

NÁVRH A OPTIMALIZACE POLYMORFNÍCH OBVODŮ

Zbyšek Gajda

Informační technologie, 4. ročník, prezenční forma studia

Školitel: Lukáš Sekanina

Fakulta informačních technologií, Vysoké učení technické v Brně

Božetěchova 2, 612 66 Brno

gajda@fit.vutbr.cz

Abstrakt. Polymorfni obvody jsou takové obvody, které jsou složeny z běžných i tzv. polymorfních hradel. Polymorfni hradla mohou vykonávat různé logické funkce pro různé provozní podmínky jako např. různé pracovní teploty, hladiny napájecího napětí apod. Ukázalo se, že taktéž polymorfni obvody mohou realizovat různé funkce v závislosti na provozních podmínkách. Proto se předpokládá, že polymorfni obvody bude výhodné použít v inteligentních zařízeních, jejichž funkce se mění v závislosti na fyzickém prostředí. Článek popisuje strukturu budoucí disertační práce, která se bude zabývat jak evolučními algoritmy, tak i konvenčními algoritmy pro návrh polymorfních obvodů, a bude se zabývat též následnou optimalizací navržených obvodů.

Klíčová slova. Polymorfni obvod, evoluční návrh, evoluční optimalizace, kartézské genetické programování, binární rozhodovací diagramy

1 Úvod

Tématem disertační práce je vytvoření metod návrhu pro polymorfni číslicové obvody. Tyto obvody jsou sestaveny jak z běžných, tak i z tzv. polymorfních hradel, která umožňují realizovat různé logické funkce za různých operačních podmínek. Úkolem je navrhnout, implementovat a ověřit metody návrhu polymorfních číslicových obvodů, které budou vycházet jak z tradičních návrhových technik, tak z metod evolučního designu. Navržené metody je pak třeba vzájemně porovnat a ověřit na vhodných aplikacích.

Jelikož je autor studentem posledního ročníku prezenčního studia, je třeba se věnovat zejména sepsání disertační práce, a proto je podstatná část tohoto článku zaměřena na strukturu disertační práce.

2 Struktura disertační práce

Ke každé části disertace je uveden stručný text, o čem daná část disertace bude. Hlubší vhled do problematiky spojené s tématem disertace poskytují odkazy na publikace jak autora, tak i na další.

2.1 Úvod

První články o polymorfni elektronice se objevily v roce 2001 [1]. Od té doby se hledají vhodné aplikace polymorfni elektroniky. Jako slibné oblasti aplikace se jeví: rychlá a levná rekonfigurace, inteligentní senzory, skryté funkce a diagnostika a testování.

V době (r. 2005) přebírání tématu disertační práce byly navrženy nepříliš složité číslicové obvody do pouze 5 vstupů [2]. Cílem mé práce je definovat metody pro návrh komplexních polymorfních obvodů, které budou navíc optimalizované na počet logických členů.

2.2 Návrh a syntéza digitálních obvodů

Za kombinační obvod se dá považovat realizace (abstraktní nebo konkrétní) funkce definované nad Booleovou binární algebrou. Při návrhu kombinačních obvodů se dá uvažovat o dvou pohledech (filozofiích): univerzální a aplikačně specifický. Univerzální metody se uplatňují především při návrhu konečných automatů; aplikačně specifické metody se uplatňují při návrhu zejména funkčních jednotek jako např. kombinační násobička s uchováním přenosu [3]. Pohledy se liší jinou mírou abstrakce. Zatímco univerzální popis pracuje s elementárními prvky jako binární operace, tak aplikačně specifický pohled pracuje s komplexnějšími prvky jako je např. úplná sčítačka, CLA (Carry-Look-Ahead) jednotka. V práci se zaměřím především na metody univerzálního návrhu. Do dnešní doby bylo vyvinuto několik univerzálních metod pro návrh obvodů. Vstupem návrhu je funkce definovaná pravdivostní tabulkou. Metody se liší jak ve způsobu optimalizace, tak v zaměření na cílovou platformu. Nejznámějšími metodami jsou např. Karnaughovy mapy, Quine-McCluskey, Espresso [4]. Významná je i reprezentace kombinačních obvodů pomocí binárních rozhodovacích diagramů (BDD), jejich optimalizace a syntéza. Kromě samotných metod metod existují i systémy, které implementují různé metody návrhu a syntézy obvodů jako SIS [5], ABC System [6] apod.

2.3 Polymorfní elektronika

V této části budeme definovat pojmy z oblasti polymorfní elektroniky a definovat problém syntézy. Neméně důležité je připravit sadu benchmarkových testů, se kterými se budou metody návrhu srovnávat.

Polymorfní hradla

Polymorfní hradlo se od běžného hradla liší tím, že mění svoji funkci, a to v závislosti na okolním prostředí [7]. Ovlivňující veličinou může být např. napájecí napětí, teplota, světlo. Příkladem může být hradlo NOR/NAND, které při napájecím napětí 5V realizuje funkci NOR (tzv. první mód) a při napětí 3,3V realizuje NAND (tzv. druhý mód).

Polymorfní obvody

Polymorfní obvody jsou složeny jak z běžných hradel, tak z hradel polymorfních. Obvody mění svoji funkci podobně jako polymorfní hradla v závislosti na okolním prostředí. Polymorfní obvody lze rozdělit na dvě skupiny [8]: (1) obvody realizující více logických funkcí závislých na okolním prostředí; a (2) obvody implementující testovací a diagnostické prvky. Obvody (1) mohou najít uplatnění v např. polymorfním FIR filtru [9], kdy za normálních podmínek funguje podle očekávané charakteristiky, kdežto při poklesu napětí dojde k odpojení části filtru a zbylá část je napětím rekonfigurována tak, aby charakteristika co nejvíce odpovídala té původní. Obvody (2) se mohou uplatnit při samočinné detekci poruchy [10] nebo pro snížení počtu testovacích vektorů [11]. V práci se zaměřím především na obvody ze skupiny (1).

Pro vývoj/testování fyzické realizace polymorfních obvodů slouží platforma REPOMO (Reconfigurable Polymorphic Module) [12].

Formulace problému syntézy

Funkce f_1 a f_2 jsou požadované logické funkce v prvním a druhém módu polymorfních hradel. Necht' Γ je množina konvenčních a polymorfních hradel. Funkce hradel se mění v závislosti na módu j . Poly-

morfní obvod lze modelovat grafem $G = (V, E, \varphi)$, kde V jsou vrcholy, $E = \{(a, b) | a, b \in V\}$ jsou hrany, $\varphi : V \rightarrow \Gamma$ je funkce, která přiřazuje vrcholům grafu hradla. Formulace problému syntézy zní: Najít takový graf G , který bude realizovat funkce f_1 (pro *mód* $j = 1$) a f_2 (pro *mód* $j = 2$). Dále mohou být specifikovány další požadavky jako minimalizace zpoždění, plochy apod. Naneštěstí tento problém nemůže být vyřešen dostupnými konvenčními metodami, neboť neumožňují pracovat s polymorfními vlastnostmi obvodu.

Sada benchmarků

Pro porovnání metod návrhu bylo třeba vytvořit sadu benchmarkových obvodů. Jako benchmarky byly zvoleny dvojice funkcí tak, aby měly stejný počet vstupů a výstupů. Těmito dvojicemi jsou Násobička/Radicí síť, Majorita/Parita a xKonstanta/xKonstanta (násobení pro každý *mód* j jinou konstantou) pro různé počty vstupů/výstupů.

2.4 Evoluční návrh a optimalizace

Evoluční návrh má tu vlastnost, že evoluční algoritmus nemusí znát povahu vzájemné interakce členů obvodu. Stačí definovat požadovanou funkci zpravidla pravdivostní tabulkou. Bylo prokázáno, že evoluce může navrhovat obvody, které mají lepší vlastnosti než obvody navržené konvenčně [13]. Tzv. fitness funkce použitá v evolučním algoritmu pak zpravidla počítá počet nekorektních bitů navrženého obvodu vůči definované pravdivostní tabulce.

Jelikož u evolučního návrhu nemusíme nic znát o interakci členů obvodu, jeví se tato metoda jako vhodná pro návrh polymorfních obvodů.

Přímý evoluční návrh

Pro evoluční návrh univerzálních obvodů, lze použít zejména genetické programování nebo kartézské genetické programování (Cartesian Genetic Programming, CGP) [14]. Zkušenosti ukazují, že vhodnější metodou je CGP, které umožňuje při návrhu větší komplexnost řešení a flexibilitu.

Pomocí CGP byly navrženy např. nejmenší známé násobičky [13]. Tato metoda se také ukázala účinná při našem návrhu polymorfních obvodů [15, 16, 8]. Dále byla během našeho výzkumu použita pro návrh sčítaček a majoritních obvodů [17], které jsou sestaveny i z nekonvenčních hradel.

Hlavní nevýhodou přímého návrhu je problém škálovatelnosti [18].

Evoluční optimalizace

Při evolučním návrhu začíná CGP obvykle s náhodně generovanou počáteční populací. Při evoluční optimalizaci začínáme s plně funkčním obvodem a cílem je redukovat např. počet hradel. Součástí evolučního návrhu bývá i následná optimalizace. Optimalizace se od návrhu liší především v podobě fitness funkce, která obvykle kromě zjištění správné funkčnosti obvodu navíc pokrývá i jiné vlastnosti obvodu jako např. počet hradel obvodu, zpoždění, spotřeba apod.

Převážně se pro optimalizace používají různé variace fitness funkce $fit1$, která zjišťuje plnou funkčnost obvodu a minimalizuje počet hradel. Pro optimalizaci na počet hradel jsme nově použili fitness funkci $fit2$, která pouze zjišťuje, zda je kandidát plně funkční či nikoliv. Ukázalo se, že zdánlivě "nesmyslná" funkce $fit2$ paradoxně optimalizuje, je odolná vůči uváznutí v lokálních extrémech a dosahuje převážně lepších výsledků než $fit1$ [19].

Zajímavou vlastností evoluční optimalizace je, že může optimalizovat jak obvody získané evolucí tak konvenčním návrhem. Na tomto poli jsme v rámci výzkumu úspěšně optimalizovali některá konvenčně navržená řešení¹.

¹V současnosti se výsledky tohoto výzkumu zpracovávají.

Hranice škálovatelnosti

Optimalizace bývá dokončena v určitém čase, závislém na době vyhodnocení jedince, velikosti populace, počtu generací. Evoluční návrh nemusí dospět k řešení v reálném čase, neboť prohledávací prostor může být příliš velký (nebo naopak příliš malý) nebo může uvíznout v lokálním extrému [18]. Hranice škálovatelnosti je určena např. počtem vstupů obvodu a říká nám, že od této hranice je velmi obtížné až nemožné najít řešení v reálném čase. Z důvodu potíží se škálovatelností je proto kladen důraz na to, aby bylo možno navrhovat libovolně velké obvody.

Hranice škálovatelnosti se dá částečně kompenzovat variantou inkrementální evoluce [20], která rozkládá problém na podmnožinu problémů [21].

2.5 Použití konvenčního návrhu a evoluční optimalizace

Využití konvenčního přístupu při návrhu polymorfních obvodů skýtá možnost navrhovat libovolně velké polymorfní obvody. Slabinou takto navržených obvodů by mohla být jejich velikost, proto je vhodné takové obvody případně optimalizovat evolučním algoritmem.

Polymorfní multiplexování

Myšlenka návrhu polymorfního multiplexování [2, 21] tkví v tom, že požadované funkce f_1 a f_2 jsou navrženy běžnými metodami. Výstupy navržených obvodů jsou pak připojeny na vstupy tzv. polymorfních multiplexorů. Ty přenášejí signál dle prostředí, buď z jedné anebo ze druhé realizace funkcí.

Realizace funkcí mohou být navrženy odděleně nebo společně. Při oddělené realizaci mohou být funkce navrženy z nejlepších známých řešení, navrženy aplikačně specificky nebo univerzálně. Pro společnou realizaci funkcí do jednoho obvodu se nabízí univerzální návrh (Espresso, ABC System, SIS).

Polymorfní BDD

Dalším způsobem, jak integrovat polymorfní hradlo do standardního algoritmu syntézy, je využít BDD. Principem vytvoření a syntézy polymorfních BDD [21] je transformace pravdivostní tabulky a syntéza MTBDD (Multi-Terminal Binary Decision Diagram) [22] na polymorfní obvod. Funkce obvodu jsou zapsány do pravdivostní tabulky, ta je pomocí bijektivního zobrazení převedena na transformovanou tabulku. Takto transformovaná tabulka slouží jako podklad pro vytvoření (optimalizovaného) MTBDD. Vytvořený BDD je pak transformován na realizaci obvodu, kdy jsou uzly převedeny na multiplexory a terminály na elementární polymorfní obvody za pomoci převodní tabulky [19], která byla navržena použitím evolučního návrhu (část 2.4).

Optimalizace

Jelikož konvenčně navržené obvody mají polymorfní hradla umístěna, buď u vstupů obvodu (BDD) anebo u výstupů (multiplexování), je proto nanejvýš vhodná optimalizace, která má za cíl umístění polymorfních hradel do všech částí obvodu a zároveň zmenšení počtu logických členů.

Optimalizaci předchází konverze implementací navržených konvenčními metodami, kdy jsou více-vstupové členy převedeny na dvoustupové logické členy. Takto konvertovaný obvod lze pak použít jako jedince počáteční populace evolučního algoritmu.

2.6 Experimentální vyhodnocení

Z výsledků [21, 19] je patrné, že není možné jednoznačně určit, která metoda návrhu je nejlepší (snad s výjimkou Espresso, které se jeví jako nevhodné). Např. pro návrh obvodu Násobička/Řadicí síť se jeví jako vhodné použít polymorfní multiplexování s využitím ABC následnou optimalizací; pro návrh

obvodů typu xKonstanta/xKonstanta je nejuvhodnější obvod navržen složením dvou nezávislých modulů polymorfním multiplexováním opět nasledovaným optimalizací. U návrhu Majorita/Parita jednoznačně nelze říct, která metoda je lepší.

2.7 Závěr

Byly srovnány různé přístupy návrhu/syntézy polymorfních obvodů. Ze srovnání plyne, že přímá evoluce s CGP není škálovatelná. Dále plyne, že modifikované konvenční metody mohou navrhovat libovolně velké (co do počtu vstupů, výstupů) realizace polymorfních obvodů. Takové realizace musejí být však optimalizovány. Tím lze získat velmi kompaktní obvody. Nevýhodou je, že CGP je pro velké obvody pomalé, avšak čím více času je dáno optimalizaci, tím lepších výsledků dosáhne.

3 Závěr

V článku byla uvedena struktura předpokládané disertační práce. Konkrétní výsledky výzkumu jsou uvedeny ve člancích, které byly v textu citovány nebo v člancích probíhajícím recenzním řízením. Výsledky však budou prezentovány na semináři PAD 2009.

Přínos své práce spatřuji ve vytvoření metodiky pro návrh polymorfních obvodů s využitím konvenčních metod návrhu, především v aplikaci BDD při návrhu a syntéze polymorfních obvodů. Dalším přínosem své práce spatřuji v užití fitness funkce *fit2* při evoluční optimalizaci (polymorfních i konvenčních) obvodů, která prokazatelně dosahuje (až na výjimky) lepších výsledků než běžně používaná *fit1*.

Poděkování

Výzkum je prováděn s podporou Grantové agentury České republiky v rámci projektu Návrh a obvodová realizace zařízení pro automatické generování patentovatelných invencí č. GA102/07/0850 a Ministerstva školství, mládeže a tělovýchovy v rámci výzkumného záměru Výzkum informačních technologií z hlediska bezpečnosti č. MSM0021630528.

Reference

- [1] Adrian Stoica, Ricardo Salem Zebulum, and Didier Keymeulen. Polymorphic electronics. In *Proc. of Evolvable Systems: From Biology to Hardware Conference*, volume 2210 of *LNCS*, pages 291–302. Springer, 2001.
- [2] Lukáš Sekanina. Design methods for polymorphic digital circuits. In *Proc. of 8th IEEE Design and Diagnostic of Electronic Circuits and Systems Workshop*, pages 145–150. University of West Hungary, 2005.
- [3] John F. Wakerly. *Digital Design: principles and practices – 3d edition*. Prentice Hall, New Jersey, US, 2000.
- [4] R. K. Brayton et al. *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Academic Publishers, Boston, MA, USA, 1984.
- [5] Ellen M. Sentovich, Kanwar Jit Singh, Luciano Lavagno, Cho Moon, Rajeev Murgai, Alexander Saldanha, Hamid Savoj, Paul R. Stephan, Robert K. Brayton, and Alberto Sangiovanni-Vincentelli. Sis: A system for sequential circuit synthesis. Technical report, University California, Berkeley, 1992.
- [6] Berkley Logic Synthesis and Verification Group. *ABC: A System for Sequential Synthesis and verification*. Dostupné z url: <http://www.eecs.berkeley.edu/~alanmi/abc/>.
- [7] Lukáš Stareček, Lukáš Sekanina, Zbyšek Gajda, Zdeněk Kotásek, Roman Prokop, and Vladislav Musil. On properties and utilization of some polymorphic gates. In *6th Electronic Circuits and Systems Conference (ECS 2007)*, pages 77–81. Faculty of Informatics and Information Technology STU, 2007.

- [8] Lukáš Sekanina, Lukáš Stareček, Zdeněk Kotásek, and Zbyšek Gajda. Polymorphic gates in design and test of digital circuits. *International Journal of Unconventional Computing*, 4(2):125–142, 2008.
- [9] Lukáš Sekanina, Richard Růžička, and Zbyšek Gajda. Polymorphic fir filters with backup mode enabling power savings. In *Proc. of the 2009 NASA/ESA Conference on Adaptive Hardware and Systems*, pages 43–50. IEEE Computer Society, 2009.
- [10] Lukáš Sekanina. Evolution of polymorphic self-checking circuits. *Lecture Notes in Computer Science*, 2007(4684):186–197, 2007.
- [11] Lukáš Stareček, Lukáš Sekanina, and Zdeněk Kotásek. Reduction of test vectors volume by means of gate-level reconfiguration. In *Proc. of 2008 IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop*, pages 255–258. IEEE Computer Society, 2008.
- [12] Lukáš Sekanina, Richard Růžička, Zdeněk Vašíček, Roman Prokop, and Lukáš Fojcik. Repomo32 - new reconfigurable polymorphic integrated circuit for adaptive hardware. In *Proc. of the 2009 IEEE Symposium Series on Computational Intelligence - Workshop on Evolvable and Adaptive Hardware*, pages 39–46. IEEE Computational Intelligence Society, 2009.
- [13] Vesselin K. Vassilev, Dominic Job, and Julian F. Miller. Towards the automatic design of more efficient digital circuits. In *The Second NASA/DoD Workshop on Evolvable Hardware*. Palo Alto, CA, USA, 2000.
- [14] Julian F. Miller, Dominic Job, and Vesselin K. Vassilev. Principles in the Evolutionary Design of Digital Circuits – Part I. *Genetic Programming and Evolvable Machines*, 1(1):8–35, 2000.
- [15] Lukáš Sekanina, Lukáš Stareček, Zbyšek Gajda, and Zdeněk Kotásek. Evolution of multifunctional combinational modules controlled by the power supply voltage. In *Proc. of the 1st NASA/ESA Conference on Adaptive Hardware and Systems*, pages 186–193. IEEE Computer Society, 2006.
- [16] Lukáš Sekanina, Tomáš Martínek, and Zbyšek Gajda. Extrinsic and intrinsic evolution of multifunctional combinational modules. In *2006 IEEE World Congress on Computational Intelligence*, pages 9676–9683. IEEE Computational Intelligence Society, 2006.
- [17] Zbyšek Gajda and Lukáš Sekanina. Reducing the number of transistors in digital circuits using gate-level evolutionary design. In *2007 Genetic and Evolutionary Computation Conference*, pages 245–252. Association for Computing Machinery, 2007.
- [18] Vesselin K. Vassilev and Julian F. Miller. Scalability problems of digital circuit evolution. In *Proc. of the 2000 NASA/DoD Conference on Evolvable Hardware*, pages 55–64, Palo Alto, CA, 2000. IEEE Computer Society.
- [19] Zbyšek Gajda and Lukáš Sekanina. On synthesis of compact polymorphic combinational circuits. *IEEE Transactions On Computer-Aided Design of Integrated Circuits And Systems*, zasláno k recenzi.
- [20] Tatiana Kalganova. Bidirectional incremental evolution in extrinsic evolvable hardware. In *Proc. of the 2nd NASA/DoD Workshop on Evolvable Hardware*, pages 65–74. IEEE Computer Society, Silicon Valley, USA, July 2000.
- [21] Zbyšek Gajda and Lukáš Sekanina. Gate-level optimization of polymorphic circuits using cartesian genetic programming. In *Proc. of 2009 IEEE Congress on Evolutionary Computation*, pages 1599–1604. IEEE Computational Intelligence Society, 2009.
- [22] Rolf Drechsler and Bernd Becker. *Binary Decision Diagrams: Theory and Implementation*. Kluwer Academic Publishers, Boston, USA, 1998.