Evoluční návrh nízkopříkonových obvodů

Vojtěch Mrázek
Výpočetní technika a informatika, 1. ročník, prezenční forma
Školitel: Lukáš Sekanina, Specialista: Zdeněk Vašíček

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

imrazek@fit.vutbr.cz

Abstrakt. Práce se zabývá tématem snižování příkonu digitálních obvodů. Rozebírá výsledky aktuální práce v oblasti využití nekonvenčních metod pro snížení spotřeby integrovaných obvodů. Prvně je ukázán evoluční návrh obvodů na úrovni tranzistorů, kde se podařilo snížit náročnost návrhu. Díky tomu byly vytvořeny obvody s desítkami tranzistorů, což doposud nebylo pomocí evolučního návrhu možné. Dále byl tento přístup akcelerován v FPGA Zynq se zrychlením $4.7\times$. Byl navržen nový přístup k evoluční optimalizaci těchto obvodů s ohledem na příkon. Tato metoda využívá nový způsob odhadu spotřeby založený na pravděpodobnostním modelu. Mimo to jsou diskutovány možnosti návrhu s ohledem obvodů na úrovni hradel na spotřebu. Navíc je představen i způsob snižování spotřeby omezením funkčnosti, tzv. aproximační počítání. Tento příklad je demonstrován na ukázce výpočtu mediánu.

Klíčová slova. Příkon, VLSI, tranzistorová úroveň, hradla, aproximační počítání.

1 Úvod

Vzhledem k velkému rozmachu mobilních zařízení a další elektroniky, která pracuje nepřetržitě, stoupají požadavky na snižování spotřeby těchto zařízení. Jedny z prvků, které ovlivňují spotřebu, jsou digitální integrované obvody ASIC. Právě těmito obvody se budu dále více zabývat. Optimalizace spotřeby probíhá, stejně jako návrh, na více úrovních [10]. Nejnižší úrovní, která se zabývá příkonem, je mikroelektronika. Nastupují nové technologie výroby a se snižujícími se rozměry spotřeba klesá. Mění se však i další parametry obvodů navržených na moderních technologiích, a proto je musí návrhové systémy respektovat. Spotřebu můžeme ovlivnit mimo jiné napájecím napětím, taktováním, ale také i rozmístěním jednotlivých tranzistorů (tzv. layoutem). Velmi významný vliv má propojení jednotlivých tranzistorů a volba správných zapojení. Dále můžeme příkon optimalizovat na úrovni propojení mezi hradly a dalšími stavebními bloky obvodů. Spotřebu těchto obvodů můžeme upravovat i na nejvyšších úrovních, jako je optimalizace softwaru nebo úprava požadované funkčnosti s využitím aproximačního počítání.

Nižší úrovně, jako je úroveň technologická a úroveň masek, nebudou v práci rozebírány, protože k těmto možnostem optimalizace často nemá vývojář přístup. Kapitola 2 se zabývá optimalizací spotřeby na úrovni tranzistorů a jejich propojení. V kapitole 3 bude prezentován způsob snižování příkonu omezením funkčnosti. Možnosti pokračování práce a popsání celkového tématu disertace je možné nalézt v kapitole 4.

2 Návrh na úrovni tranzistorů

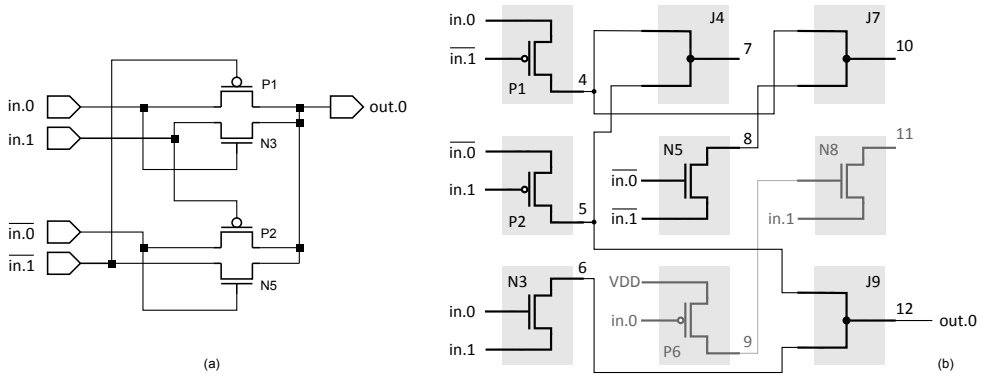
Tranzistorová úroveň popisu umožňuje významně optimalizovat příkon celého obvodu. Například pro implementaci čtyřvstupového obvodu AND-OR-INVERT může být ušetřeno 60 % prostředků při přechodu z popisu na úrovni hradel na úroveň tranzistorů. Při tvorbě jednotlivých bloků se na rozdíl od CMOS technologie nemusíme omezovat na ustálený způsob zapojení *komplemementární logiky CMOS*, ale můžeme využít řadu dalších možností. Tím může být například *Pass–Transistor logic* (normální nebo komplementární verze), *Single–Rail Pass–Transistor logic*, dynamické přístupy jako je *pseudo n-MOS* nebo diferenční přístup *Differential Cascade Voltage Switch logic* [7].

V posledních letech několik autorů ukázalo výhody techniky evolučního návrhu obvodů popsaných na úrovni tranzistorů. Tato metoda pracuje na principu generování a testu řady kandidátních řešení. Proto také výkonnost použitého simulátoru má významný vliv na škálovatelnost celého evolučního přístupu. Pro urychlení evoluce Žaloudek et al. navrhl přístup založený na jednoduchém simulátoru [13]. Díky nepřesnostem v simulaci nebyla řada nalezených řešení funkční v reálném prostředí. Jiný přístup navrhl Trefzer [8], který místo použití simulátoru použil rekonfigurovatelný analogový obvod. Nicméně bylo ukázáno, že přibližně 50 % nalezených řešení nebylo funkčních v přesném simulátoru SPICE. Později Walker et al. použil techniku hrubé síly pro evoluci obvodů s odolností vůči variabilitě výrobního procesu [11]. Pro evaluaci využil cluster přesných SPICE simulátorů.

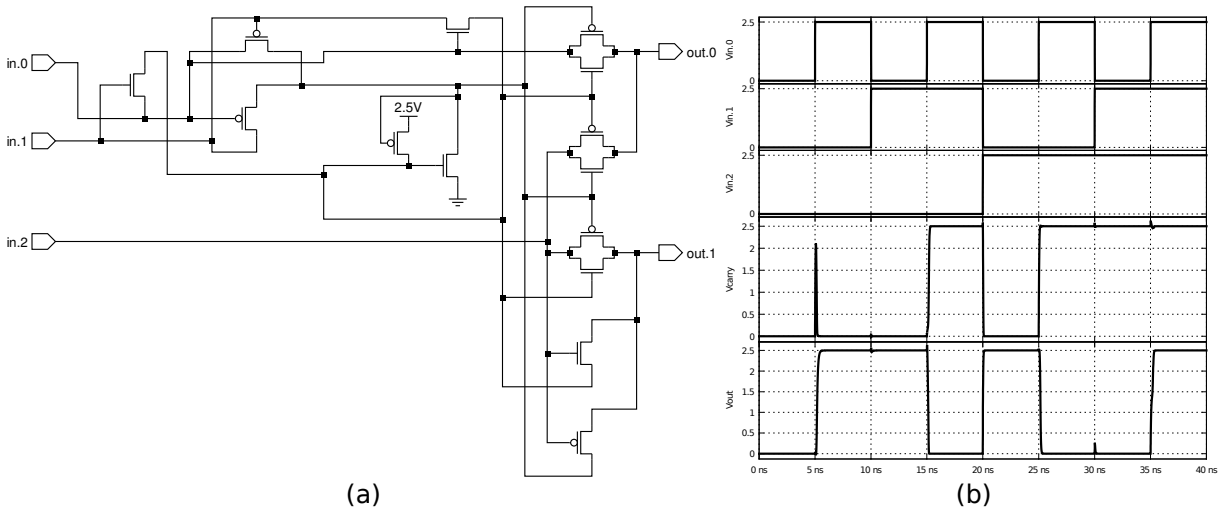
Jak je vidět, předcházející přístupy měly dva základní problémy. Prvním byl problém časové náročnosti simulace ve SPICE, kde model každého tranzistoru má stovky parametrů. Pokud se autoři snažili použít vlastní způsob simulace, projevovala se odchylka od reality. Při větších obvodech se tyto nedostatky začaly projevovat mnohem více a kvůli tomu se podařilo nalézt pouze velmi malé obvody s jednotkami tranzistorů.

2.1 Evoluční návrh obvodů

Vzhledem k náročnosti výpočtu bylo pro návrh obvodů nutné použít vlastní diskrétní simulátor. Tento simulátor vychází z vícehodnotové simulace a snaží se co nejvíce kopírovat chování tranzistorů včetně degradace signálů [12]. Pro návrh byla navržena vlastní reprezentace obvodů odvozená z kartézského genetického programování [1]. Jedná se o pevnou mřížku uzlů, kde každý uzel může plnit funkci *nmos* nebo *pmos tranzistoru*, nebo *propojky*. Bylo ukázáno, že tato reprezentace je dostačující a jsme schopni v ní definovat všechny obvody. Ukázka propojení je znázorněna na obrázku 1.



Obrázek 1: Příklad kandidátního obvodu implementující funkci XNOR s použitím osmi tranzistorů (čtyři jsou využity pro implementaci invertované hodnoty proměnných $\overline{in.0}$ a $\overline{in.1}$) v (a) schematické a (b) interní reprezentaci. Parametry dle [1] jsou následující: n_i=4 (0,V_{dd},in.0,in.1), n_o=1 (out.0), n_c=3, n_r=3, l=2. Chromozom: *(2,-3,pmos)(-2,3,pmos)(3,2,nmos)(4,5,junction)(-3,-2,nmos)(1,2,pmos)(4,8,junction)(9,3,nmos)(5,6,junction)(12).*



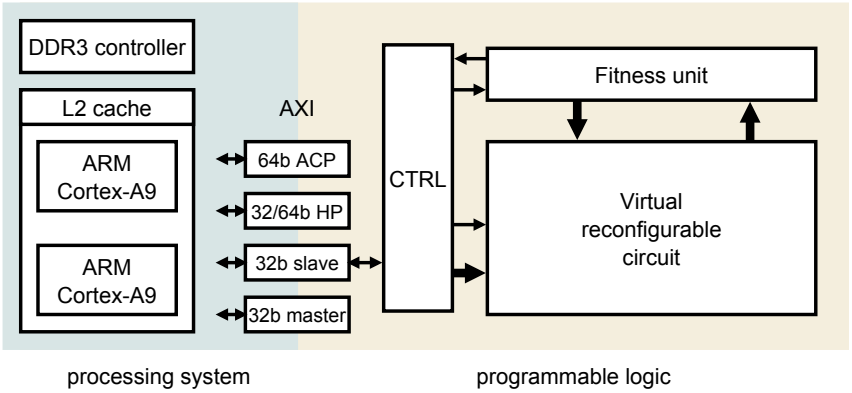
Obrázek 2: (a) Nejmenší a nejrychlejší nalezený obvod obsahující 14 tranzistorů, který plní funkci jednobitové sčítačky. (b) Výstupní signál získaný pomocí SPICE simulátoru.

Jako algoritmus evolučního návrhu byla použita evoluční strategie $(1 + \lambda)$ [1]. Pomocí této techniky se podařilo zlepšit úspěšnost návrhu obvodů na úrovni tranzistorů. Oproti předcházejícím pracím, kde se jednalo o obvody s jednotkami tranzistorů, se při použití nové techniky podařilo objevit obvody obsahující např. 25 tranzistorů. Také se podařilo vytvořit 14 tranzistorovou sčítačku (obrázek 2), která je téměř identická s řešením vytvořeným člověkem – expertem [6].

Další výsledky jsou dostupné v článku shrnujícím problematiku evolučního návrhu obvodů na úrovni tranzistorů [3].

2.2 Akcelerace evolučního návrhu

Pro evoluční návrh je výhodné, abychom byli schopni rychle ohodnotit kandidátní řešení. Výše popsaná metoda nabízí přímo možnost paralelizace na úrovni hardware. Protože se jedná o pevnou mřížku uzlů, je možné realizovat takzvaný virtuální rekonfigurovatelný obvod (VRC). Jedná se o obvod v FPGA, který obsahuje entity a dynamické přepínání funkce těchto entit a jejich propojení. Oproti existujícím VRC, které byly určeny pro kombinační logiku na úrovni hradel, musí náš VRC, určený pro úroveň tranzistorů, umožňovat modelovat tok proudu všemi směry. To komplikuje řešení dynamického propojení entity, takže je nutné implementovat spojování jednotlivých signálů do jednoho pomocí stromové redukce.



Obrázek 3: Architektura HW akcelerátoru v systému Zynq.

Další náročnou částí HW akcelerace je to, že evoluční algoritmus je poměrně složitý a obtížně implementovatelný v FPGA, zejména díky složitosti operace mutace. Aby nedošlo ke zpomalení evolučního návrhu, byl pro řízení použit výkonný procesor. S výhodou byla využita platforma systému na čipu Xilinx Zynq, který kombinuje dvoujádrový procesor ARM s taktem 1 GHz s FPGA částí. Rozložení komponent akcelerátoru je vidět na obrázku 3.

Při použití akcelerátoru dochází ke zrychlení evolučního návrhu pro obvody s 5 vstupy a 80 prvky přibližně $4.7\times$. Samotný akcelerátor bez použití evolučního algoritmu je dokonce $25\times$ rychlejší než diskrétní simulace. Zrychlení oproti SPICE simulátoru je dokonce více než tisícinásobné. Podrobné výsledky byly publikovány v článku [2].

2.3 Optimalizace obvodů s ohledem na příkon

Předcházející práce se většinou zabývaly optimalizací na počet tranzistorů. Je zřejmé, že na spotřebu mají vliv i další parametry, zejména přepínací aktivita jednotlivých tranzistorů. Proto bylo nutné navrhnout metodu, která bude správně odhadovat příkon obvodu popsaného na úrovni tranzistorů. Využívá se výsledků z diskrétní 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*), která je dána

$$P_{a\to b} = 2 \cdot \frac{C_a}{2^i} \cdot \frac{C_b}{2^i}, \tag{1}$$

s tím, že i určuje počet vstupů a C_x určuje četnost stavu x. V simulátoru SPICE byla určena spotřeba pro každou možnou dvojici přepnutí a celková spotřeba tranzistoru je dána jako suma součinů pravděpodobností přechodů a změřených spotřeb pro danou dvojici přechodů. Pro tuto metodu byla na vzorku 200 obvodů zjištěna věrnost odhadu, která určuje to, že pokud je reálná hodnota spotřeby prvního obvodu větší, respektive menší, než spotřeba druhého, tak i odhadnutá spotřeba prvního obvodu musí být větší, respektive menší, než odhadnutá spotřeba druhého. U testovaných obvodů se věrnost odhadu pohybovala mezi 75 – 100 %.

Vzhledem k mírné chybě odhadu byly výsledky jednou za $1000--3000$ generací validovány pomocí simulátoru. Jako případová studie byla představena optimalizace čtyřbitových násobiček, které byly optimalizovány na úrovni hradel. Jedná se o osmivstupové obvody, které obsahují přibližně 300 tranzistorů. Ukázalo se, že touto metodou jsme schopni snížit příkon o 4 – 12 % s tím, že zpoždění obvodu zůstane přibližně zachováno, nebo se zlepší. Také se ukázalo, že je lepší používat metodu kombinující diskrétní i numerickou simulaci ve SPICE spíše než metodu používající pouze numerickou simulaci, protože tvoří variabilnější řešení, která nejsou tolik závislá na použité technologii. Výsledky jsou zpracovány v článku [4], který bude prezentován na konferenci EUC.

3 Systémové snižování příkonu

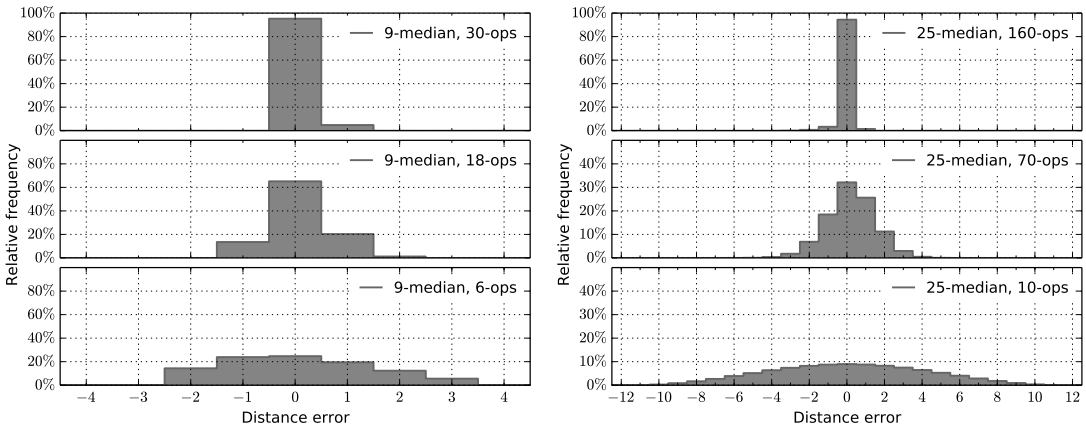
Dalším faktorem, kterým můžeme ovlivnit spotřebu, je úprava celkového chování systému. Ukazuje se, že v některých případech jsme ochotni snížit požadavky na funkčnost za cenu menší spotřeby a rychlejšího zpracování. Typickým příkladem jsou multimediální aplikace nebo jiné zpracování signálů. Celému přístupu se často říká *aproximační počítání*.

3.1 Návrh mediánového filtru

Ukázkou tohoto přístupu bude výpočet mediánu. Tato funkce byla aproximována pomocí kartézského genetického programování z optimálního řešení [9]. Při návrhu se používaly pouze bloky s funkcí *minimum* a *maximum*. Hlavním problémem návrhu je zjištění chyby aproximace. Ta se nejčastěji určuje

jako suma odchylek od plně funkčního řešení. Například pro algoritmu počítající medián z devíti osmibitových vstupů je potřeba k plnému ohodonocení $2^{8^9} = 2^{72}$ testů, což je neřešitelné. Proto se testy řeší náhodným výběrem vstupních vektorů. Potom je však chyba závislá na rozložení náhodného výběru.

Úkolem tedy bylo navrhnout metriku, která by správně vyhodnotila chybu. Vyjdeme z faktu, že díky použitému typu funkce je výsledek kandidátního řešení vždy jedním ze vstupních prvků. Další vlastností mediánu je to, že z množiny o velikosti $2n+1$ vybere právě $(i+1)$-tý nejmenší prvek. Proto při libovolné permutaci množiny $\{-n, -n+1, \ldots, 0, \ldots, n-1, n\}$ je validní medián 0. Hodnota, kterou aproximační medián vrátí při libovolné permutaci uvedené množiny, určuje odchylku pozice nejmenšího prvku. Tato metrika nám tedy určuje odchylku polohy mediánu a je nezávislá na vstupních datech. Výsledky pro různé aproximační mediány jsou vidět na obrázcích 4.



(a) Medián pro 9 vstupů

(b) Medián pro 25 vstupů

Obrázek 4: Histogram chyb v pozici mediánu pro jednotlivé aproximace. Hodnota OPS určuje počet operací *min / max* v aproximovaném řešení, kdy pro plně funkční 9 vstupový medián je potřeba 38 operací a pro 25 vstupů 220 operací.

3.2 Měření příkonu na mikroprocesorech

Dále bylo nutné prakticky ověřit, že použitím aproximace dochází ke zlepšení celkového příkonu. Proto byly aproximované mediány implementovány v mikroprocesorech. Jako ukázky jsem vybral osmibitový procesor akumulátorové architektury řady PIC16, dále šestnáctibitový procesor registrové architektury řady PIC24. Zástupce moderních procesorů řady ARM byl procesor STM32F4. Na těchto procesorech byl implementovaný medián a byl zkoumán vliv aproximace na spotřebu oproti plně funkčnímu řešení. Ukázalo se, že pro 9 vstupů a při povolení chyby o 1 pozici dochází k redukci příkonu o 21 %. U odchylky 2 pozic, kde správný medián nebyl určen v 35 % případů, je redukce příkonu 52 %. Podobně u 25-vstupého byl při odchylce o 5 pozic snížen příkon o 27 %. Kompletní výsledky jsou v článku na GECCO [5].

4 Cíl disertace

Předcházející kapitoly shrnují optimalizaci příkonu na tranzistorové úrovni a na úrovni aproximace funkce. Dalším místem optimalizace je úroveň hradel. Výsledky ve výzkumné skupině EHW@FIT ukazují, že evoluční algoritmy mají velký potenciál v optimalizaci obvodů na této úrovni. Zatím však jediným cílem optimalizace byla výsledná plocha na čipu. Ukazuje se však, že pro určení spotřeby je nutné obvod analyzovat detailněji. Na toto téma bylo publikováno mnoho literatury a možností odhadu

spotřeby je více. Celá disertační práce by měla ukázat, že evoluční metody návrhu digitálních obvodů jsou schopny efektivně optimalizovat spotřebu na různých úrovních — od tranzistorů, přes hradla, až po elementární softwarové metody. Kromě toho by měla nabídnout ucelený přehled metod, které je možné využít pro rychlý odhad příkonu.

5 Závěr

Práce se zabývá optimalizací spotřeby obvodů s využitím evolučních algoritmů. Byl představen nový přístup k návrhu obvodů na úrovni tranzistorů, který zlepšuje schopnost navrhnout obvody z jednotek na desítky tranzistorů. Tato metoda byla potom hardwarově akcelerována v čipu Xilinx Zynq. Byla také představena metoda odhadu příkonu těchto obvodů, se kterou jsme schopni optimalizovat spotřebu obvodů obsahující stovky tranzistorů. Práce také ukazuje možnosti snižování spotřeby na systémové úrovni, které jsou demonstrovány při aproximaci výpočtu mediánu. Ukázalo se, že při použití aproximací dochází k reálnému snížení spotřeby mikroprocesorů. Úspěšnost aproximace mediánu je vyjádřena novou metrikou založenou na pozici odchylky.

Poděkování Tato práce vznikla za podpory projektu FIT-S-14-2297 Architektury paralelních a vestavěných počítačových systémů.

Reference

[1] Miller, J. F.: *Cartesian Genetic Programming*. Springer Verlag, 2011.

[2] Mrazek, V.; Vasicek, Z.: Acceleration of transistor-level evolution using Xilinx Zynq Platform. In *Evolvable Systems (ICES), 2014 IEEE International Conference on*, Dec 2014, s. 9–16.

[3] Mrazek, V.; Vasicek, Z.: Evolutionary Design of Transistor Level Digital Circuits Using Discrete Simulation. In *Genetic Programming*, *LNCS*, ročník 9025, Springer, 2015, ISBN 978-3-319-16500-4, s. 66–77.

[4] Mrazek, V.; Vasicek, Z.: Automatic design of low-power arithmetic approximate VLSI circuits. In *13th IEEE International Conference on Embedded and Ubiquitous Computing, EUC*, 2015. V tisku., str. 8.

[5] Mrazek, V.; Vasicek, Z.; Sekanina, L.: Evolutionary Approximation of Software for Embedded Systems: Median Function. In *GECCO'15 Conference*, ACM, 2015, ISBN 978-1-4503-3488-4, s. 795–801.

[6] Shams, A.; Bayoumi, M.: A novel high-performance CMOS 1-bit full-adder cell. *IEEE Tr. on Circuits and Systems II: Analog and Digital Signal Processing*, ročník 47, č. 5, May 2000: s. 478–481, ISSN 1057-7130.

[7] Soudris, D.; Piguet, C.; Goutis, C.: *Designing CMOS Circuits for Low Power*. European low-power initiative for electronic system design, Springer, 2002, ISBN 9781402072345.

[8] Trefzer, M.: *Evolution of Transistor Circuits*. Dizertační práce, Ruprecht-Karls-Universität Heidelberg, 2006.

[9] Vasicek, Z.; Sekanina, L.: Evolutionary Approach to Approximate Digital Circuits Design. *IEEE Transactions on Evolutionary Computation*, ročník 19, č. 3, 2015: s. 432–444, ISSN 1089-778X.

[10] Venkatachalam, V.; Franz, M.: Power Reduction Techniques for Microprocessor Systems. *ACM Comput. Surv.*, ročník 37, č. 3, New York, NY, USA: ACM, Září 2005: s. 195–237, ISSN 0360-0300.

[11] Walker, J.; Hilder, J.; Tyrrell, A.: Towards evolving industry-feasible intrinsic variability tolerant CMOS designs. In *IEEE Congress on Evolutionary Computation*, 2009, s. 1591–1598.

[12] Weste, N. H.; Harris, D.: *CMOS VLSI design: a circuits and systemes perspective*. Boston, USA: Addison-Wesley, třetí vydání, 2005, ISBN 0-321-14901-7, 968 s.

[13] Zaloudek, L.; Sekanina, L.: Transistor-Level Evolution of Digital Circuits Using a Special Circuit Simulator. In *Evolvable Systems: From Biology to Hardware*, *LNCS*, ročník 5216, Springer Verlag, 2008, s. 320–331.