

Evoluční snižování příkonu: Od obvodů na úrovni tranzistorů po neuronové sítě na čipu

Vojtěch Mrázek

2. ročník, prezenční studium

Školitel: Lukáš Sekanina, specialista: Zdeněk Vašíček

Vysoké učení technické v Brně

Božetěchova 2, 612 66 Brno

imrazek@fit.vutbr.cz

Abstrakt—Snižování příkonu integrovaných obvodů je v dnešní době, například u mobilních aplikací, velmi důležité. Práce se zabývá využitím evolučních algoritmů pro optimalizaci příkonu kombinačních obvodů. Zaměřuje se zejména na aplikaci této optimalizace v reálných systémech. Na čtyřech vybraných aplikacích ukazuje možnosti snížení příkonu pomocí evolučního algoritmu. Byly navrženy nové prvky vhodné do technologické knihovny definující strukturu jednotlivých hradel na úrovni tranzistorů, byla snížena spotřeba mediánových filtrů, byly představeny nové metody aproximačního řazení a zefektivněna energetická náročnost klasifikace pomocí neuronových sítí. Všechny tyto aplikace spojuje stejná metoda návrhu využívající evoluční přístup.

Klíčová slova—Evoluční návrh, tranzistorová úroveň, aproximace, neuronové sítě, řazení

I. ÚVOD

Evoluční návrh obvodů je metoda, která používá biologií inspirované prohledávací algoritmy pro syntézu a optimalizaci elektronických obvodů. Tento návrh umožnil získat mnoho zajímavých výsledků, ovšem má několik základních nedostatků, které je nutné řešit. Prvním problémem je, že navržené obvody často nereflektovaly požadavky HW komunity. Jedním z důvodů je fakt, že optimalizace většinou probíhala s ohledem na počet výpočetních prvků, což nemusí přímo odpovídat skutečným vlastnostem obvodů z důvodu různé složitosti jednotlivých prvků, výstupní kapacity, přepínací aktivity a podobně. Druhou nevýhodou je nemožnost dobré škálovatelnosti evolučního návrhu. Problém škálovatelnosti je obtížně řešitelný, ale existují techniky (například využití pokročilých verifikačních technik či dekompozice), které umožňují evolučně optimalizovat i složitější obvody.

Proto je cílem této práce navrhnout metodiku, která umožní upravit evoluční návrh obvodů tak, aby navržené obvody byly použitelné v reálných případech. Jako vhodné parametry pro optimalizaci se jeví plocha, zpoždění a příkon. V dnešní době vysokého stupně integrace už vliv plochy není tak důležitý oproti požadavkům na příkon. Díky rozvoji přenosných zařízení, jako jsou mobilní technologie, wearable elektronika a podobně, je právě spotřeba elektronických zařízení klíčovou otázkou.

V práci je rozšířen evoluční návrh, zejména kartézské genetické programování [1], o možnost navrhovat obvody s nízkým příkonem. Práce ukazuje, že redukce spotřeby elektrické energie je možné dosáhnout na více úrovních abstrakce (popisu) obvodů. Pohybuje se na strukturální doméně, jako jsou tranzistory a hradla, a využívá snížení příkonu s využitím znalosti funkce na doméně behaviorální.

II. CÍLE DISERTAČNÍ PRÁCE

Je známo, že existuje řada konvenčních přístupů pro redukcii spotřeby kombinačních obvodů. Příkon je možné snižovat jak na úrovni technologické (volba vhodné technologie a knihovny [2]), tak na úrovni zapojení jednotlivých částí obvodů — tranzistorů [3] i hradel. Další úsporu můžeme dosáhnout využitím znalosti aplikace těchto obvodů a redukcí obvyklé funkčnosti. Typickým příkladem je aproximační počítání, kdy prvky pracují nepřesně za předpokladu, že výsledná aplikace je částečně imunní vůči chybám (*error resilient*) [4]. Abychom mohli příkon za pomoci navrženého algoritmu snižovat, musíme ho být schopni správně odhadnout, abychom určili vhodnost provedené změny obvodu.

Mimo to se také ukazuje, že evoluční algoritmy, zejména kartézské genetické programování, umožňují efektivně navrhovat kombinační obvody na úrovni tranzistorů [5], [6], [7], hradel [8], [9] či aproximačních obvodů [10]. Ovšem tyto metody nebyly kombinovány. Z dosavadních znalostí můžeme formulovat hypotézu disertační práce: *Za pomoci evolučního algoritmu upraveného pro konkrétní aplikaci je možné snížit celkový příkon kombinačních obvodů popsaných na různých úrovních oproti řešením poskytovaným konvenčním přístupem.* Byly stanoveny následující cíle pro disertační práci:

- 1) Vytvoření výpočetně nenáročného metody pro odhad příkonu číslicového obvodu.
- 2) Návrh evolučního algoritmu využívajícího odhad příkonu pro optimalizaci celkové spotřeby.
- 3) Využití akcelerace pro urychlení evolučního návrhu.
- 4) Experimentální ověření pro různé aplikace a úrovně popisu.
- 5) Porovnání navrženého přístupu s konvenčními metodami.

Demonstrační aplikace na různých úrovních popisu budou následující:

- Optimalizace obvodů technologické knihovny (*úroveň tranzistorů*).
- Zefektivnění klasifikace pomocí neuronových sítí (*úroveň hradel a funkce*).
- Snížení spotřeby mediánových filtrů v oblasti obrazových filtrů a filtrů signálů (*úroveň funkce*).
- Aplikace přibližných řadicích sítí (*úroveň funkce*).

III. NAVRŽENÉ METODY A ZÍSKANÉ VÝSLEDKY

Několik případových studií úzce souvisejících s tématem jsem již provedl a publikoval, nebo jsou příslušné odborné články připravovány. Tato kapitola poskytuje pouze základní přehled dosažených výsledků, podrobnější informace nalezneme v odkazovaných publikacích. Mimo níže zmíněné publikace byl vytvořen článek o multikriteriálním návrhu aproximačních aritmetických obvodů [11].

A. Optimalizace obvodů popsaných na úrovni tranzistorů

Abychom byli schopni navrhovat obvody na úrovni tranzistorů, je nutné upravit evoluční algoritmus, který je obvykle stavěn pro návrh obvodů na úrovni hradel. Úprava spočívá v zavedení nového kódování obvodů, změně výpočtu fitness funkce a zavedení vhodného odhadu spotřeby. V rámci porovnání s klasickými přístupy budou navržené obvody porovnány vůči technice přepisu z hradlové úrovně.

V článku [12] byla představena nová reprezentace obvodů popsaných na úrovni tranzistorů. Díky použití vlastního diskretního simulátoru, který pracoval s vícehodnotovou logikou, se podařilo evolučně navrhnout obvody o velikosti až 30 tranzistorů. To byl poměrně velký pokrok oproti předchozím výzkumům, kde se cílilo na obvody do 10 tranzistorů [13], [7].

Abychom urychlili celkový evoluční návrh obvodů, byl vytvořen akcelerátor využívající čip Xilinx Zynq. v hardware se akceleruje simulace obvodů pro 2^n vstupních kombinací, kde n je počet vstupních signálů. Návrh se podařilo zrychlit až $4.7\times$ oproti procesoru Intel Xeon. Samotná akcelerovaná simulace obvodů je dokonce $25\times$ rychlejší než stejný diskretní algoritmus na procesoru. Výsledky byly publikovány na konferenci ICES [14].

Předcházející práce využívající evoluci se většinou zabývaly optimalizací na počet tranzistorů. Je zřejmé, že na spotřebu mají vliv i další parametry, zejména přepínací aktivita jednotlivých tranzistorů. Proto jsem navrhl metodu, která odhaduje příkon obvodu popsaného na úrovni tranzistorů a je inspirovaná metodami používanými pro odhad spotřeby na úrovni hradel. Využívá se výsledků z diskretní simulace, kde ke každému tranzistoru je možné určit četnost výskytu kombinace nastavení vstupu *source* a vstupu *gate*. Pro aktivní stavy se pak vypočítá pravděpodobnost přepnutí ze stavu a do stavu b (kde stav je dvojice hodnoty *source* a *gate*). V simulátoru SPICE byla určena spotřeba pro každou možnou dvojici přepnutí, kde pak součtem součinů spotřeby přechodů mezi stavy a jejich pravděpodobností dojde k odhadu spotřeby. Ukázalo se,

že tato metoda přináší zajímavé výsledky, kdy jsme schopni snížit příkon o 4 – 12 % s tím, že zpoždění obvodu zůstane přibližně zachováno, nebo se zlepší. Tímto způsobem byly optimalizovány obvody obsahující až 300 tranzistorů. Navržená metoda optimalizace byla prezentována na konferenci EUC [15].

B. Optimalizace a aproximace mediánových filtrů

Existuje několik typů filtrů, které jsou postaveny na řazení prvků pomocí řadicích sítí. Typickým případem je mediánový filtr. Pro tyto aplikace dokáží evoluční algoritmy najít kvalitní implementaci pomocí elementárních komponent [16]. Ovšem s rostoucím počtem vstupů tyto filtry narůstají do velkých rozměrů a jejich energetická náročnost roste. Proto bylo navázáno na výsledky prezentované v [10] a byl navržen efektivní algoritmus pro optimalizaci filtrů na bázi řadicích sítí i za cenu rozumného snížení přesnosti. Vzhledem k tomu, že optimální řadicí sítě jsou poměrně dobře prozkoumané [17], [18], zaměřím se spíše na optimalizaci filtrů vzniklých z optimálních řadicích sítí [19].

1) *Metrika vyhodnocení přesnosti*: Stejně jako u všech přibližných výpočtů je nutné definovat odchylky od správného řešení. Klasický přístup pro výpočet chyby používaný v předcházejících výzkumech byl založen na výpočtu sumy odchylek pro zadaná vstupní data. Pokud bychom pro mediánový filtr s 9 vstupy (na 8 bitech) počítali chybu přes všechny vstupy, byl by celkový počet kombinací $2^{9^8} = 2^{72} \sim 4.7 \cdot 10^{21}$. Tento počet vstupních kombinací je neřešitelný, proto se používá pouze podmnožina vstupních kombinací. Navíc výsledné ohodnocení není příliš vypovídající.

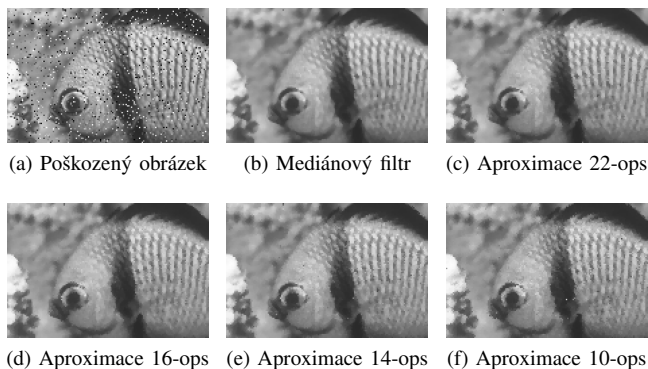
Z těchto důvodů jsem navrhl novou metriku pro určení chyby nazvanou *permutační princip*, která je využitelná pro návrh mediánového filtru. Stejně tak je aplikovatelná i pro řadicí sítě. Knut dokázal, že pro ověření funkčnosti řadicí sítě nemusíme použít všechny možné kombinace, ale stačí nám pouze všechny kombinace 0 a 1 (tzv. zero-one princip) [17]. Podobně můžeme dokázat, že pro ověření funkčnosti n vstupového mediánu můžeme použít permutace množiny $S = \{0, 1, \dots, n-1\}$. Pokud budeme sledovat výstup i -tého prvku, tak pro všechny permutace množiny S bude mít plně funkční řadicí síť výstup i . Když síť nebude plně funkční, tak rozdíl hodnoty výstupu a i udává tzv. poziční odchylku, jejíž rozložení můžeme sledovat. Sledování odchylky snižuje složitost z exponenciální na faktoriálovou. Mimo to získáme přehlednější ohodnocení, které je navíc nezávislé na výběru podmnožiny vstupních dat. Tato metrika je představena v publikacích [20], [21]. Navíc v časopise [21] ($Q2$, $IF=1.143$) byl představen formální důkaz o tom, že otestováním všech permutací kompletně ověříme řadicí síť.

Ukázalo se, jak je prezentováno v odeslané publikaci [22], že jsme schopni snížit složitost z $n!$ na 2^n upravením *zero-one principu*. S využitím formálního aparátu, jako je BDD (Binary Decision Diagram — binární rozhodovací stromy, řešitelné např. pomocí knihovny BuDDy), jsme schopni získat rozložení chyb ve velmi krátkém čase.

2) *Přibližné mediánové filtry*: Mediánový filtr je specifický druh komparátorové sítě, která má n vstupů a jeden výstup. Byly navrženy dva druhy implementace — softwarová a hardwarová. Oba druhy implementace vyžadují specifické kódování problému. Pro softwarovou implementaci se jedná o zapojení dvouvstupových bloků s jedním výstupem realizujících operaci výběru minima nebo maxima. Hardwarová implementace vychází z použití *compare&swap* bloků. Vzhledem k paralelnímu zpracování jsou jednotlivé operace naplánovány pomocí ASAP algoritmu. Výstupem plánování je počet bloků, počet registrů pro zřetěženou architekturu a její latence.

Vstupem evolučního algoritmu byl, kromě nastavení parametrů běhu, plně funkční obvod a omezení (*constraints*) požadovaného obvodu. Omezena byla maximální velikost a latence. Algoritmus pracoval ve dvou fázích. Během první fáze byl výchozí obvod zmenšován tak, aby vyhovoval požadavkům. Při druhé fázi byla evolučně optimalizována kvalita (určena pomocí permutačního či BDD principu) při splnění všech omezení.

Navržené obvody byly testovány v reálných aplikacích zpracování obrazu či signálů. Na obrázku 1 můžeme vidět kvalitu filtrace obrazů pomocí aproximačních mediánových filtrů navržených pomocí představeného algoritmu. Můžeme si všimnout, že při implementaci využívající 14 operací je jen nepatrný rozdíl oproti plně funkční implementaci, přičemž úspora energie je více jak 50 %.



Obrázek 1: Detailní výřez obrázku, který byl poškozen 10% náhodným šumem typu pepř-sůl. Obrázek byl filtrován b) 9-vstupým mediánem a aproximačními filtry využívajícími c) 22 (73 %) operací, d) 16 (53 %), e) 14 (46 %) a f) 10 (33 %) operací.

3) *Analýza spotřeby mediánových filtrů*: Jak již bylo řečeno, byly optimalizovány dva druhy implementací aproximačních mediánových filtrů — hardwarová a softwarová. Hardwarové implementace byly realizovány pomocí zřetěžené linky. Tyto obvody byly syntetizovány s využitím 45 nm technologie. Ukázalo se, že celková spotřeba nejvíce koreluje s váženým součtem počtu komparátorů a registrů. Tento výpočet spotřeby a plochy byl potom použit v evolučním návrhu při omezení velikosti obvodu. Výsledky budou odeslány k publikaci [22].

Dále byly aproximovány mediánové filtry pro vestavěné zařízení (mikroprocesory). Na čtyřech architekturách byla

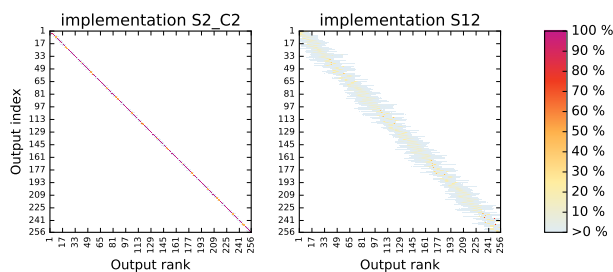
změřena spotřeba mediánového filtrování vstupních dat. Analýzou spotřeby reálných systémů se ukázalo, že celková spotřeba přímo závisí na počtu operací a to, že aproximací filtru ve vestavěném systému jsme schopni energii ušetřit.

C. Přibližné řadicí sítě

Vzhledem k tomu, že genetické algoritmy přinesly zajímavé výsledky v oblasti řadicích sítí, byla tato oblast vybrána jako další demonstrační aplikace. Řadicí sítě, jejichž výsledkem je určitá permutace vstupů, nachází uplatnění nejen v oblasti řazení v HW akcelerátorech, ale také v aritmetických operacích či řízení přepínacích sítí [23].

Při analýze konvenčních způsobů se ukázalo, že je nejvýhodnější stavět síť z menších prvků, podobně jak to dělá algoritmus Bitonic. Menší prvky představují řadicí a spojovací sítě. Spojovací sítě se od klasických liší v tom, že spojují dvě seřazené posloupnosti do jedné. Při evolučním návrhu byl využit permutační princip vyhodnocení kvality kandidátní řadicí sítě.

Podarilo se nalézt implementace těchto menších bloků s různou kvalitou a velikostí. Evolučně navrženými bloky byly nahrazeny různé části plně funkční sítě. Na obrázku 2 je znázorněna kvalita (četnost odchylek) pro dvě vybraná řešení s 256 vstupy. Implementace S2_C2 využívá 89 % operací s tím, že v 99 % případů se bude výsledek lišit maximálně o jednu pozici oproti seřazené posloupnosti. Druhá vybraná implementace S12 využívá 46 % operací s tím, že v 95 % případů bude odchylka do 10 pozic. Ukázalo se, že při použití stejných bloků pro 1024 vstupů je kvalita řešení (t.j. kvartily odchylek) stejná, pouze se liší redukce operací. Výsledky implementace navržených řešení v FPGA a v ASIC obvodech byly odeslány na konferenci PATMOS [24].



Obrázek 2: Histogram četnosti výskytů jednotlivých hodnot z posloupnosti $\{1 \dots 256\}$ na jednotlivých pozicích pro dvě vybrané implementace aproximačních řadicích sítí s 256 vstupy.

D. Optimalizace násobiček pro NN

Neuronové sítě (NN) představují další velkou skupinu aplikací, jejichž celkovou spotřebu je dobré snížit. Nejnáročnější a nejčastější operací je násobení, a proto je vhodné optimalizovat zejména je. Vzhledem k tomu, že pro realizaci násobení je řada velmi efektivních algoritmů [2], je velmi malý prostor pro snížení celkové spotřeby. Pokud však budeme tolerovat chybu na výstupu, prostor se rapidně zvětšuje. Důležité je správně určit, jakou chybu si můžeme dovolit. Jako testovací případ bylo zvoleno rozpoznávání písmen MNIST databáze.

Pokud použijeme násobičky, které vykazují 5% chybu výstupu vůči rozsahu, dojde ke snížení celkové spotřeby výpočetní jednotky NN. Při porovnání schopnosti klasifikace však klesne úspěšnost z 94.16 % na 10.77 %. Podrobnější zkoumání však ukázalo, že problém je v tom, že přibližně 80 % násobení má mít výsledek 0. Pokud navrheme násobičku tak, aby ve všech případech kromě násobení 0 měla chybu do 5 % a při násobení 0 by výsledek byl přesný, celková přesnost klasifikace stoupne až na 94.20 %.

V dalším kroku bylo využito genetické programování pro návrh 8 a 12 bitových násobiček pro různé maximální chyby s tím, že násobení nulou je vždy přesné. Ukázalo se, že například 91% redukce příkonu násobení vede k poklesu přesnosti klasifikace po přetrénování o méně než 2.8 % u SVHN datasetu, u datasetu MNIST je pokles přesnosti v řádu desetin procent. Výsledky byly přijaty na HW konferenci ICCAD'16 [25].

IV. ZÁVĚR

Z výše uvedených ukávek můžeme vidět, že metodika snižování příkonu pomocí genetického programování kombinuje znalosti nejen z oblasti evolučního návrhu, ale také i z oblasti návrhu obvodů s ohledem na příkon. Toto využití evolučních algoritmů nebylo zatím v literatuře publikované. Navíc výsledky návrhu integrovaných obvodů byly přijaty HW komunitou na specializovaných konferencích. Disertační práce by měla celkovou metodiku návrhu nízkopříkonových kombináčnických obvodů pomocí nekonvenčního přístupu představit a ukázat její výhody na několika příkladech. Hlavní myšlenka je založena na kombinaci vhodného kódování problému, správného odhadu spotřeby a zejména na analýze cílové aplikace. Celkový přínosem práce je zejména prokázání toho, že evoluční návrh může přinášet zajímavé výsledky optimalizace příkonu na různých úrovních popisu.

Proto byl s ohledem na hypotézu a cíle disertační práce vytvořen následující plán. Některé cíle již byly splněny a ostatní jsou rozpracované.

- Evoluční optimalizace obvodů popsaných na úrovni tranzistorů [12], [14] a optimalizace jejich spotřeby [15].
- Návrh nové metriky pro přibližné mediánové filtry a jejich aplikace [20], [21].
- Návrh přibližných mediánových filtrů a řadicích sítí pomocí formálních metod [22] (*rozpracováno*).
- Návrh a aplikace přibližných řadicích sítí [24].
- Snížení spotřeby neuronových sítí [25] (*rozpracováno*).

PODĚKOVÁNÍ

Tato práce vznikla za podpory projektu Vysokého učení technického v Brně FIT-S-14-2297 Architektury paralelních a vestavěných počítačových systémů.

REFERENCE

- [1] J. F. Miller, Ed., *Cartesian genetic programming*, ser. Natural Computing Series. Berlin: Springer, 2011.
- [2] N. H. Weste and D. Harris, *CMOS VLSI design: a circuits and systems perspective*, 3rd ed. Boston, USA: Addison-Wesley, 2005.
- [3] D. Soudris, C. Piguet, and C. Goutis, *Designing CMOS Circuits for Low Power*, ser. European low-power initiative for electronic system design. Springer, 2002.
- [4] H. Cho, L. Leem, and S. Mitra, "ERSA: Error resilient system architecture for probabilistic applications," *IEEE Tr. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 31, no. 4, pp. 546–558, April 2012.
- [5] A. Stoica, R. Zebulum, X. Guo *et al.*, "Taking evolutionary circuit design from experimentation to implementation: some useful techniques and a silicon demonstration," *IEEE Proc. Computers and Digital Techniques*, vol. 151, no. 4, pp. 295–300, July 2004.
- [6] L. Zaloudek and L. Sekanina, "Transistor-level evolution of digital circuits using a special circuit simulator," in *Evolvable Systems: From Biology to Hardware*, ser. LNCS, vol. 5216. Springer Verlag, 2008.
- [7] M. Trefzer, "Evolution of Transistor Circuits," Ph.D. dissertation, Ruprecht-Karls-Universität Heidelberg, 2006.
- [8] Z. Vasicek, "Cartesian gp in optimization of combinational circuits with hundreds of inputs and thousands of gates," in *Genetic Programming, 18th European Conference, EuroGP 2015*, ser. LCNS 9025. Springer International Publishing, 2015, pp. 139–150.
- [9] Z. Vasicek and L. Sekanina, "How to evolve complex combinational circuits from scratch?" in *2014 IEEE International Conference on Evolvable Systems Proceedings*, 2014, pp. 133–140.
- [10] Z. Vasicek and L. Sekanina, "Evolutionary approach to approximate digital circuits design," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 432–444, 2015.
- [11] R. Hrbacek, V. Mrazek, and Z. Vasicek, "Automatic design of approximate circuits by means of multi-objective evolutionary algorithms," in *Proceedings of the 11th International Conference on Design & Technology of Integrated Systems in Nanoscale Era*. IEEE Computer Society, 2016, pp. 239–244.
- [12] V. Mrazek and Z. Vasicek, "Evolutionary design of transistor level digital circuits using discrete simulation," in *Genetic Programming*, ser. LNCS. Springer International Publishing, 2015, vol. 9025, pp. 66–77.
- [13] J. Walker, J. Hilder, and A. Tyrrell, "Towards evolving industry-feasible intrinsic variability tolerant cmos designs," in *IEEE Congress on Evolutionary Computation*, 2009, pp. 1591–1598.
- [14] V. Mrazek and Z. Vasicek, "Acceleration of transistor-level evolution using xilinx zynq platform," in *IEEE Int. Conf. on Evolvable Systems (ICES)*, Dec 2014, pp. 9–16.
- [15] V. Mrazek and Z. Vasicek, "Automatic design of low-power VLSI circuits: Accurate and approximate multipliers," in *IEEE Int. Conf. on Embedded and Ubiquitous Computing (EUC)*, Oct 2015, pp. 106–113.
- [16] S.-S. Choi and B. R. Moon, "Isomorphism, normalization, and a genetic algorithm for sorting network optimization," in *GECCO*, 2002, pp. 327–334.
- [17] D. E. Knuth, *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1998.
- [18] D. Bundala and J. Zavodný, "Optimal sorting networks," in *Language and Automata Theory and Applications*, ser. LNCS. Springer International Publishing, 2014, vol. 8370, pp. 236–247.
- [19] N. Devillard, "Fast Median Search: An ANSI C Implementation," 1998, <http://ndevilla.free.fr/median/median.pdf>.
- [20] V. Mrazek, Z. Vasicek, and L. Sekanina, "Evolutionary approximation of software for embedded systems: Median function," in *Proc. of the Genetic and Evolutionary Computation Conference (GECCO)*, ser. ACM, 2015, pp. 795–801.
- [21] Z. Vasicek and V. Mrazek, "Trading between quality and non-functional properties of median filter in embedded systems," *Genetic Programming and Evolvable Machines*, p. 35 (*přijato*).
- [22] Z. Vasicek and V. Mrazek, "Quality-driven design of low-power approximate median architectures with provable error and suitable for high-throughput systems," in (*připraveno k odeslání*), 2017, p. 6.
- [23] K. E. Batcher, "Sorting networks and their applications," in *Proceedings of the April 30–May 2, 1968, Spring Joint Computer Conference*, ser. AFIPS '68 (Spring). New York, NY, USA: ACM, 1968, pp. 307–314.
- [24] V. Mrazek and Z. Vasicek, "Automatic design of arbitrary-size approximate sorting networks with error guarantee," in *26th International Workshop on Power and Timing Modeling, Optimization and Simulation (odesláno)*, 2016, p. 8.
- [25] V. Mrazek, S. S. Sarwar, L. Sekanina, Z. Vasicek, and K. Roy, "Design of power-efficient approximate multipliers for approximate artificial neural networks," in *2016 International Conference On Computer Aided Design (ICCAD) (přijato)*, 2016, p. 7.