, the arithmetic-logic unit (ALU) was selected. The block diagram of ALU and the description of its signals is provided in Figure 4. For ALU, we were able to define **28 coverage scenarios** with **1989 functional properties**. The aim of verification is to check all of these properties.



- `CLK, RST, ACT` (in): the clock, reset, activation signal.
- `REG_A` (in): the first operand for every operation.
- `MOVI` (in): the selection signal, according to its value the second operand is picked either from data memory (`MEM`), register (`REG_B`), or as an immediate value (`IMM`).
- `OP` (in): the selected operation (16 options supported).
- `ALU_RDY, EX_ALU, EX_ALU_RDY` (out): output ALU signals.

Figure 4: The demonstration circuit - ALU.

At this point, three experiments with ALU are described. As our goal is to show how effective GA-driven search is in the process of CDV optimization, we decided to compare its results with basic random search and constrained random search. They are shortly described in the next paragraphs.

In GA-driven search, constraints for the pseudo-random generator are encoded into the chromosome. For ALU, constraints are represented by probabilities. At first, all possible values of control signals are specified (every control sequence is important and need to be checked). In case of data signals, ranges of all possible values are selected (as these possible values can be reduced to ”interesting” ranges using approximation). Afterwards, probabilities are defined for every control value and for every range of data values. For example, the input signal MOVI can have three valid values (00, 10, 01). In the chromosome, for every of them a number is specified that defines a probability with which these values are generated as input of MOVI. Probabilities in the initial population of candidate solutions are created randomly.

The basic random search does not specify probability constraints for generation of input transactions. Instead, they are generated randomly. This approach represents the original concept that is used in functional verification commonly. However, it can take a very long time to cover all properties, because without the coverage feedback, the generator produces transactions that cover some properties repeatedly.

The constrained random search uses probabilities for constraining the input transactions generation as GA approach does but these probabilities are generated randomly. So good constraints are not remembered and propagated further.

Because of the restricted size of the paper, just the best measures of basic random search, constrained random search and GA-driven search are visibly compared in the graph in Figure 5. As the HDL simulator, ModelSim from Mentor Graphics was used. In the background, 20 measures with different settings were performed for every search algorithm. The x-axis represents the number of required input transactions and the y-axis represents the achieved level of coverage of functional properties for ALU. GA achieves much better results than both random approaches. The convergence to the maximum coverage is significantly faster and the number of required transactions is lower. It can be stated that for ALU, GA really drives the generation of input transactions successfully.

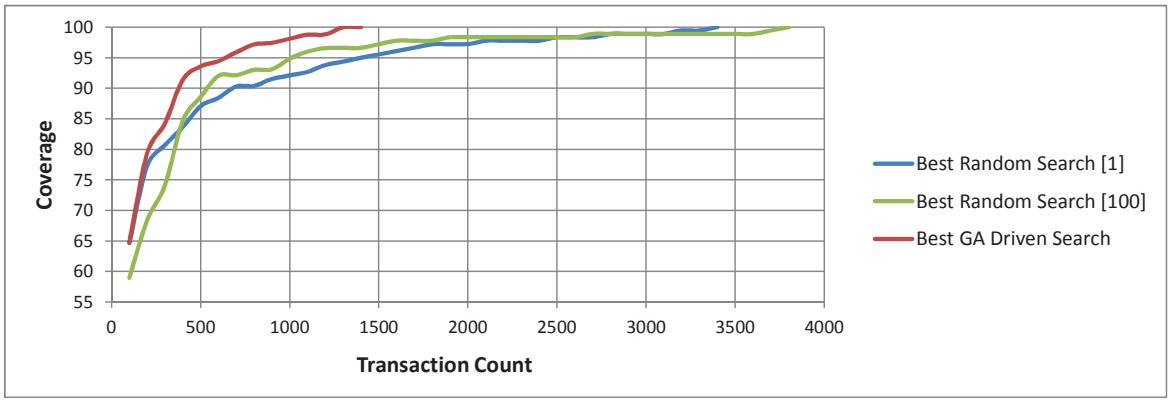


Figure 5: The comparison of the best results.

5 The Goals of the PhD Thesis

This paper introduced a new method for optimization of CDV based on the adapted genetic algorithm. Together with the hardware acceleration of verification runs and the automated generation of UVM environments, this method represents the core part of the PhD thesis. Throughout the last year, the algorithm was fully implemented, integrated into UVM and evaluated. The results of the experiments show that the algorithm works well when the speed of the convergence to the maximum coverage and the amount of transactions are considered.

In the next months, we plan to evaluate the GA-driven method on the RISC processor in order to show its generality and scalability to complex systems. In this case, we will use functional, code and instruction coverage as a coverage feedback. Afterwards, the PhD thesis will be written.

References

[1] Mentor Graphics. The 2012 Wilson Research Group Functional Verification Study. 2013. `https://verificationacademy.com/seminars/2012-functional-verification-study`

[2] A. Molina, and O. Cadenas. Functional Verification: Approaches and Challenges. *Latin American Applied Research*, 2007, pp. 65–68.

[3] A. Piziali. Functional Verification Coverage Measurement and Analysis. Springer, 2004, ISBN: 978-0-387-73992-2.

[4] IEEE Standard 1800-2005 for SystemVerilog - Unified Hardware Design, Specification, and Verification Language. IEEE, 2004, ISBN: 0-7381-4811-3.

[5] S. Fine, and A. Ziv. Coverage Directed Test Generation for Functional Verification using Bayesian Networks. In *Proc. of DAC'03*, pp. 286–291, June 2-6, ACM, USA.

[6] I. Wagner, V. Bertacco, T. Austin. Microprocessor verification via feedback-adjusted Markov models. In *Proc. of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1126–1138, 2007.

[7] Y. Guo, W. Qu, T. Li, S. Li. Coverage Driven Test Generation Framework for RTL Functional Verification. In *Proc. IEEE Computer-Aided Design and Computer Graphics*, pp. 321–326, 2007.

[8] G. Rozenberg, T. Bäck, and J.N. Kok. Handbook of Natural Computing. Springer-Verlag, Berlin Heidelberg, 2012, p. 2052, ISBN: 978-3-540-92909-3.

[9] Mentor Graphics Verification Academy. UVM Cookbook. 2014. `https://verificationacademy.com/cookbook/uvm`.

Time and Frequency Transfer in Local Networks

Jiří Dostál
Informatics, 3rd class, full-time study
Supervisor: Vladimír Smotlacha

Faculty of Information Technology, Czech Technical University in Prague
Thákurova 9, 160 00 Prague 6

jiri.dostal@fit.cvut.cz

Abstract. This paper deals with three main topics: time protocols, an atomic clock timescale comparison and a precise timestamping. There is described a theoretical background and a state-of-the-art approach. The main topic is the research in the field of time transfer protocol IEEE 1588 – PTP. It is aimed at one of essential tasks – the precise timestamping. The design of accurate FPGA based time measurement device with an interpolating counter is described as well. Achieved results were verified and utilized in adapters for accurate time transfer in optical links.

Keywords. measurement, precise time, frequency, IEEE 1588, PTP, FPGA, interpolating counter, transparent clock, timestamper

1 Introduction

Time is a SI base physical quantity and has very broad area of influence for all people and application fields. The need for precise time and frequency synchronization between devices with microsecond or better accuracy is nowadays challenging task for both scientific and engineering point of view. There are also new fields of precise time application e.g. finance and high frequency trading. Timekeeping is a specialized branch that deals with precise time management. As we have a precise time, another problem is the distribution of this time to other timekeeping devices. Many methods of time transfer are employed (e.g. satellite transmission). In recent times, a new method is proposed – time transfer over universal optical networks.

Time Protocols In the scope of computer networks the NTP protocol is the most dominant but does not require such a strict resolution of timestamps. Another case is a modern IEEE 1588 protocol also known as a Precision Time Protocol (PTP) which manipulates with nanoseconds and sub-nanoseconds resolution of timestamps. My research is focused also on the PTP protocol especially on the time distribution infrastructure. A transparent clock (TC) node is a part of PTP hierarchy but in the present days TC are not fully available with desired quality of operation. Another field for HW-based precise time measuring systems are specialized applications for a time distribution e.g. for a time scale comparison between distant atomic clocks.

Clock Comparison Nowadays, an accurate time signal is mostly acquired from the global positioning system (GPS). The Common-View GPS satellite method [6] is used for the atomic clock's time-scale calibration as well. Since the GPS time transfer is prone to the accuracy degradation at distances over

1000 km or there is a problem with a GPS antenna or receiver installation, an alternative method has been developed – the precise time and frequency transfer in optical networks. A fiber-optic cable based network can carry a signal up to 2000 km utilizing an optical amplification only.

Timestamping Modern applications for time distribution demands a precise timestamping of external asynchronous events. There is a need for a sub-nanosecond resolution in such cases – this means that the required timestamp's resolution is below the clock period of most digital systems. Generally, it is necessary to generate and evaluate timestamps with a time interval which is shorter than a system clock period. Examples of the precise timestamping are embedded interval counter in a control systems or a network-based application for the time distribution (details about HW support in [4]).

2 Background and State-of-the-Art

2.1 IEEE 1588

The IEEE 1588 (also know as the "Precise Time Protocol" – PTP) protocol was developed for the need of precise time distribution wit more precise synchronization (in comparison to the NTP protocol [16]). It is not a RFC by IETF but it is standardized under IEEE supervision. This protocol is intended to be an standard for devices connected via switched IP networks. The accuracy of synchronization is intended to be beyond one microsecond.

2.1.1 Transparent Clock

The physical layout of a machine determines the topology of an automation network, which is in many cases a daisy chain. When such a topology is built up with BCs, the result is a chain of control loops which is susceptible to error accumulation. That's why the automation community has proposed the new clock type TC. This is an Ethernet bridge which is capable to measure the residence time of PTP event messages, i.e. the time the message has spent in the bridge during transit. Because the residence time is the difference of two timestamps, the TC does not need to be synchronized. It is sufficient if it can measure short time intervals with reasonable accuracy. Syntonization of the local timer improves accuracy. The residence time of the traversed TCs is summed up in the correction field of the Sync message, if the TC is capable to modify the correction field on the fly, or in the respective Follow_Up message [19].

End-to-End Transparent Clock In the case of end-to-end (e2e) transparent clock, the slave measures the delay to the master with an end-to-end delay request/response message exchange 1.

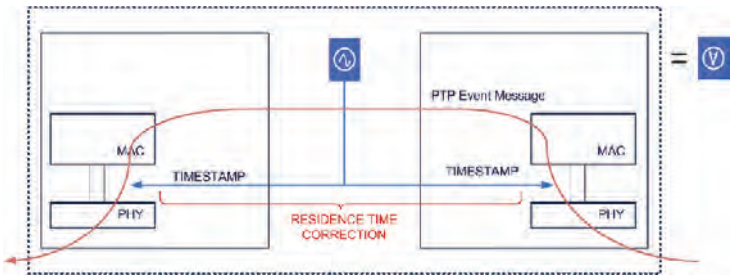


Figure 1: End-to-end transparent clock [20].

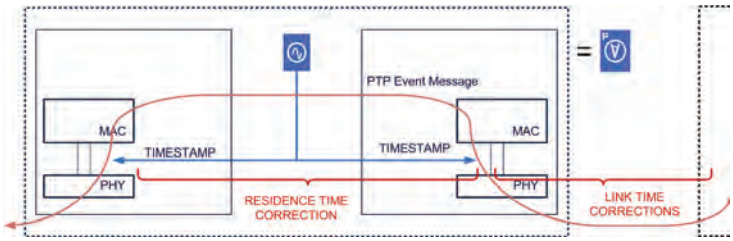


Figure 2: Peer-to-peer transparent clock [20].

Peer-to-Peer Transparent Clock The peer-to-peer (p2p) transparent clock measure the link delay to all neighboring clocks with Pdelay_Req/Pdelay_Resp messages. A third message type may be required for this purpose, the Pdelay_Follow_Up. When a Sync traverses a p2p transparent clock, not only the residence time is added to the correction field but also the uplink delay, i.e. the delay of the link over which the Sync has been received [19].

3 Time Transfer over Optical Network

There is a basic scheme of two atomic clock systems comparison in figure 3. The time transfer method relies on symmetrical transport delay in both directions. Two systems are connected by a bidirectional optical link. Each system is provided by a 1PPS signal from local clock and each systems has two outputs: Tr is a 1PPS signal received via optical interface from the other system and Ts represents epoch in which was sent out the encoded 1PPS signal.

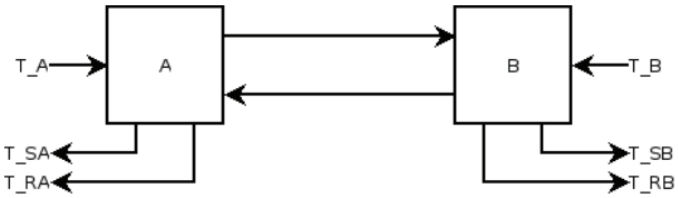


Figure 3: Time transfer method.

Both t_S and t_R signals are connected to STOP inputs of two time interval counters (TIC). The first TIC measures interval x between local 1PPS and t_R (i.e. difference between local second and received second from remote site) and the second TIC measures delay ϵ of processed 1PPS inside the transmitting part of the system. 1PPS pulse from a local clock arrives to system A in time t_A. It is transmitted by system A through the optical fiber to the remote site in time t_{SA} and the reception is signalized by system B in time t_{RB}. Analogically 1PPS pulse from remote clock raised in time t_B is transmitted by system B in time t_{SB} and received by system A in time t_{RA}. Here $\Theta_{AB} = t_B - t_A$ is the clock offset, $\epsilon_{Si} = t_{Si} - t_i, i = A$ B is the delay of system i and $\delta_{AB} = t_{RB} - t_{SA}, \delta_{BA} = t_{RA} - t_{SB}$ is the link delay from site A to site B and from site B to site A respectively.

Using a pair of time interval counters at both sites it possible to measure the system delays and the time intervals:

$$\begin{aligned} x_A &= t_{RA} - t_A = \Theta_{AB} + \epsilon_{SB} + \delta_{BA} \\ x_B &= t_{RB} - t_B = -\Theta_{AB} + \epsilon_{SA} + \delta_{AB} \end{aligned} \tag{1}$$

On a symmetrical link where the delay in both directions is $\delta = \delta_{AB} = \delta_{BA}$, the clock offset may be calculated as:

$$\Theta_{AB} = \frac{((x_A - x_B) + (\epsilon_{SA} - \epsilon_{SB}))}{2} \tag{2}$$

4 Actual and future work

5 IEEE 1588 Timestamper

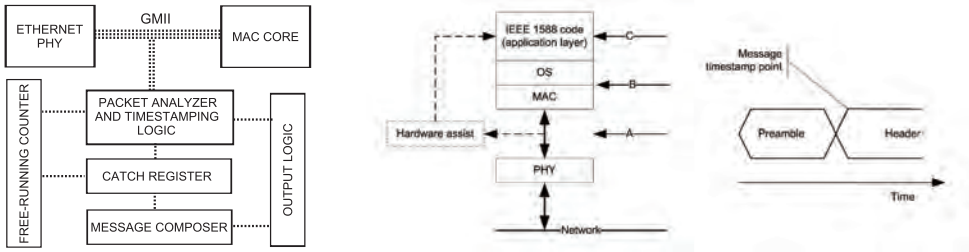


Figure 4: Simplified block diagram of the IEEE 1588 timestamper and PTP timestamp generation model

IEEE 1588 is a relatively new protocol standard for a precision clok synchronization. It operates mostly over TCP/IP networks and Ethernet. The protocol is also known as the Precision Time Protocol – PTP. Synchronization architecture is a master–slave model with nodes comunicating primary by multicast. The main difference between PTP and its predecessor NTP is that the PTP enabled nodes have to be equipped by some HW support to precise the delay measurement. You can find more about this protocol in [17].

The IEEE 1588 is a device which creates timestamps of incoming/outcomming packets in network interface hardware and the timestamp is further used in the PTP functionality. The timestamper is placed between the PHY and MAC layer on a MII interface and listen to the traffic. Every PTP packet is timestamped so if there is a lag in the network hardware between the composition and sending, we will know the correct time of physical transmission of the packet. The correct timestamp of a Sync message is sent as a Follow up message.

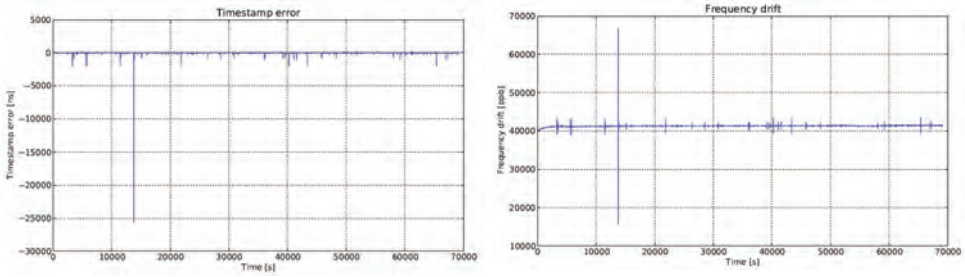


Figure 5: IEEE 1588 timestamper evaluation.

In figure 4, there is a simplified block diagram of the FPGA counter with carry chain interpolation. It utilizes a coarse free-runnig counter driven by reference frequency. The intervals within one clock period is measured by the tapped delay line interpolator (carry chain implementation). The propagation rate of the delay line is computed in the pipelined priority encoder. Measured values are stored in the catch registers and sent by the output logic. The I2C block reads service information from transceivers

and manage the frequency synthesis on the daughter card. The timestamper is implemented as IP core for FPGA in VHDL language. The timestamping core is configurable and can operate on Layer 2, 3 and 4 of the ISO/OSI model and also the reference frequency of the free-running counter. You can see a timestamp generation model in figure 4.

I am working on the timestamper evaluation now. In figure 5 you can see some preliminary measured data. There is one issue of an accidental glitch which is the subject of a further research.

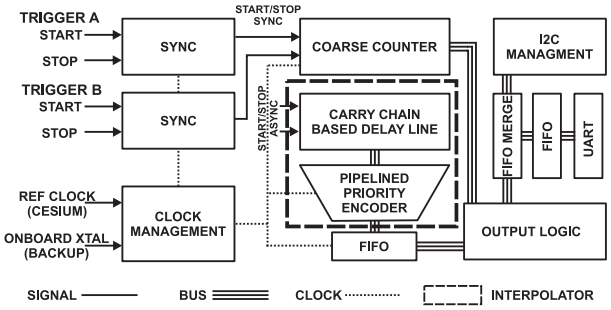


Figure 6: Simplified block diagram of the FPGA counter with carry chain interpolation.

5.1 Atomic clock time scale comparison

Generally, atomic clocks generate a time scale which is the subject of comparison. An elementary method is depicted in Figure 7. We have local and remote system consisting of an atomic clock and an adapter. Both systems are interconnected via an optical network and the objective is to measure a difference between local and remote one pulse per second (1PPS) signals (the 1PPS signal is commonly used in a timekeeping – there is a rising-edge every second). Some variation of this difference is expected and has two originis: a temperature expansion of the optical fiber and a mutual shift of both atomic clocks time scales. As described in [9], for the optical path with equal delay in both directions, we can apply formula:

$$\Theta_{AB} = \frac{((x_A - x_B) + (\epsilon_{SA} - \epsilon_{SB}))}{2} \tag{3}$$

Θ_{AB} stands for clock offset, x for receive and ϵ for send delay. Indexes A and B determine the measuring site. The prefix S indicates that the value is sent from the system. The formula is valid only for a symmetric link.

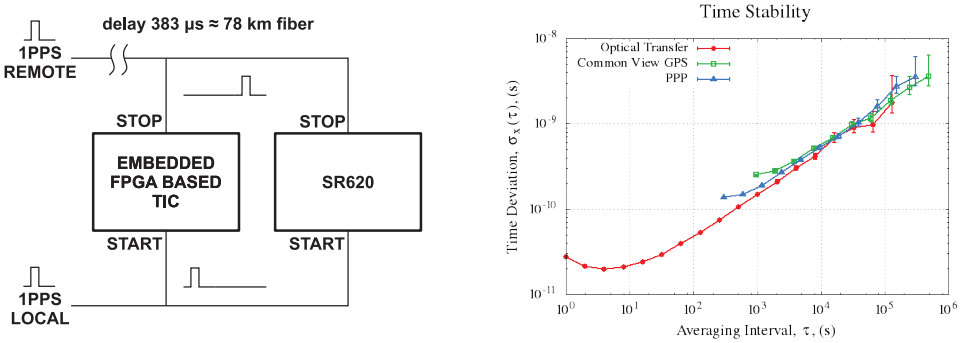


Figure 7: TIC evaluation and time stability.

The graph in Figure 7 presents comparison of our time transfer method with standard GPS based methods: Common-View (CV) and Precise Point Positioning (PPP). The optical transfer utilizes our new

generation adapter with FPGA based counters. As can be seen, our method provides better resolution and stability and lower noise (in terms of TDEV) compared to the GPS based method. The CV method uses data in the format CGGTTS [18], where the granularity is 960s. The PPP has computation period of 300 s. This is the reason that there are no data for CV and PPP methods at the beginning of the graph (until time interval 300 s or 960 s). The lowest observed noise is about 20 ps using averaging interval of 4 s. Detailed description of TDEV statistics and GPS based methods is out of scope this paper. We can conclude that the FPGA based counter is suitable for the atomic clock comparison as it is not worse than standard GPS based methods.

6 Proposed Doctoral Thesis Structure

Transparent clock with deterministic delay Deterministic delay of Transparent clock is challenging idea that might simplify the deployment of IEEE 1588 to many simple client systems. The client can estimate delay from Master clock by knowing number of passing segments especially in geographically small area, i.e. LAN or MAN. According to the opinion, such topic was not yet published. The goal is to implement an working design as a standard ethernet switch auxiliary system. I have already started collaborate on it with my colleagues from Department of Measurement, FEE CTU.

Time transfer over optical networks I continue in my work on optical transfer adapters. I focus on the system evaluation and adapter design improvement (e.g. two delay-lines support, online calibration...). Challenging topics is delay stabilization in either optical or electrical domains – there already exist several approaches, however all of them have same drawbacks and generally accepted solution was not found yet.

FPGA-based time interval counters I will work on refinement of FPGA-based interpolating counters. The performance, accuracy and stability of TIC can be improved and it is goal of my future work. Additionally, the future work on optical transfer and transparent clocks is based on this design.

References

[1] J. Dostál and V. Smotlacha, *Atomic Clock Comparison Over Optical Network,* IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2013, Accepted.

[2] J. Dostál and V. Smotlacha, *The Hardware Architecture and Device for Accurate Time Signal Processing,* 11th East-West Design & Test Symposium (EWDTS), 2013.

[3] J. Dostál, *Precision Time and Frequency Distribution,* Počítačové architektury a diagnostika (PAD), 2013, pp. 99-103, ISBN 978-80-261-0270-0.

[4] J. Dostál, *Hardware Support For Precise Time and Frequency Distribution,* Embedded Systems Workshop (EWS), 2013.

[5] J. Dostál, *Přenos času a frekvence v lokálních sítích,* Počítačové architektury a diagnostika (PAD), 2012, pp. 43-48, ISBN 978-80-01-05106-1.

[6] D. W. Allan and M. A. Weiss, *Accurate Time and Frequency Transfer During Common-View of a GPS Satellite,* 34th Annual Frequency Control Symposium, pp. 334–346, May 1980.

[7] V. Smotlacha, A. Kuna and W. Mache, *Time Transfer Using Fiber Links,* in Proceedings of the EFTF 2010.

[8] V. Smotlacha, A. Kuna and J. Vojtěch, *Optical Infrastructure for Time and Frequency Transfer,* in Proceedings of the EFTF 2013.

[9] V. Smotlacha, A. Kuna and W. Mache, *Time Transfer in Optical Network,* in Proceedings of the 42nd Annual Precise Time and Time Interval (PTTI) Systems and Applications Meeting, Reston, Virginia, USA, 2010, pp. 427-436.

[10] S. Loffredo, *Design, construction and tests of a high resolution, high dynamic range Time to Digital Converter,* 2010.

[11] A. Aloisio, P. Branchini, R. Cicalese, R. Giordano, V. Izzo, S. Loffredo and R. Lomoro, *High-resolution time-to-digital converter in field programmable gate array,* in Proceedings of Topical Workshop on Electronics for Particle physics (TWEPP), 2008.

[12] K. Pedersen, *Low cost, high performance frequency/interval counters,* 2008.

[13] J. Kalisz, *Review of methods for time interval measurements with picosecond resolution,* Metrologia, Vol.41, No.1, pp. 17–32, 2004.

[14] J. Kalisz and R. Szplet, *A PC-based time interval counter with 200 ps resolution,* 2003.

[15] C. Favi and E. Carbon, *A 17 ps Time-to-Digital Converter Implemented in 65nm FPGA Technology,* 2009.

[16] D. Mills, *Network Time Protocol (NTP),* RFC 958, Linkabit, September 1985.

[17] *IEEE standard for a precision clock synchronization protocol for networked measurement and control systems.* New York, 2008. ISBN 978-073-8154-008.

[18] US Naval Observatory, *About the CGGTTS data format*

[19] H. Weibel, *Technology Update on IEEE 1588: The Second Edition of the High Precision Clock Synchronization Protocol,* 2009.

[20] R. Cohen, *Precision Time Protocol: IEEE1588v2,* TICTOC BOF IETF Prague 2007.

MĚŘENÍ KRÁTKÝCH ZPOŽDĚNÍ S POUŽITÍM NEEKVIDISTANTNÍ FOURIEROVY TRANSFORMACE

Karel Dudáček
Informatika a výpočetní technika, 3. ročník, prezenční studium
Školitel: Vlastimil Vavřička

Fakulta aplikovaných věd, Západočeská univerzita v Plzni
Univerzitní 8, 306 14, Plzeň

karlos@kiv.zcu.cz

Abstrakt. Článek popisuje metodu měření vzájemného posuvu signálů s využitím neekvidistantní Fourierovy transformace. V prvních kapitolách jsou charakterizovány zkoumané signály a je popsána metoda měření zpoždění signálů s využitím Fourierovy transformace. Další kapitoly obsahují stručný popis neekvidistantního vzorkování a odvození neekvidistantní Fourierovy transformace. Poslední kapitola popisuje výsledky numerických experimentů.

Klíčová slova. Zpoždění, fázový posun, Fourierova transformace, neekvidistantní Fourierova transformace, NDFT, neekvidistantní vzorkování.

1 Úvod

V mnoha aplikacích je požadováno přesné měření vzájemného zpoždění analogových signálů. Pro měření zpoždění signálů s ostrými hranami bylo navrženo mnoho metod [1–3], ale pro signály bez ostrých hran jsou stále používány analogové metody. Použití analogových metod klade velké požadavky na přesnost výroby a seřízení každého vyrobeného exempláře zařízení. Použití číslicového zpracování signálu umožňuje zlevnit a urychlit vývoj a výrobu. Tento článek se zabývá číslicovou metodou měření vzájemného zpoždění analogových signálů s využitím neekvidistantní Fourierovy transformace.

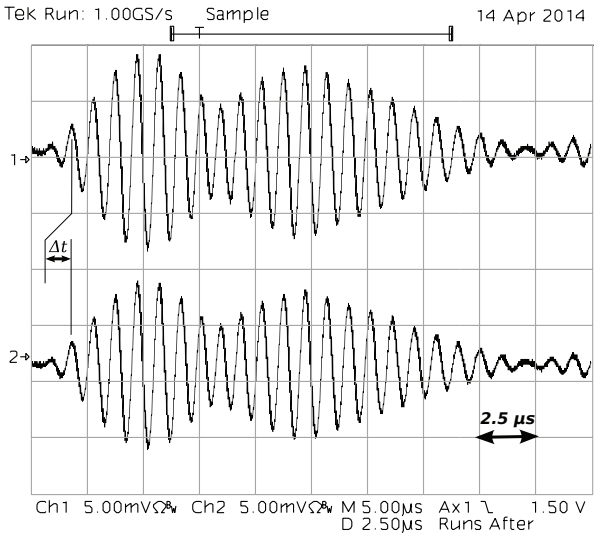
2 Vlastnosti signálů

Mějme dvojici signálů z čidel, například ultrazvukových senzorů. Předpokládejme, že dvojice signálů má následující vlastnosti:

- Signály se vyskytují v pulzech konečné délky,
- signály v páru mají stejnou obalovou křivku, ale jsou vzájemně posunuté o Δt,
- pulzy nejsou periodické,
- signály jsou tvořeny harmonickou funkcí o neznámé frekvenci f_p,
- analytický popis obalové křivky není známý.

Příklad takových signálů je na obrázku 1.

Naším cílem je měření vzájemného posuvu Δt mezi signály v páru. Časový posuv signálů je obvykle velmi malý ($\Delta t \approx 10^{-8} \div 10^{-11}\, s$) a v mnoha aplikacích je požadováno měření s vysokou přesností (řádově $10ps$). Použití jednoduché korelační metody není možné vzhledem k požadované přesnosti. Pro měření takovýchto zpoždění existuje několik metod, například výpočet zpoždění z fázového posuvu nebo korelační metoda s aproximací [4]. Vylepšením metody výpočtu zpoždění z fázového posuvu se zabývá tento článek.



Obrázek 1: Příklad signálu.
Příklad dvojice signálů zaznamenaných na prototypu ultrazvukového průtokoměru. Vzájemný posuv signálů je těžko postřehnutelný protože $\Delta t \approx 0.01 \cdot f_p^{-1}$.

3 Výpočet zpoždění s použitím Fourierovy transformace

Mějme pár diskrétních harmonických (amplitudově modulovaných) signálů $x_1(t)$ a $x_2(t)$ o frekvenci f_p. Pokud známe jejich fázový posuv $\Delta\varphi$, můžeme určit jejich zpoždění Δt podle rovnice (1). Fázový posuv $\Delta\varphi$ a frekvence f_p mohou být snadno spočteny s použitím diskrétní Fourierovy transformace.

$$\Delta t = \frac{\Delta\varphi}{2 \cdot \pi \cdot f_p} \tag{1}$$

4 Obtíže s použitím Fourierovy transformace pro měření zpoždění

Předpokládejme, že máme pár signálů popsaných v kapitole 2. Pro výpočet zpoždění musí být signály vzorkovány s frekvencí nejméně $f_s = 2 \cdot f_p$. Když je frekvence f_p velmi vysoká, musí být signály také vzorkovány s vysokou frekvencí. Když je vzorkovaný úsek zároveň dlouhý, dostaneme velké množství dat, které musíme zpracovat. To může vyžadovat velkou paměť pro jejich uložení, rychlou komunikaci pro přenos dat v reálném čase a podobně. Pro snížení množství zaznamenaných dat můžeme buď snížit vzorkovací frekvenci nebo vzorkovat kratší úsek signálu. Oba přístupy mají ovšem své nevýhody: snížení vzorkovací frekvence může způsobit problémy s aliasingem. Při zkrácení měřeného úseku může krátký impuls šumu ovlivnit výsledky [1]. Tento problém může být elegantně vyřešen použitím neekvidistantního vzorkování.

Při použití neekvidistantního vzorkování můžeme použít rozlišení dt místo vzorkovací periody $\frac{1}{f_s}$, a proto může být průměrná vzorkovací frekvence nižší než $2 \cdot f_p$. Díky tomu může být vzorkován celý signál nižší frekvencí bez rizika aliasingu a tím může být sníženo množství zaznamenaných dat. Použitím neekvidistantní Fourierovy transformace můžeme vypočítat vzájemný posuv navzorkovaných signálů.

5 Neekvidistantní vzorkování

Mějme pásmově omezený signál o šířce pásma B. Nyquistův teorém říká, že vzorkovací frekvence musí být minimálně $2 \cdot B$ aby nedošlo k aliasingu. Použitím nekonečně vysoké vzorkovací frekvence můžeme dosáhnout nekonečné šířky pásma. Ve skutečném systému je nekonečná vzorkovací frekvence nedosažitelná,

[1] Impuls šumu samozřejmě ovlivní výsledek i v případě vzorkování celého signálu, ale v tom případě nebude jeho vliv tak zásadní vzhledem k množství vzorků.

ale můžeme použít aproximace. Kdybychom vzorkovali nekonečně dlouhý signál složený z harmonických funkcí v náhodných okamžicích, dosáhli bychom nekonečné šířky pásma [6]. Ale to je také nemožné — žádný skutečný signál není nekonečně dlouhý. Dalším důvodem je, že nedokážeme měřit čas s nekonečnou přesností. Při dodržení určitých podmínek je ale možné použít aproximaci.

Při vzorkování signálu v náhodných okamžicích je nutné zaznamenat čas pořízení jednotlivých vzorků t_m. Vzdálenosti mezi vzorky je možné popsat jako násobky časového kvanta dt. Výsledkem je omezené rozlišení měřeného času s rozlišením dt. (Toto časové kvantum představuje rozlišení při měření času, nikoli přesnost měření.) Časy pořízení vzorků t_m mohou být popsány jako násobky tohoto časového kvanta (2).

$$\begin{aligned} t_m &= n \cdot dt \\ &n \ldots \text{náhodné číslo} \\ &m \ldots \text{číslo vzorku} \end{aligned} \tag{2}$$

Z rovnice (2) je zřejmé, že na neekvidistantní vzorkování je možné pohlížet jako na ekvidistantní vzorkování o vzorkovací frekvenci dt^{-1} s některými vzorky chybějícími. Známe-li vzorkovací frekvenci, můžeme aplikovat Nyquistův teorém a určit maximální šířku pásma vzorkovaného signálu (3).

$$\begin{aligned} B &= \frac{1}{2 \cdot dt} \\ &B \ldots \text{šířka pásma} \end{aligned} \tag{3}$$

Pro rozložení vzorků je běžné použít vztah (4), který zaručuje plochou distribuční funkci když se čas blíží k nekonečnu.

$$\begin{aligned} t_{n+1} &= t_n + r_n \cdot dt \\ &r_n \sim Po(\frac{\overline{f_s}}{dt}); \\ &r_n \ldots \text{náhodná proměnná} \\ &\overline{f_s} \ldots \text{průměrná vzorkovací frekvence} \end{aligned} \tag{4}$$

6 Neekvidistantní Fourierova transformace

Mějme signál $x(t)$. Definice Fourierovy transformace $\mathbb{X}(k)$ signálu $x(t)$ je (5)[2] [8]. N je celkový počet vzorků, t_n je čas získání vzorku, $\Delta\omega$ je frekvenční krok (v kruhové frekvenci).

$$\mathbb{X}(k) = \sum_{n=0}^{N-1} x(t_n) \cdot e^{-j \cdot k \cdot \Delta\omega \cdot t_n} \tag{5}$$

Použitím frekvenčního kroku (6) a časů vzorkování $t_n = n$ dostaneme standartní definici DFT (7).

$$\Delta\omega = \frac{2 \cdot \pi}{N} \tag{6}$$

$$\mathbb{X}(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi k \frac{n}{N}} \tag{7}$$

Použitím frekvenčního kroku (8) dostaneme definici neekvidistantní DFT (9).

$$\begin{aligned} \Delta\omega &= \frac{2 \cdot \pi}{T} \\ &T \ldots \text{délka signálu (okna)} \end{aligned} \tag{8}$$

$$\mathbb{X}(k) = \sum_{n=0}^{N-1} x(t_n) \cdot e^{-j2\pi k \frac{t_n}{T}} \tag{9}$$

[2] V této a všech následujících rovnicích neuvažujeme normalizaci podle počtu vzorků

Tabulka 1: Porovnání modifikací Fourierovy transformace.

	DFT	NDFT	Padded DFT[a]
Frekvenční rozlišení	$\frac{1}{T}$	$\frac{1}{T}$	$\frac{1}{l \cdot T}$
Maximální frekvence	$\frac{fs}{2}$	$\frac{1}{2 \cdot dt}$	$\frac{fs}{2}$

[a]Doplnění nulami na l násobek původní délky.

Tato definice předpokládá spojitý čas, ale čas může být měřen jen po diskrétních časových kvantech (hodinových ticích) délky dt. Díky tomu můžeme použít počet hodinových tiků místo času (10).

$$m = \frac{t}{dt} \tag{10}$$

Dosazením (10) do (9) dostaneme (11). Neformálně řečeno jsme signál vzorkovali v N bodech z M možných.

$$\begin{aligned} \mathbb{X}(k) &= \sum_{n=0}^{N-1} x(m_n) \cdot e^{-j2\pi k \frac{m_n}{M}} \\ M &= \frac{T}{dt}; \quad m_n = \frac{t_n}{dt} \end{aligned} \tag{11}$$

$M \ldots$ délka signálu (okna) v hodinových ticích dt

$m_n \ldots$ čas vzorkování v hodinových ticích dt

Rozlišení ve frekvenci a maximální frekvence bez aliasingu běžné (DFT) a neekvidistantní (NDFT) Fourierovy transformace jsou uvedeny v tabulce 1.

Neekvidistantní Fourierovu transformaci je možno spočítat podle definice s výpočetní náročností $O(N^2)$, s použitím rychlé Fourierovy transformace (FFT) s náročností $O(M \cdot \log(M))$ nebo rychleji s použitím některého aproximačního algoritmu, například [9–11].

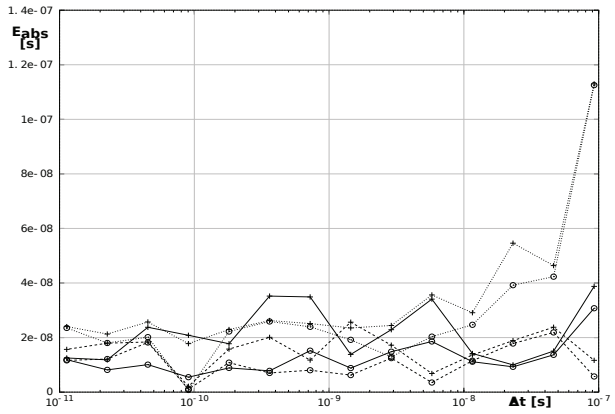
7 Experimentální ověření vlastností NDFT

Jako součást výzkumu bylo provedeno numerické ověření požadovaných vlastností neekvidistantní Fourierovy transformace. Byly vygenerovány dvojice signálů, zašuměny bílým šumem a poté byl vypočten jejich vzájemný posuv a určena absolutní a relativní chyba výpočtu. Tento postup byl proveden pro mnoho kombinací vzorkovací frekvence a úrovně šumu. Pro každou kombinaci byl výpočet proveden opakovaně a výsledky byly statisticky vyhodnoceny.

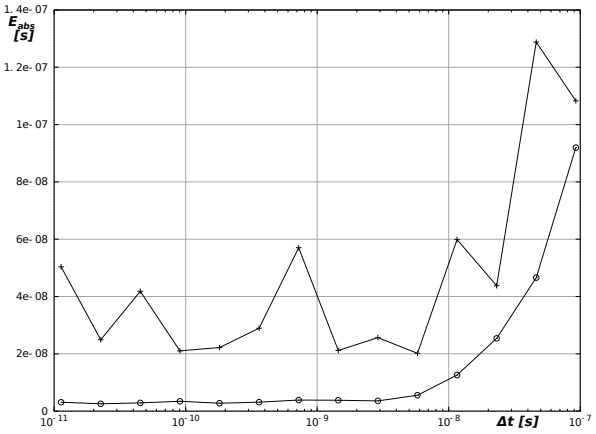
Když je SNR vysoké ($SNR \approx 20\,dB$) a vzorkovací frekvence také vysoká ($fs \approx 50 \cdot f_p$), je metoda využívající NDFT horší než korelační metody. S klesajícím odstupem signálu od šumu ($SNR \to 0\,dB$) a s klesající vzorkovací frekvencí začínají metody využívající Fourierovu transformaci podávat lepší výsledky, viz obrázek 2. Když vzorkovací frekvence klesne k Nyquistově frekvenci ($fs = 2 \cdot f_p$) a pod ní, podává metoda s NDFT zřetelně lepší výsledky než metoda využívající (klasickou) DFT, viz obrázek 3.

8 Cíle práce

Cílem práce je nalezení metody pro měření vzájemného posuvu neperiodických analogových signálů. Dosavadní výzkumy v této oblasti ukázaly, že metody, které dávají nejlepší výsledky, jsou výpočetně velmi náročné. V reálných aplikacích musí být měření prováděno velkou rychlostí (řádově stovky měření za sekundu). Proto je dalším cílem práce návrh postupů pro implementaci zvolené metody do hradlových polí nebo signálových procesorů.



Obrázek 2: Porovnání metod.
$f_s = 10 \cdot f_p$, $dt = (100 \cdot f_p)^{-1}$, $SNR = 0\,dB$, $f_p = 1\,MHz$, $N = 1000$, $M = 10\,000$
Tečkovaná čára . . . metoda polynomial fit, čárkovaná čára . . . metoda s klasickou (ekvidistantní) Fourierovo transformací, plná čára . . . metoda s neekvidistantní Fourierovo transformací. Korelační metoda má příliš velkou chybu.
Výpočet proveden opakovaně:
"o". . . střední chyba, "+". . . maximální chyba.



Obrázek 3: Absolutní chyba metody s NDFT.
$f_s = 0.2 \cdot f_p$, $dt = (100 \cdot f_p)^{-1}$, $SNR = 0\,dB$, $f_p = 1\,MHz$, $N = 20$, $M = 10\,000$
Graf pouze pro metodu s NDFT, ostatní metody selhaly pro nedostatek vzorků..
Výpočet proveden opakovaně:
"o". . . střední chyba, "+". . . maximální chyba.
Poznámka: Špičky v maximální chybě se mohou vyskytnout díky dělení malým číslem v rovnici (1). Řešením je statistické zpracování výsledků nebo apriori hrubá znalost znalost frekvence f_p.

9 Další postup práce

Další postup práce bude následující:

- Dosud provedené práce umožnily stanovit dvě metody, které jsou vhodné pro měření vzájemného posuvu rychlých analogových signálů. Cílem dalšího výzkumu bude porovnání obou metod s ohledem na výpočetní náročnost a na možnosti jejich implementace.
- Návrh postupů pro implementaci metody do hradlového pole nebo signálového procesoru.
- Implementace metody do hradlového pole a signálového procesoru.
- Experimentální ověření metody a porovnání vlastností obou implementací.

10 Závěr

V článku je odvozena metoda výpočtu vzájemného posuvu signálů s použitím neekvidistantního vzorkování a neekvidistantní Fourierovy transformace (NDFT) a popsány její výhody pro snížení potřebného počtu vzorků. Numerické experimenty potvrdily, že metoda využívající neekvidistantní Fourierovu transformaci poskytuje nejméně stejně dobré výsledky jako jiné metody. Pro určité kombinace odtupu signálu od šumu a vzorkovací frekvence metoda poskytuje lepší výsledky, než ostatní metody.

Poděkování

Tato práce byla podpořena interním grantem ZČU SGS–2013–029 Advanced Computer and Information Systems a grantem European Regional Development Fund (ERDF) — project NTIS (New Technologies for Information Society), European Centre of Excellence, CZ.1.05/1.1.00/0.2.0090.

Reference

[1] Xiangwei Zhu *et al.*, "A High-Precision Time Interval Measurement Method Using Phase-Estimation Algorithm," *Instrumentation and Measurement, IEEE Transactions on* , vol.57, no.11, pp.2670,2676, Nov. 2008. doi: 10.1109 / TIM.2008.925025

[2] P. Pánek, "Time-Interval Measurement Based on SAW Filter Excitation," *Instrumentation and Measurement, IEEE Transactions on* , vol.57, no.11, pp.2582,2588, Nov. 2008 doi: 10.1109 / TIM.2008.925014

[3] Ming-Chien Tsai and Ching-Hwa Cheng, "A fullsynthesizable high-precision built-in delay time measurement circuit" *Design Automation Conference, 2009. ASP-DAC 2009. Asia and South Pacific*, vol., no., pp.123,124, 19-22 Jan. 2009 doi: 10.1109 / ASPDAC.2009.4796463

[4] Xiaoming Lai and H. Torp, "Interpolation methods for time-delay estimation using cross-correlation method for blood velocity measurement," *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on* , vol.46, no.2, pp.277,290, March 1999 doi: 10.1109 / 58.753016

[5] K. Dudáček jr. *et al.*, V. "Měření vzájemného posuvu rychlých neperiodických signálů," *Elektrorevue*, submitted for publication.

[6] Jian-Jiun Ding, "Non-uniform Sampling" [Online]. Available: http://djj.ee.ntu.edu.tw/-Nonuniform_Sampling.docx [2014, Apr. 3].

[7] J. Reif and Z. Kobeda: *Úvod do pravděpodobnosti a spolehlivosti*, 2nd ed., Pilsen, Czech rep.: ZČU, 2004.

[8] Jae-Jeong Hwang *et al.*, "Non-uniform DFT based on nonequispaced sampling," in *Proc. 5th WSEAS Int. Conf on Signal, Speech and Image Processing*, Corfu, Greece, 2005, pp. 11–16.

[9] A. Dutt, "Fast Fourier Transform for Nonequispaced Data," Ph.D. dissertation, Yale University, Connecticut, 1993.

[10] A. Dutt and V. Rokhlin, "Fast Fourier Transform for Nonequispaced Data II," Yale University, Connecticut, Research Report 980, 1993.

[11] D. Potts, (2014, Jan. 17). "NFFT" [Online]. Available: http://www-user.tu-chemnitz.de/∼potts/nfft/ [2014, Apr. 3]

GNU/Linux and Reconfigurable Multiprocessor FPGA Platform

Ing. Petr Cvek
Technical Cybernetics, 2-nd class, full-time study
Supervisor: Prof. Ing. Ondřej Novák, CSc.

The Institute of Information Technology and Electronics, FMIIS
Technická univerzita v Liberci, Studentská 1402/2, 461 17 Liberec 1

petr.cvek@tul.cz

Abstract. The article presents design of MPSoC (MultiProcessor System on Chip) with DPR (Dynamic Partial Reconfiguration) support for any processor. MPSoC is based on the softcore processors and is controlled by a modified GNU/Linux operating system. Modifications of operating system allow to develop reconfiguration triggered by change in the type of performed task. DPR requires hardware support, which has been achieved by modification of the standard interface (AXI). The article presents implemented system and measured benchmarks.

Keywords. FPGA, multiprocessor system, reconfiguration, GNU/Linux

1 Introduction

With ending era of systems with single processor we can observe expanding of multiprocessor systems to more and more sectors of embedded computing. FPGA (Field Programmable Gate Array) is one of the sectors where multiprocessor systems already exist. This article describes systems, where multiprocessor is implemented as softcore. Softcore implementation gives us an unique feature of reconfiguration, where a part of processor can be replaced by software intervention. When reconfiguration occurs during the operation, we call it DPR (Dynamic Partial Reconfiguration) [18].

There are already projects which use hardware reconfiguration, multiple processors on an FPGA or operating system like GNU/Linux on softcore processor, but combination of these features is still little explored. This article presents methodology for a design of GNU/Linux based multiprocessor system. Final MPSoC can be used for research in DPR area.

1.1 Related works

Some of multiprocessor designs using FPGA are:

RAMPSoC [6] which uses single processor for controlling processing elements. This system is controlled by special operating system CAP-OS. RAMPSoC supports only single tasking and requires special developing toolchain.

Heracles [9] is a system without OS. Every hardware application must be compiled into Verilog description.

The Raptor [13] system is implemented on a board with multiple FPGAs and it is controlled by PowerPC and GNU/Linux. Hardware modules are reconfigured by DPR.

Similar to the Raptor is the Borph [14]. This design uses a concept of hardware processes in GNU/Linux OS.

RAMP Blue [10] is highly scalable Microblaze system consisting of multiple FPGA boards and multiple processor cores per FPGA. RAMP Blue has got uCLinux kernel instance on every processor. Each kernel communicates by message passing through interface network. Any application must be designed for executing on GNU/Linux cluster. RAMP Blue does not mention any type of reconfiguration.

System described in [8] utilizes single Microblaze processor, which controls PicoBlaze processors. Every PicoBlaze processor is a controller for floating point units. System can perform only one task, which must be specially developed for PicoBlaze architecture.

PolyBlaze [12] is nearly the same as system described in this article. It supports multiple processors (up to eight) and it uses symmetrical multiprocessing with Linux kernel. PolyBlaze does not have data cache and article does not describe any reconfiguration methods.

1.2 Objectives of dissertation

The goal of the research is to develop methodology for creation of the heterogenous systems with ability to execute general computer tasks on general or specialized hardware. Task transmission between general and specialized hardware is intended to be the method for optimizing the running job and will be conducted online by use of the dynamic partial reconfiguration. Main goal of the dissertation can be divided into some subobjectives. These are:

- Research of the existing projects of system on the chip on the FPGA
- Design of the reconfigurable platform for the research
- Developing theory for the hardware reconfiguration task scheduler of the operating system
- Exploration of the possibilities for increasing reliability, power efficiency and computing power

2 Multiprocessor Microblaze Linux system

2.1 Hardware

Upgrading from single-procesor to multiprocessor SoC requires modification of the existing parts of system and addition of new functions. These functions are:

- Communication between processors (task rescheduling messages, function calls, etc...)
- Interrupt controller with adjustable routing to any processor in the system
- Protection against data hazards during shared memory access (read-modify-write from two or more processors)
- Support for gated clocks on clock nets routed into partitions selected for reconfiguration

A further description of these essential requirements is in [4]. Speed of application execution can be increased by integrating cache with coherent data. Alternatively we can connect groups of processors to non-uniform memory.

Application acceleration can be achieved by addition of co-processors to Microblaze or by replacement of unused general processor by specialized one. There must always be some general processor left to execute operating system.

If we want replace accelerators, we need to ensure, that data from replaced hardware will be saved and reconfiguration will not affects other parts of the system. This is ensured by integration of accelerators into private partition and by clock gating support.

2.2 Scheduler

In general, the scheduler is responsible for effective task switching. For multiprocessor SoC this responsibility extends to effective switching between processors. Default Linux kernel scheduler is called CFS (Completely Fair Scheduler). CFS is implemented using red-black tree for storing tasks, which are waiting for execution. Choosing task for execution can be done in constant time complexity and task reinserting has logarithmic complexity.

Scheduler, which is designed to be compatible with reconfiguration, has some practical limitations. One of them is speed of ICAP (Internal Configuration Access Port) programming interface. Maximum recommended speed of ICAP (in used FPGA models) is 100 MHz and word length is 32 bits. Bitstream of one Microblaze processor is over 500 kB, so reconfiguration of one Microblaze cannot take less than 1.25 milliseconds. Additional time will be taken by initialization. Some Linux kernel configuration can change tasks faster than that. We will develop scheduler in future research.

3 Experimental system

Research of GNU/Linux MPSoC on an FPGA requires implementation of experimental system, which was implemented on development board ML605 (XC6VLX240T) and later on KC705 (XC7K325T). We can see its block diagram on Figure 1. The system was designed for maximal compatibility with Xilinx development tools. We have chosen Microblaze [17] processor and AXI (Advanced eXtensible Interface) [3, 15] as main interface. With Kintex FPGA we can implement system with at least four processors running on 150 MHz. The maximum frequency is limited by number of signals between processor and memory interfaces, for example maximum frequency decreased to 100 MHz after addition the cache interface to all processors. The experimental system has an additional limitation in number of options where the logic can be placed. This limitation is required by FPGA reconfiguration described in [5]. The reconfigurable logic has to be placed to particular positions into a reserved region. It requires possibility of gating clock signals in order to enable stopping processors during reconfiguration. This additional logic causes more latency, therefore lower frequency.

3.1 Main hardware

We modified the default interrupt controller [16] to ensure basic requirements of multiprocessor systems. Internal functionality and register interface of the interrupt controller for each processor was simply duplicated, so any interrupt request can be directed to any desired processor. In order to support partition clock gating, reset control and inter-processor communication, the original register set of the interrupt controller was expanded. Finally, independent timer has been added for each processor in the system.

Operating systems of symmetric multiprocessor system often use the concept of “per CPU” variables. Per CPU variable represents a redirection mechanism, which is transparent for the user program and which guarantees, that every CPU has its own memory location for storing value of one per CPU variable. Any CPU which accesses the per CPU variable will be redirected to its own value (unless explicitly requested otherwise). With this mechanism, same machine code can achieve different behavior on any processor, where the machine code is executed. We can consider a register (as in processor architecture) as the simplest per CPU variable. Per CPU variables are used in many places in the operating system and their number is much greater than number of the registers. Their number varies for different configurations. Therefore per CPU variables are located in the main memory.

Accessing per CPU variables can be emulated by a single value, which contains unique identification number of each processor. ID number can be used for addressing variable, structure or array in the shared memory. A problem can arise for low level functions like interrupt handling or task switching. These functions need to distinguish between processors, but some CPU architectures lack available register

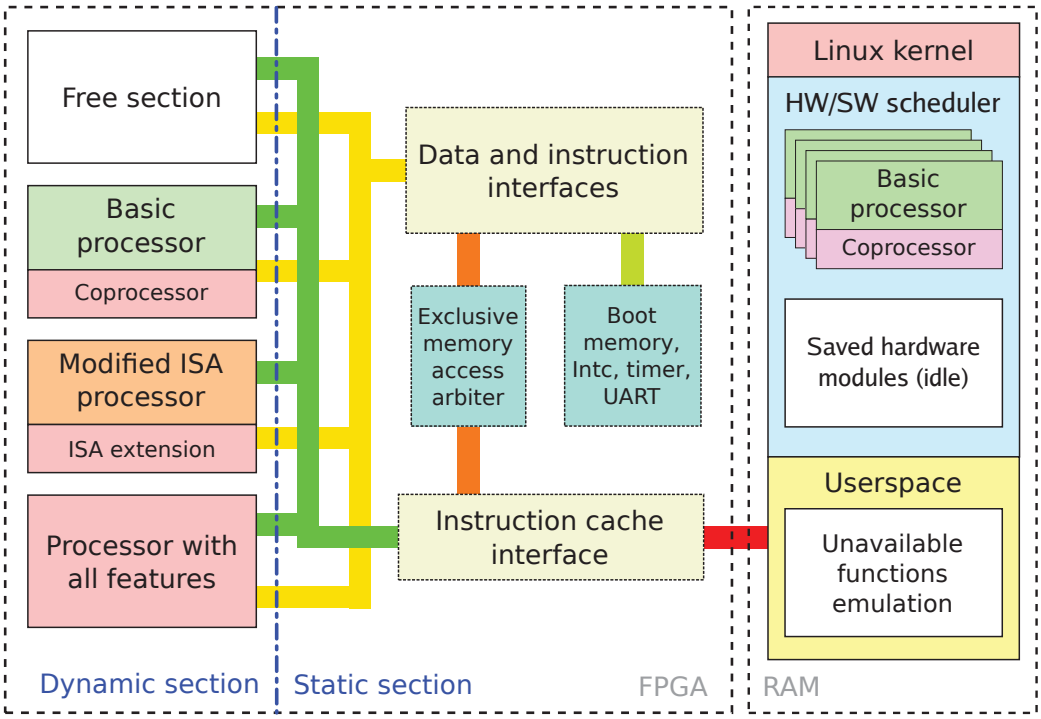


Figure 1: Block diagram of Microblaze MPSoC

or memory (for example: stack) for storing unique identification. Microblaze used in our MPSoC is this case, too. Lack of free registers requires addition of at least two registers with direct access from operating code. One register contains unique ID and the second serves as a temporary store of register content in the interrupt handlers. These registers would not be necessary if the Microblaze architecture has reserved general purpose registers in ABI (Application Binary Interface) or some userspace auxiliary registers in SPR (Special Purpose Register) set. The registers were implemented as two AXI Stream slaves. Instruction for accessing any data through AXI Stream can directly access data of the register, so it does not overwrite any data stored in any register.

3.2 Operating system

The experimental system uses a modified Linux kernel for Microblaze architecture. The original kernel provides no support for Microblaze multiprocessor system. We had to re-implement parts of the kernel like processor interrupt handling routines, IPI, timer for events, processor clocking control, hardware semaphores and per CPU access methods. The experimental system supports very large number of processors (limited by kernel functions shared with other architectures), so the main limitation is caused by underlying hardware (JTAG debug module, AXI interface connections, size of FPGA, ...) only. We can enable or disable any processor during runtime operation (so called processor hotplug) with the help of the clocking control. Every task on disabled processor is automatically migrated to another processor by the task scheduler. Experimental system supports cgroups [1], so we can attach any group of processes to any processor. These properties directly lead to implementation of reconfiguration support for the task scheduler.

Table 1: Delay loop calibration on MPSoC, computed by calibrate_delay_converge()

Configuration	BogoMIPS	loops_per_jiffy
50 MHz, no cache	0.32	1624
50 MHz, 8 kiB L1 i-cache	22.01	110080
100 MHz, 8 kiB L1 i-cache	46.08	230400
150 MHz, 8 kiB L1 i-cache, 256 kiB shared L2 cache	70.86	354304

Table 2: Unixbench results

Number of loops per 1 second	Microblaze 150 MHz	PXA272	Intel Core i5-3570
dhry2	$1.3 \cdot 10^4$	$7.5 \cdot 10^5$	$3.5 \cdot 10^7$
dhry2reg	$1.3 \cdot 10^4$	$7.7 \cdot 10^5$	$3.5 \cdot 10^7$
hanoi	$1.7 \cdot 10^2$	$6.2 \cdot 10^3$	$2.7 \cdot 10^5$
spawn	$3.8 \cdot 10^1$	$1.6 \cdot 10^2$	$2.0 \cdot 10^4$
pipe	$1.4 \cdot 10^3$	$8.5 \cdot 10^4$	$2.4 \cdot 10^6$
context1	$4.9 \cdot 10^2$	$1.7 \cdot 10^3$	$5.1 \cdot 10^5$
syscall	$3.8 \cdot 10^3$	$2.6 \cdot 10^5$	$4.1 \cdot 10^6$

4 Experiments and results

4.1 Impact of the i-cache size

We measured an impact of the cache size of multiprocessor benchmarks. Table 1 illustrates values of internal timing variable `loops_per_jiffy` from Linux kernel. This variable is used to estimate system speed during kernel boot. Its value represents the number of delay loops executed between two local timer events. First line shows the system without any cache. As we can see on the second line, when we have configured a system to include small instruction cache, we have obtained huge improvement. With configured L2 cache (on last line), there is not any speed improvement, because delay loop procedure code fits in the L1 i-cache and it does not periodically access same data from the memory (data are stored in the registers).

Delay loops measuring is useful only to illustrate the speed of instruction execution during the early start of the system (only single processor is running), therefore the system has been further tested by UnixBench 5.1.3 [2] and lat_mem_rd from the LMbench 3 package [11].

4.2 Unixbench tests

UnixBench is a set of some basic programs, which measure number of completed test executions during the predefined time interval. Executed tests of UnixBench are:

- dhry2reg - Implementation of Dhrystone using registers
- dhry2 - Dhrystone implementation (Microblaze architecture can access a variable in memory only through registers)
- hanoi - Tower of Hanoi solving algorithm for ten disks
- spawn - Benchmark for measuring time of starting child process
- pipe - Reading and writing 512 bytes into pipe in single process (no context switch)

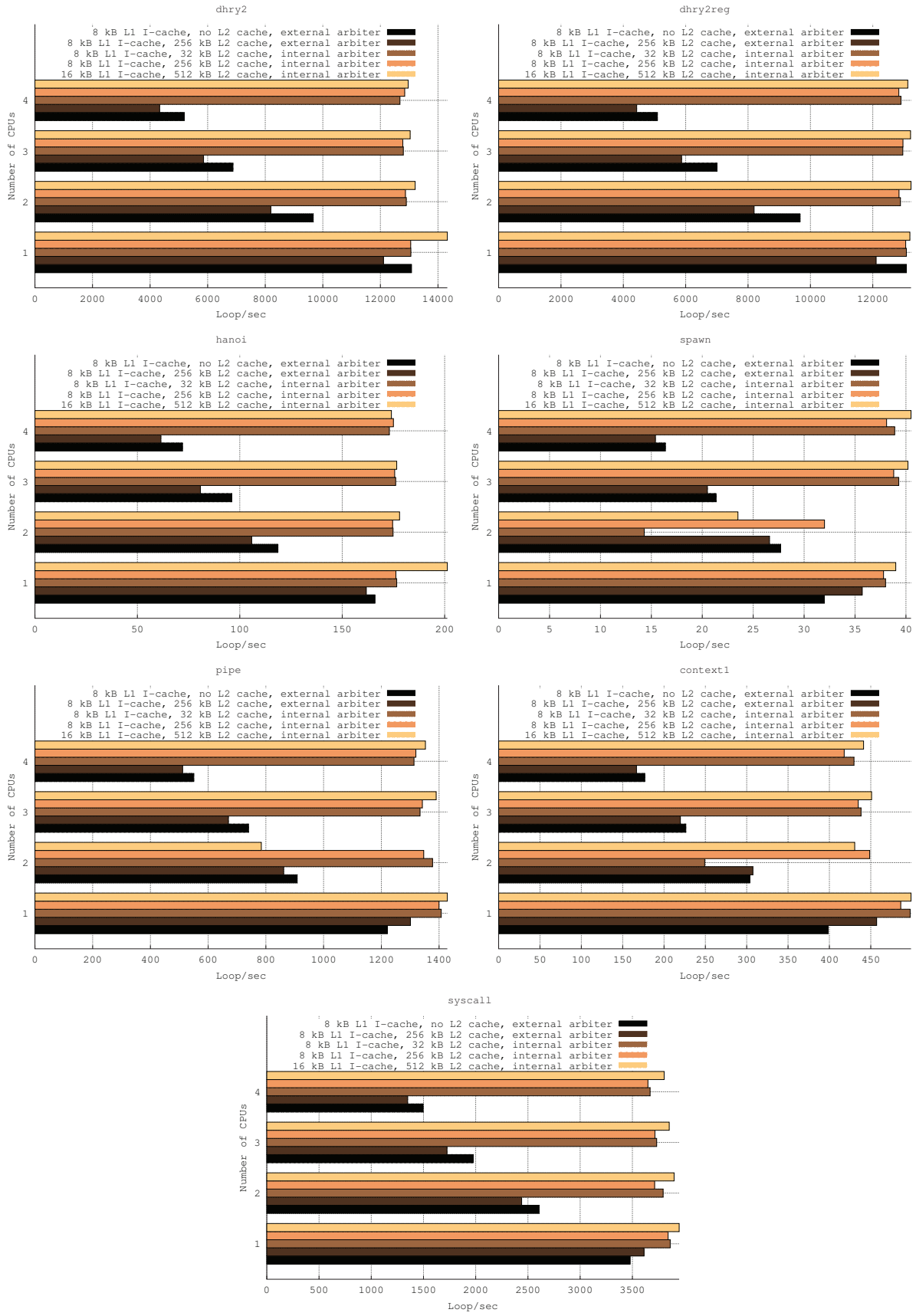


Figure 2: Benchmarks for different number of activated CPUs (Dhrystone, Registered Dhrystone, Hanoi tower solver, Process spawning, Pipe communication, Context switching, System call)

- context1 - Measuring the speed of switching between two processes by communication through a pipe
- syscall - Kernel calling tests (duplicate file descriptor, close, get process ID, get user ID, setting default file mode)

Results of tests using two or more processes can differ on single and multi processor system. Figure 2 shows results of these benchmarks on five different hardware configurations. Each hardware configuration has been tested on 1 to 4 processors (disabled/enabled at runtime).

In Table 2 we have compared last version of the Microblaze multiprocessor system:

- 4x Microblaze at 150 MHz, L1 I-cache 8 kB, 16 B cacheline, L1 D-cache disabled
- 1 GB DDR3 at 800 MHz
- Interface to DDR3 128 bit at 200 MHz
- system_cache (L2) 256 kB, 64 B cacheline, two associative sets

with Intel ARM:

- PXA272 at 416 MHz, L1 I-cache 32 kB, L1 D-cache 32 kB
- 64 MB SDR 32bit at 104 MHz

and with Intel i5:

- Intel Core i5-3570 at 3.40 GHz, L1 I-cache 4x 32 kB, L1 D-cache 4x 32 kB, L2 cache 4x 256 kB, L3 cache 6 MB
- 16 GB DDR3 at 800 MHz

Only one processor was enabled on both Microblaze and i5 system.

Table 2 shows that Microblaze is around x10 times slower than PXA272 and about x1000 times slower than i5.

4.3 LMbench memory latency test

The cache benchmark was tested by lat_mem_rd program (example in IBM article [7]), which can be used for estimation of cache hierarchy and size. This is achieved by measuring a latency for linear reading from an array. This reading can be done in different array sizes and with different stride lengths. Each configuration is measured multiple times in order to minimize interferences from OS. When the array size exceeds the cache size, next request will result in the cache miss, which will fetch data from the next level of cache (or from the main memory). This causes a rise of the access latency, which can be used for estimation of the cache size. If the stride length is close to size of one page and the array size is big enough we can observe the TLB (Translation Lookaside Buffer) miss latency.

The memory latency of the Microblaze system will differ from the IBM POWER system. The Microblaze system has not an L1 d-cache and its architecture is simpler. When the array size exceeds L2 cache size, we get three cases. First one is situation, where the stride length equals the cacheline size. Any access results in the cache miss. Testing the configuration with stride length less than the cacheline size causes the first access to cache miss and the cache controller transmits the whole cacheline. If the next access targets the same cacheline, it results in the cache hit. The latency between L2 cache and the memory is divided by the number of accesses into the same cacheline. The third case has stride length

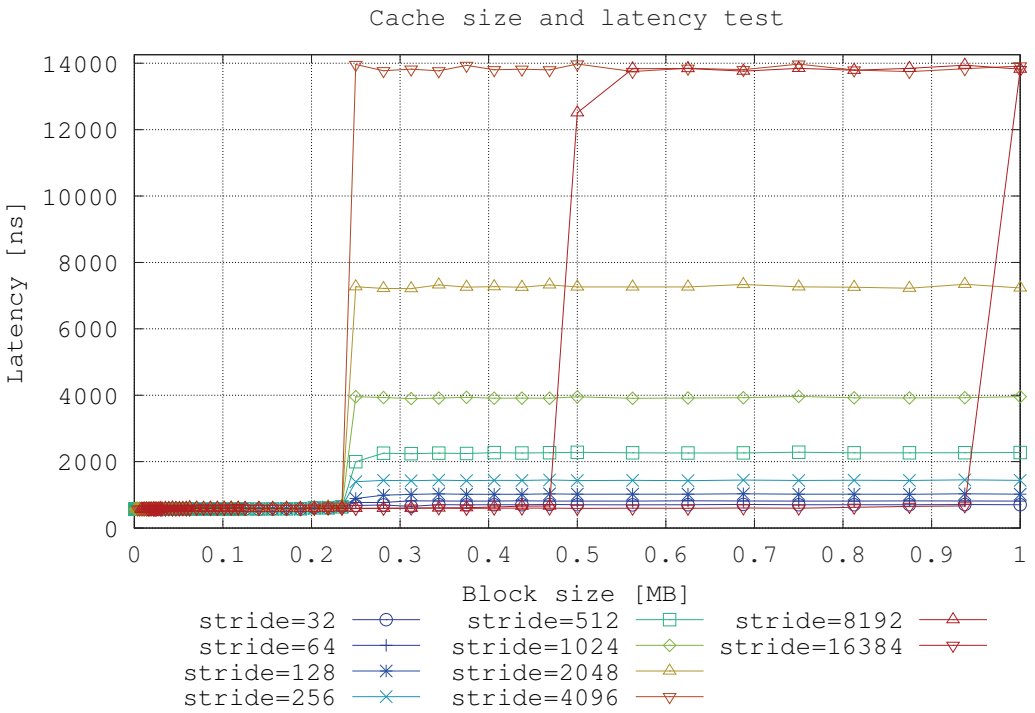


Figure 3: Memory access latency (lat_mem_rd)

greater than the cacheline size. Any access causes the cache miss, but some cachelines are skipped, therefore the average latency is similar to the latency with stride length that equals the cacheline size.

Virtual memory is another property which can greatly affect the latency. The Linux kernel is using pages with page size of 4096 bytes. Every page requires a TLB entry, which is stored in the TLB cache with the size of 64 entries. Access to any page, which is not inside the TLB cache, results in the TLB miss and the new entry is located (by OS exception routine) and written into TLB cache. These properties mean that Microblaze, which is accessing more than 256 kB (64 pages) has an increased latency due to the software handling for the TLB miss.

Tests used 1 MB long block and various stride length (from less than cache line to more than page size). Configuration of the Microblaze system was: L1 i-cache only, 512 kB L2 cache and 150 MHz system clock frequency. As we can see in Figure 3 we measured latency rise around 256 kB, which is maximal size of the memory, which can be serviced without the TLB miss. We are not having any significant change in the L2 cache around 512 kB (size of L2 cache), so it seems, that L2 cache in this configuration is not required. Stride lengths less than page size have lower latency, because there are multiple TLB hits in one page and so access latency is averaged. For stride lengths greater than page size, the access latency is no longer rising, because it still one TLB miss per access. For these stride lengths, the latency rises at bigger block size, because some pages are simply skipped.

5 Conclusions

The first configuration of the system was from early version of the experimental system. It lacks any data caches and hardware for protection against the data hazards. The exclusivity access arbiter can accept a new transaction after completing the first one only. This serialization has a big impact on the performance of the memory interface and consequently the processors in all experiments. Shared bandwidth of the interface basically means that the speed decreases when a new processor is added into the system. This

attribute was rendered as unacceptable and this system was used to modify the Linux kernel. Next stage of the development was addition of the L2 cache. The system was equipped with our experimental exclusivity access arbiter. This configuration caused latency, because the L2 cache required another interface interconnect. We can see decrease of the performance from the measured data in Unixbench tests in the Section 4.2. The critical part is the interface latency and not the cache. With new version [19] of the L2 cache, we used integrated exclusivity arbiter, which supports concurrent transactions. This allowed us to drop experimental exclusivity access arbiter and redundant AXI interconnect. Results show great improvement on three different L2 sizes. We can see only minor slowdown for each new processor. Little slowdown still exists because interface is blocked for the transmission request. Results also confirm that impact of the cache size is minor. This would be different if we have used the L1 data cache, but to this time we have not a usable solution. Actual L2 cache [19] supports the L1 data cache interface (cache coherent AXI), but we have found, that it is not compatible with the reconfigurable multiprocessor design.

Tests "spawn", "pipe" and "context1" have highly fluctuating results when two processors were activated (other two were disabled). These tests use two processes or are focused on the data transmissions.

Big ratio to the speed of Intel i5 is caused by lack of branch predictor in the implemented design, lack of the L1 data cache and high latency of the interface interconnect. The latency can exceed 20 clock cycles for a single transmission. On the other side, modern hardcore CPU uses L1 cache with latencies of one cycle. Intel i5 is superscalar processor with very long pipeline and its clock frequency about 22 times higher than Microblaze clock frequency. The Microblaze system with L1 data cache and with enabled branch prediction could achieve similar MIPS/MHz ratio.

The memory latency test shows weakness of Microblaze architecture: MMU exception handling. Any memory page, which is not located inside TLB cache creates exception and must be handled by the kernel. This takes lots of the instruction cycles. Effect of TLB miss could be reduced by adding more TLB cache entries or by implementing a hardware TLB manager.

6 Future work

In future work we will concentrate on the scheduler design for reconfiguring partitions with processors. Another possible work which can be done is a raise of computational speed of the system. Actual system lacks L1 data cache and despite the fact that speed of the DDR3 memory greatly exceeds speed of the implemented Microblaze, any read-modify-write function deals with the interface latency.

Outlined system design can allow its use as a basis for system, which can replace its hardware on application demand. This can be useful for any general purpose computing.

We hope that this work will allow us better understand multiprocessor and reconfiguration designs.

Acknowledgment

The research is supported by the Student Grant Scheme (SGS) at the Technical University of Liberec, and co-financed by the Czech Ministry of Education, Youth and Sport. The work is also supported by the COST LD-13019 program and the COST Action IC1103-Median program.

References

[1] Linux kernel documentation. 2014.
URL <https://www.kernel.org/doc/Documentation/cgroups/cgroups.txt>

[2] References Unixbench. 2014.
URL <http://code.google.com/p/byte-unixbench/>

[3] ARM: AMBA AXI and ACE Protocol Specification. 2011.
URL <http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.ihi0022e/index.html>

[4] Cvek, P.; Drahonovsky, T.; Rozkovec, M.: GNU/Linux and reconfigurable multiprocessor FPGA platform. In *Electronics, Control, Measurement, Signals and their application to Mechatronics (ECMSM), 2013 IEEE 11th International Workshop of*, June 2013, s. 1–5, doi:10.1109/ECMSM.2013.6648932.

[5] Drahonovsky, T.; Rozkovec, M.; Novak, O.: Relocation of reconfigurable modules on Xilinx FPGA. In *Design and Diagnostics of Electronic Circuits Systems (DDECS), 2013 IEEE 16th International Symposium on*, 2013, s. 175–180, doi:10.1109/DDECS.2013.6549812.

[6] Gohringer, D.; Becker, J.: High performance reconfigurable multi-processor-based computing on FPGAs. In *Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), 2010 IEEE International Symposium on*, 2010, s. 1 –4, doi:10.1109/IPDPSW.2010.5470800.

[7] Hopper, J.: Untangling memory access measurements. 2013.
URL <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/W51a7ffcf4dfd_4b40_9d82_446ebc23c550/page/Untangling%20memory%20access%20measurements%20-%20memory%20latency>

[8] Kadlec, J.; Bartosinski, R.; Danek, M.: Accelerating Microblaze Floating Point Operations. In *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, 2007, s. 621 –624, doi:10.1109/FPL.2007.4380731.

[9] Kinsy, M.; Pellauer, M.; Devadas, S.: Heracles: Fully Synthesizable Parameterized MIPS-Based Multicore System. In *Field Programmable Logic and Applications (FPL), 2011 International Conference on*, 2011, s. 356 –362, doi:10.1109/FPL.2011.70.

[10] Krasnov, A.; Schultz, A.; Wawrzynek, J.; aj.: RAMP Blue: A Message-Passing Manycore System in FPGAs. In *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, 2007, s. 54 –61, doi:10.1109/FPL.2007.4380625.

[11] Larry McVoy, C., L.; Staelin: LMBench. 2014.
URL <http://www.bitmover.com/lmbench/>

[12] Matthews, E.; Shannon, L.; Fedorova, A.: Polyblaze: From one to many bringing the microblaze into the multicore era with Linux SMP support. In *Field Programmable Logic and Applications (FPL), 2012 22nd International Conference on*, 2012, s. 224–230, doi:10.1109/FPL.2012.6339185.

[13] Rana, V.; Santambrogio, M.; Sciuto, D.; aj.: Partial Dynamic Reconfiguration in a Multi-FPGA Clustered Architecture Based on Linux. In *Parallel and Distributed Processing Symposium, 2007. IPDPS 2007. IEEE International*, 2007, s. 1 –8, doi:10.1109/IPDPS.2007.370363.

[14] So, H. K.-H.; Brodersen, R.: Improving Usability of FPGA-Based Reconfigurable Computers Through Operating System Support. In *Field Programmable Logic and Applications, 2006. FPL '06. International Conference on*, 2006, s. 1 –6, doi:10.1109/FPL.2006.311236.

[15] Xilinx: AXI Reference Guide, UG761. 2012.

[16] Xilinx: LogiCORE IP AXI INTC (v1.02a). 2012.
URL <http://www.xilinx.com/support/documentation/ipembedprocess_processorinterface_axi_intc.htm>

[17] Xilinx: MicroBlaze Processor Reference Guide, UG081. 2012.

[18] Xilinx: Partial Reconfiguration User Guide, UG702. 2012.

[19] Xilinx: LogiCORE IP System Cache v2.00a, PG031. 2014.

ADAPTÁCIA ALGORITMU OPRAVY PAMÄTÍ RAM NA BLOKOVÚ ARCHITEKTÚRU

Štefan Krištofík
Aplikovaná informatika, 3.ročník, denná forma
Školiteľka: doc. RNDr. Elena Gramatová, PhD.

Fakulta informatiky a informačných technológií, STU Bratislava
Ilkovičova 2, 842 16 Bratislava 4

stefan.kristofik@stuba.sk

Abstrakt. Vstavaná samočinná oprava vnorených pamätí RAM sa používa na zvýšenie výťažnosti výroby systémov na čipe. V článku je opísaný návrh nového algoritmu opravy pamätí, ktorý je rozšírením, modifikáciou a adaptáciou existujúceho algoritmu určeného na opravu pamätí s tradičnou architektúrou, na blokovú architektúru. Úspešnosť opravy nového algoritmu je experimentálne overená a porovnaná s existujúcim.

Kľúčové slová. Vnorená pamäť, výťažnosť, systém na čipe, algoritmus opravy pamäte, úspešnosť opravy.

1 Úvod

Pamäte tvoria v súčasnosti asi 2/3 plochy systémov na čipe (SoC – *System-on-a-chip*) [1] a ich spoľahlivosť má najväčší vplyv na výťažnosť výroby SoC. Testovaniu a oprave pamätí sa preto v súčasnosti venuje pozornosť a realizuje sa priamo na čipe ako vstavaná samočinná oprava (BISR – *Built-in Self-repair*). Princípom BISR je pridanie záložných elementov ku pamäti a nahradenie poruchových buniek záložnými. O spôsobe nahradenia (riešení opravy) rozhoduje algoritmus opravy pamäte (*redundancy/repair analysis algorithm*). Z literatúry sú známe mnohé algoritmy opravy, zväčša určené na opravu pamätí s tradičnou architektúrou [2], ale niektoré aj na opravu pamätí s blokovou architektúrou [3]. Pamäte s touto špeciálnou architektúrou sú fyzicky rozdelené na samostatne adresovateľné bloky (kvadranty) rozdelením bitového a slovného vodiča. Zálohy, tiež rozdelené na bloky, sú využité efektívnejšie ako v tradičnej architektúre, ale za cenu mierneho navýšenia hardvéru potrebného na implementáciu oddeľovacích tranzistorov a logických členov.

Doposiaľ existujúce algoritmy opravy pre blokovú architektúru boli adaptáciou len jednoduchých algoritmov určených na opravu pamätí s tradičnou architektúrou. Motiváciou pre návrh nového algoritmu opravy pre blokovú architektúru bolo overenie predpokladu, že použitie zložitejšieho algoritmu v blokovej architektúre pamätí bude viesť k vyššej úspešnosti opravy (definovaná v časti 4) ako u doposiaľ známych prístupov. Ďalšou motiváciou bolo to, že doposiaľ existujúce algoritmy pre blokové pamäte nie sú schopné garantovať nájdenie optimálneho riešenia opravy pamäte. Optimálne riešenie opravy je také, ktoré využíva na opravu pamäte najmenší možný počet záloh [2].

Článok je pokračovaním [4], kde boli uvedené základné koncepty návrhu nového algoritmu opravy pamätí MSFCC (*Modified SFCC – Modified selected fail count comparison*). V časti 2 sú opísané základné vlastnosti MSFCC. Časť 3 sa zameriava na uvedenie viacerých navrhnutých rozšírení a modifikácií pôvodného algoritmu SFCC, z ktorého sa pri návrhu MSFCC vychádzalo, a ktorých výsledkom je úspešná adaptácia na blokovú architektúru. V časti 4 je uvedené experimentálne overenie úspešnosti opravy MSFCC a porovnanie s existujúcim riešením pre blokovú architektúru.

Časť 5 je zhrnutím dosiahnutých výsledkov. Podrobnejšie informácie je možné nájsť v predbežnej verzii dizertačnej práce [5].

2 Opis algoritmu

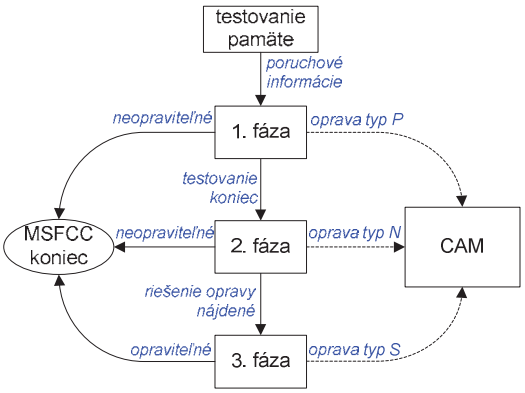
Nový algoritmus opravy MSFCC je založený na princípoch algoritmov SFCC [2] a MESP (*Modified essential spare pivoting*) [3]. Kombinuje dosahovanú vysokú úspešnosť opravy SFCC s výhodami, ktoré poskytuje použitie blokovej architektúry pamätí a globálnych záloh podľa princípu MESP. MSFCC je hybridný 2-D [2] algoritmus opravy určený na opravu výlučne bitovo orientovaných pamätí RAM. Jedným zo základných princípov je rozdelenie porúch na 3 typy:

- **Typ P: povinne opravované poruchy r/s (riadkovým/stĺpcovým) blokom.** Vo svojom r/s bloku majú spolu viac porúch, ako je k dispozícii záložných s/r blokov. Na ich opravu je preto nutné použiť záložný r/s blok.
- **Typ N: násobné poruchy v r/s bloku.** Vo svojom r/s bloku majú viac ako 1 poruchu, ale menej, nanajvýš rovnako porúch ako je k dispozícii záložných s/r blokov. Na ich opravu je možné použiť 1 r/s blok alebo viacero s/r blokov. O spôsobe opravy rozhoduje MSFCC.
- **Typ S: samostatné poruchy.** Vo svojom r/s bloku sa nachádzajú ako jediné. Na ich opravu je možné použiť akýkoľvek typ zálohy.

MSFCC pracuje v troch fázach:

- **1. fáza: Zber porúch a oprava porúch typu P.** Zber porúch prebieha počas testovania pamäte a informácie o každej detekovanej poruche sa ihneď po jej detekcii ukladajú do malých pomocných pamätí adresovaných obsahom (CAM – *content-addressable memory*). Algoritmus je v tejto fáze schopný rozlíšiť poruchy typu P od ostatných, a v prípade nájdenia takýchto porúch ihneď pridelí príslušnú zálohu na ich opravu.
- **2. fáza: Prideľovanie záloh a oprava porúch typu N.** Prideľovanie prebieha až po ukončení testovania pamäte, kedy sú známe poruchové informácie o všetkých poruchách v pamäti. Algoritmus pomocou informácií uložených do CAM v 1. fáze vytvorí zoznam všetkých poruchových r/s blokov v pamäti obsahujúcich poruchy typu N. Tento zoznam uloží do malého pomocného buffera (realizovaného tiež ako pamäť typu CAM). Následne na základe analýzy obsahu buffera vyberie vhodné riešenie opravy porúch typu N [5].
- **3. fáza: Oprava zvyšných porúch a porúch typu S.** Oprava prebieha po skončení 2. fázy. Algoritmus pomocou informácií v CAM identifikuje špeciálne zvyšné neopravené poruchy z 2. fázy (ktoré v tejto fáze už spĺňajú vlastnosti porúch typu S [5]), a tiež poruchy typu S. Na opravu každej z takýchto porúch sa pridelí jedna záloha, napríklad náhodným spôsobom.

Priebeh MSFCC je ilustrovaný na obr. 1. Algoritmus je spustený spolu s testovaním pamäte. V každej fáze je možná skorá detekcia neopraviteľných pamätí a predčasné ukončenie.



Obr. 1: Priebeh MSFCC.

V prípade, ak riešenie opravy pre pamäť existuje, MSFCC garantuje nájdenie optimálneho riešenia opravy. Je to zabezpečené v 2. fáze algoritmu, kedy sa po nájdení prvého riešenia opravy oprava neukončí, ale pokračuje sa ďalej, až kým sa nenájde optimálne riešenie.

3 Modifikácie a rozšírenia algoritmu

Pri podrobnej analýze pôvodného algoritmu SFCC [2] boli nájdené a identifikované niektoré jeho nedostatky, ktoré preukázateľným spôsobom [5] negatívne ovplyvňujú jeho schopnosť opravy pamätí s niektorými špecifickými rozloženiami porúch a teda jeho úspešnosť opravy. Taktiež niektoré aspekty jeho fungovania boli len spomenuté a neboli podrobne navrhnuté.

Z týchto dôvodov boli v pôvodnom algoritme vykonané viaceré modifikácie a rozšírenia tak, aby po jeho adaptácii do blokovej architektúry pamätí v algoritme MSFCC boli nájdené nedostatky odstránené. Ďalej boli navrhnuté nové presné postupy riešenia niektorých aspektov pôvodného algoritmu s ohľadom na použitie nového algoritmu v blokovej architektúre. Zoznam vykonaných modifikácií a rozšírení je uvedený v tab. 1.

Modifikácie (a) a (b) boli nevyhnutne potrebné na adaptáciu algoritmu do blokovej architektúry, kde sa bunky pamäte adresujú nielen pomocou dvojice adries, ale je potrebné identifikovať aj blokové adresy. Odhadované navýšenie počtu bitov potrebných pre použité pamäte CAM (vrátane buffera) je však len minimálne [5].

Tab. 1: Modifikácie a rozšírenia MSFCC.

Modifikácia/rozšírenie	**Riešenie v SFCC [2]**	**Riešenie v MSFCC [5]**	**Poznámky**
(a) Zmena štruktúry CAM	-	Pridané polia na uchovanie blokových adries porúch, nezvýšil sa poč. bitov CAM	Potrebné na adaptáciu do blok. architektúry
(b) Zmena štruktúry buffera	-	Pridané polia na uchovanie blokových adries porucho-vých r/s blokov, malé zvýšenie poč. bitov buffera	Potrebné na adaptáciu do blok. architektúry
(c) Spájanie porúch na konci 1. fázy	Predpokladá sa využitie, nie je uvedený postup	Navrhnutý nový postup	-
(d) Napĺňanie buffera v 2. fáze	Predpokladá sa využitie, nie je uvedený postup ani či sú ošetrené špeciálne prípady	Navrhnutý nový postup, ošetrené aj špeciálne prípady	Dôsledok: zvýšenie úspešnosti opravy
(e) Vyhodnocovanie riešení opravy v 2. fáze	Predpokladá sa využitie, nie je uvedený postup	Navrhnutý nový postup	-
(f) 3. fáza	Predpokladá sa využitie, nie je uvedený postup ani či sú ošetrené špeciálne prípady	Navrhnutý nový postup, ošetrené aj špeciálne prípady	-
(g) Zmena maximálnej kapacity jednej z pamätí CAM	- Dôsledok: zníženie úspešnosti opravy	Zvýšenie max. kapacity jeden z pamätí CAM	Dôsledok: zvýšenie úspešnosti opravy
(h) Adopcia porúch	- Dôsledok: možná nekonzistencia údajov v CAM	Navrhnutý nový postup	Ošetrenie nekonzistencie
(i) Dospenie porúch	- Dôsledok: možná nekonzistencia údajov v CAM	Navrhnutý nový postup	Ošetrenie nekonzistencie

Rozšírenia (c) až (f) predstavujú návrhy nových presných postupov pre riešenie daných aspektov pôvodného algoritmu, pričom sa berie ohľad aj na ošetrenie možných špeciálnych prípadov, ktoré môžu počas behu algoritmu nastať [5]. Tieto nové postupy sú navrhnuté až na úroveň základných operácií nad pamäťami CAM, čo by značne zjednodušilo prípadnú implementáciu algoritmu. Pozitívny vplyv modifikácie (g) na úspešnosť opravy MSFCC je možné dokázať [5]. Rozšírenia (h) a (i) odstraňujú prípady, kedy mohlo v pôvodnom algoritme dôjsť k nekonzistencii údajov uložených v pamätiach CAM a teda nebol zaručený správny priebeh opravy pamäte. Nový algoritmus MSFCC je vďaka uvedeným zmenám schopný opraviť aj pamäte so špecifickými typmi rozložení porúch, s ktorými pôvodný algoritmus SFCC nepočítal.

4 Experimenty

Jedným z parametrov používaných na vyhodnotenie efektivity algoritmov opravy je úspešnosť opravy, definovaná podľa vzťahu (1) [2]. V menovateli (1) sa do počtu všetkých pamätí počítajú aj neopraviteľné pamäte. Pri danom počte záloh je za efektívnejší považovaný ten algoritmus opravy, ktorý dosahuje vyššiu mieru úspešnosti opravy.

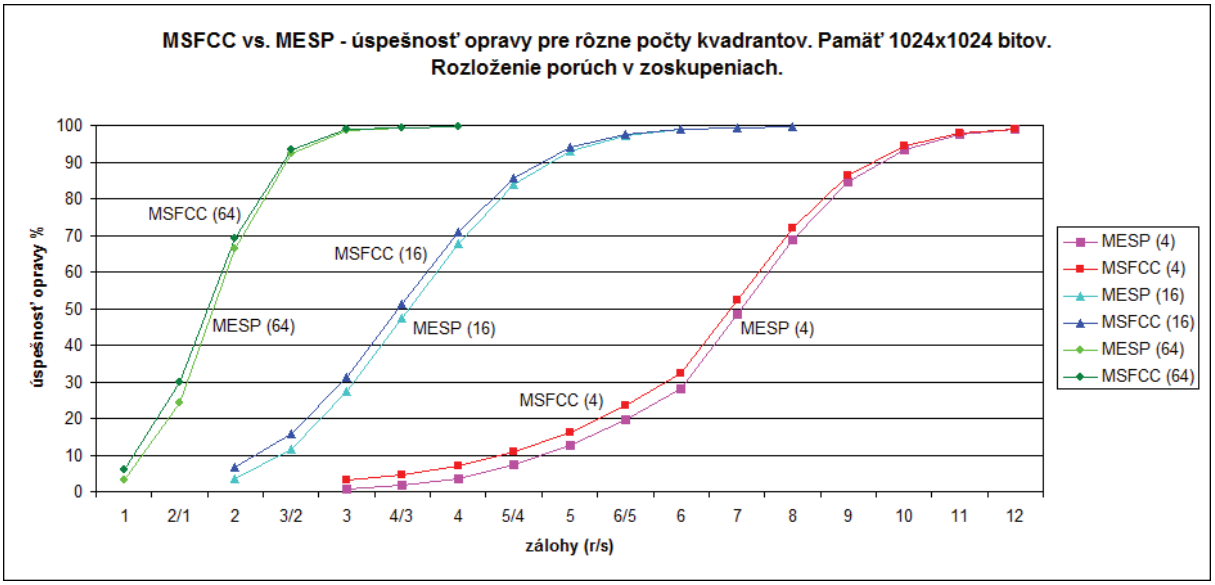
$$úspešnosť\_opravy = \frac{\#opravené\_pamäte}{\#všetky\_pamäte}.100[\%] \qquad (1)$$

Úspešnosť opravy MSFCC bola vyhodnotená a porovnaná s MESP [3], ktorý z doposiaľ známych algoritmov opravy pre blokové pamäte dosahuje najvyššiu úspešnosť, a to pomocou softvérových simulácií. Na tento účel boli vytvorené 2 softvérové nástroje:

- RNDCLUS (*Randomized Clusters Generator*) – generátor poruchových máp pamätí.
- RAREST (*Repair Algorithm Repair-rate Estimator*) – simulátor algoritmov opravy pamätí.

Generátor RNDCLUS bol navrhnutý na základe podrobnej analýzy podobných prístupov ku generovaniu poruchových máp pamätí [5] tak, aby rozloženia porúch v generovaných mapách pamätí zodpovedali v čo najväčšej možnej miere skutočne pozorovaným rozloženiam. Simulátor RAREST simuluje činnosť rôznych algoritmov opravy, pričom ako vstup berie poruchové mapy z generátora RNDCLUS a ako výstup poskytuje odhad úspešnosti opravy podľa (1).

Výsledky porovnania úspešnosti opravy MESP a MSFCC použitím uvedených softvérových nástrojov sú zobrazené na obr. 2 v grafickej forme a tiež zosumarizované v tab. 2.



Obr. 2: Porovnanie úspešnosti opravy MESP a MSFCC.

Tab. 2: Úspešnosť opravy MESP a MSFCC v %.

Kvadranty →	4		16		64	
Algoritmus → / **Zálohy (4 kv.) r/s ↓**	**MESP**	**MSFCC**	**MESP**	**MSFCC**	**MESP**	**MSFCC**
1	-	-	-	-	3,11	6,04
2/1	-	-	-	-	24,22	29,96
2	-	-	3,38	6,64	66,15	69,02
3/2	-	-	11,72	15,74	92,32	93,32
3	0,63	3,12	27,31	31,36	98,70	98,83
4/3	1,61	4,54	47,20	51,16	99,44	99,46
4	3,51	6,94	67,73	70,80	99,60	99,61
5/4	7,24	10,74	83,79	85,68	-	-
5	12,56	16,16	93,08	93,94	-	-
6/5	19,56	23,45	97,32	97,60	-	-
6	28,01	32,19	98,78	98,89	-	-
7	48,28	52,14	99,44	99,47	-	-
8	68,64	71,85	99,60	99,61	-	-
9	84,43	86,39	-	-	-	-
10	93,46	94,30	-	-	-	-
11	97,48	97,75	-	-	-	-
12	98,81	98,93	-	-	-	-

Experiment prebehol na vygenerovanej množine 100 000 pamätí veľkosti 1024x1024 bitov (1MB je najčastejšie sa vyskytujúca veľkosť vnorených pamätí [5]) rozdelených na 4, 16 alebo 64 kvadrantov. Počet kvadrantov je naznačený číslami v zátvorkách na obr. 2. Poruchy v pamätiach sa vyskytovali v zoskupeniach (*clusters*), čo je pozorované často aj v reálnych pamätiach [5]. Použitý počet záložných blokov sa pohyboval od 1 záložného riadkového bloku a 1 stĺpcového bloku (1r+1s) až po 12r+12s. Uvedený rozsah bol platný v pamätiach rozdelených na 4 kvadranty. V pamätiach rozdelených na 16 resp. 64 kvadrantov bol počet záloh 2 krát resp. 4 krát vyšší (dôvodom je delenie záloh na záložné bloky v blokových pamätiach). Napríklad použitie záloh 3r+3s pre 4-kvadrantovú pamäť zodpovedá použitiu 6r+6s pre 16-kvadrantovú pamäť a podobne.

Výsledky potvrdili predpoklad, že použitie hybridného algoritmu opravy so zložitejším princípom v blokovej architektúre pamätí by mohlo viesť ku vyššej úspešnosti opravy ako u doteraz známych prístupov. Pre niektoré kombinácie počtov použitých záloh bolo dosiahnuté navýšenie úspešnosti až na úrovni 4 %, čo napríklad v prípade 1 milióna poruchových pamätí predstavuje až 40000 pamätí. Ďalej sa potvrdil predpoklad [2], že zvyšovaním počtu kvadrantov pri zodpovedajúcich počtoch záloh sa zvyšuje úspešnosť opravy algoritmov určených pre blokové pamäte.

Nasledujúca skupina experimentov mala za cieľ overiť predpoklad, že čím menej porúch je v pamätiach rozložených v zoskupeniach, a teda čím viac porúch je rozmiestnených náhodne, tým viac sa stráca rozdiel v úspešnosti opravy medzi algoritmami MESP a MSFCC. Dôvodom je vysoký podiel samostatných porúch v pamätiach s náhodným rozložením, a teda pokročilé princípy MSFCC sa neuplatnia tak často ako pri zoskupených poruchách. Na prideľovanie záloh pre samostatné poruchy sú postačujúce aj jednoduché algoritmy opravy. Daný predpoklad sa podarilo overiť [5].

Ďalšia skupina experimentov mala za cieľ overiť predpoklad, že aj v blokových pamätiach menších ako 1MB je použitie nového algoritmu MSFCC rovnako výhodné. Daný predpoklad sa podarilo overiť na pamätiach rôznych veľkostí od 64B do 256kB [5].

5 Záver

Článok nadväzuje na [4], kde boli opísané základné koncepty návrhu nového algoritmu opravy pamätí RAM určeného pre opravu bitovo orientovaných pamätí s blokovou architektúrou s názvom MSFCC.

Článok opisuje základné vlastnosti MSFCC, ale viac je zameraný na uvedenie navrhnutých modifikácií a rozšírení pôvodného algoritmu SFCC, na ktorého princípoch bol návrh nového algoritmu MSFCC založený a na uvedenie experimentálnych výsledkov porovnania úspešnosti opravy nového a existujúceho algoritmu. Niektoré navrhnuté zmeny boli nevyhnutné z dôvodu úspešnej adaptácie do blokovej architektúry pamätí, kde sa používa iný formát adresovania pamäti. Ďalšie zmeny boli do adaptácie zahrnuté z dôvodu nájdenia určitých nedostatkov a nepresností SFCC, ktoré znižovali jeho úspešnosť opravy. Adaptovaný algoritmus MSFCC odstránením nájdených nedostatkov nadobudol schopnosť opraviť aj poruchové pamäte s niektorými špeciálnymi rozloženiami porúch, ktoré by nebol schopný opraviť v prípade, že by bol iba adaptáciou SFCC v jeho pôvodnej verzii.

Súčasťou návrhu MSFCC sú aj podrobné nové návrhy postupov riešenia niektorých dôležitých aspektov algoritmu, a to aj s ošetrením špeciálnych prípadov, ktoré môžu nastať. Nie je zrejmé, či SFCC s výskytom takýchto prípadov počítal, avšak u MSFCC je to už zaručené. Tieto návrhy sú opísané až na úroveň základných operácií nad pamäťami CAM, čo by mohlo zjednodušiť prípadnú implementáciu algoritmu.

Na experimentálne overenie a porovnanie úspešnosti opravy MSFCC s doposiaľ najlepším algoritmom na opravu blokových pamätí (MESP) boli vytvorené pomocné softvérové nástroje: generátor poruchových pamätí RNDCLUS a simulátor algoritmov opravy RAREST. Použitím vytvorených nástrojov s nastavením hodnôt parametrov simulácií, ktoré boli inšpirované inou dostupnou literatúrou, sa podarila overiť vhodnosť použitia MSFCC.

Výsledkom je algoritmus MSFCC (aj publikovaný, napríklad [6]), vhodný na opravu blokových pamätí, ktorý dosahuje vyššiu úspešnosť opravy ako doteraz najlepší známy podobný algoritmus MESP. Zároveň ako jediný z tejto kategórie algoritmov garantuje nájdenie optimálneho riešenia opravy pamäte. Podrobnejšie sa o postupoch adaptácie pôvodného algoritmu do blokovej architektúry v článku nepojednáva. Viac informácií je k dispozícii v predbežnej verzii dizert. práce [5].

Poďakovanie

Práca je podporovaná Slovenskou vedeckou grantovou agentúrou MŠVVaŠ SR a SAV, VEGA 1/1008/12.

Literatúra

[1] Semico Research Corp.: Semico: Sysem(s)-on-a-Chip – A Braver New World. 2007. URL – http://www.semico.com/content/semico-systems-chip-%E2%80%93-braver-new-world (pristúpené 20. 6. 2014).

[2] Jeong, W., Kang, I., Jin, K., Kang, S.: A Fast Built-in Redundancy Analysis for Memories with Optimal Repair Rate Using a Line-Based Search Tree. IEEE Trans. on VLSI systems, vol. 17, 2009, no. 12, s. 1665-1678.

[3] Yang, C.-L. et al.: Efficient BISR Techniques for Embedded Memories Considering Cluster Faults. IEEE Trans. on VLSI systems, vol. 18, 2009, no. 2, s. 184-193.

[4] Krištofík, Š.: Algoritmus vstavanej opravy pre vnorené pamäte s blokovou architektúrou záloh. Počítačové architektúry a diagnostika (PAD), 2012, ISBN 978-80-01-05106-1, s. 103-108.

[5] Krištofík, Š.: Príspevok k architektúram a algoritmom samočinnej opravy pamätí RAM. URL – http://student.fiit.stuba.sk/~kristofi03/DP_V_Kristofik.pdf (pristúpené 3. 7. 2014). FIIT STU, dizertačná práca, predbežná verzia. 120 s.

[6] Krištofík, Š., Gramatová, E.: Redundancy Algorithm for Embedded Memories with Block-Based Architecture. In Proc. IEEE 16th Int. Symp. Design & Diag. El. Circuits & Systems (DDECS), 2013, s. 272-274.

JABLOTRON 100

Alarm s revolučním ovládáním

www.jablotron.cz

JABLOTRON
CREATING ALARMS


PRECIOSA
Nadace Preciosa
pomáhá už 20 let
PRECIOSA

Rejstřík jmen [1]:

[1] Kurzívou jsou uvedeni školitelé…

Název	**Sborník příspěvků PAD-2014 – elektronická verze**
Autor	Autoři jednotlivých příspěvků prof. Ing. Zdeněk Plíva, Ph.D., Ing. Martin Rozkovec, Ph.D. (editor publikace)
Určeno	pro účastníky semináře
Vydavatel	Technická univerzita v Liberci
Schváleno	Rektorátem TU v Liberci dne 30. 7. 2014, čj. RE 66/14
Vyšlo	v srpnu 2014
Počet stran	174
Vydání	první
Číslo publikace	55-065-14
ISBN	978-80-7494-027-9

Tato publikace neprošla redakční ani jazykovou úpravou.

9 788074 940279

ISBN: 978-80-7494-027-9