

TESTOVANIE ASYNCHRÓNNYCH DIGITÁLNYCH OBVODOV

Roland Dobai

Aplikovaná informatika, 2. ročník (3 semestre), denná prezenčná forma štúdia
Školiteľka: Elena Gramatová

Ústav informatiky, Slovenská akadémia vied
Dúbravská cesta 9, 845 07 Bratislava 45, Slovensko

{roland.dobai, elena.gramatova}@savba.sk

Abstrakt. Asynchrónne obvody sa vyznačujú s oveľa lepšími vlastnosťami ako synchronne obvody z hľadiska výkonu, elektrickej spotreby a znovupoužitelnosti. Nutnou podmienkou rozšírenia asynchrónnych obvodov je vývoj efektívnejších metód ich testovania. Cieľom tohto projektu je prispieť k zlepšeniu testovania asynchrónnych obvodov. Pre rýchlostne nezávislé asynchrónne obvody bol navrhnutý a implementovaný deduktívny poruchový simulátor trvalých porúch. Bol implementovaný aj sériový poruchový simulátor z dôvodu porovnania výsledkov v tom istom výpočtovom prostredí. Navrhnutým deduktívnym poruchovým simulátorom sa znížil čas simulácie o 60% až 80%, pričom pamäťové nároky sa zvýšili maximálne iba o 14%. Cieľom ďalšej práce bude prispieť k zlepšeniu testovateľnosti porúch oneskorení v asynchrónnych obvodoch.

Kľúčové slová. Asynchrónny obvod, rýchlostne nezávislý obvod, deduktívny poruchový simulátor, trvalé poruchy.

1 Úvod

Väčšina sekvenčných obvodov je synchronna. Dôvodom toho je ich relatívne jednoduchý návrh a dostupnosť profesionálnych softvérových nástrojov pre ich návrh [1]. Dnešné požiadavky na výkonnosť digitálnych systémov sú tak vysoké, že vznikajú nové problémy v prípade používania synchronných obvodov, napríklad vysoká úroveň elektromagnetického vyžarovania a problém distribuovania hodinového signálu. Na odstránenie týchto problémov môže byť vhodné použiť asynchrónne obvody [1, 2]. Aby bolo možné nahradiť v budúcich digitálnych systémoch synchronne obvody asynchrónnymi, je potrebné odstrániť niekoľko problémov súvisiace s ich aplikáciou. Sú nevyhnutné softvérové návrhové prostriedky na podporu asynchrónneho návrhu (prostriedky CAD — *Computer-Aided Design*) ako aj vyvinutie a overenie metód testovania asynchrónnych obvodov [3, 4].

Asynchrónne obvody je možné rozdeliť podľa modelu oneskorenia do troch skupín [5]: (1) obvody so samočinným časovaním (*self-timed*), (2) rýchlostne nezávislé obvody (*speed-independent*) [6] a (3) obvody odolné proti oneskoreniu (DI — *delay-insensitive*). Podskupinou obvodov DI sú obvody pseudo-odolné proti oneskoreniu (QDI — *quasi-delay-insensitive*).

Rýchlostne nezávislý obvod pracuje správne v prípade ľubovoľných, neohraničených oneskorení (oneskorenie 0 až ∞) na logických členoch a pri nulových oneskoreniach na vodičoch. Obvod je obvod typu DI, ak pracuje správne v prípade ľubovoľných, neohraničených oneskorení na logických členoch a na vodičoch. Obvod DI s izochrónnymi rozvetveniami je obvod typu QDI. Rozvetvenie je izochrónne,

ak oneskorenia na každej vetve po rozvetvení sú rovnaké. Všetky ostatné obvody, ktoré nemožno zaradiť do predchádzajúcich skupín a pritom fungujú správne v prípade dodržania určitých časovacích pravidiel, sú obvody so samočinným časovaním.

Obvody DI sú robustné proti poruchám oneskorenia, t.j. poruchy oneskorenia u nich môžu spôsobovať iba degradáciu výkonnosti. Dnešné vysoko rýchlostné asynchrónne obvody nie sú naďalej robustné, preto poruchy oneskorenia u nich môžu spôsobovať aj nefunkčnosť obvodu [7].

Rýchlostne nezávislé obvody sa navrhujú napr. pomocou nástroja Petrify [8] podľa grafových špecifikácií (STG — *signal transition graph*). Špecifikácia STG je používaná na zadefinovanie správania sa obvodu.

Najpoužívanejším pamäťovým členom asynchrónnych obvodov je logický člen C. Ak na všetkých jeho vstupoch sú logické jednotky (nuly), tak hodnota na výstupe sa zmení na logickú jednotku (nulu). Pre všetky ostatné kombinácie vstupov výstup ostane nezmenený [5].

Počas fungovania asynchrónnych obvodov môžu vzniknúť nepríjemné dočasné zmeny hodnôt signálov, ktoré sa nazývajú hazardy. Hazardy môžu byť statické a dynamické. V prítomnosti statických hazardov signály, ktoré by mali mať stabilné hodnoty, dočasne zmenia svoju logickú hodnotu na opačnú. V prítomnosti dynamických hazardov na signáli, kde by sa mala uskutočniť iba jedna zmena v hodnote signálu, v skutočnosti dôjde k viacerým dočasným zmenám [9]. Petrify umožňuje generovanie obvodov so zložitými logickými členmi na zníženie počtu vzniknutia hazardov.

V časti 2 sú vymenované publikácie z oblasti testovania asynchrónnych obvodov. V časti 3 sú navrhnuté ciele dizertačnej práce. Časť 4 a časť 5 obsahujú dosiahnuté výsledky a plány do budúcnosti.

2 Súčasný stav v testovaní asynchrónnych obvodov

Sériový poruchový simulátor pre rýchlostne nezávislé obvody (SPIN-SIM) bol navrhnutý a publikovaný v [10], ktorý spolupracuje s automatickým generátorom testovacích vektorov (ATPG) SPIN-TEST [11]. SPIN-TEST generuje test pre trvalé poruchy. Poruchová simulácia asynchrónnych obvodov je zložitejším problémom ako poruchová simulácia synchronných obvodov, lebo musí sa zaoberať s hazardmi, osciláciami a kritickými súčasnými zmenami signálov. SPIN-SIM používa 13-hodnotovú logiku [12] a časové pečiatky [10] na detekovanie hazardov. SPIN-SIM transformuje asynchrónny obvod na kombinačný v dvoch krokoch: (1) Všetky logické členy C sú nahradené množinou pseudo-logických členov. Pseudo-logický člen je logický člen, ktorý neoneskoruje signál. Táto množina je zostavená tak, aby bola bezhazardná. (2) Všetky spätné väzby sú odstránené transformáciou na pseudo-primárne vstupy a výstupy, aby na obvod mohla byť aplikovaná iteračná metóda sekvenčných obvodov [13].

SPIN-TEST používa automatický generátor testovacích vektorov ATALANTA na vygenerovanie testu pre trvalé poruchy [14]. Prehľadávací algoritmus A* [15] je použitý na nájdenie najlepšej postupnosti testovacích vektorov k dosiahnutiu stavu potrebného na aplikovanie jednotlivých vygenerovaných vektorov. SPIN-TEST používa jednoduchú heuristickú funkciu k odhadnutiu náročnosti aktivácie poruchy a šírenie poruchy na primárne výstupy.

SPIN-SIM aplikuje sériovú metódu poruchovej simulácie, preto simulačný čas je možné skrátiť aplikovaním deduktívnej alebo súbežnej metódy. Metódy deduktívnej alebo súbežnej poruchovej simulácie nie sú nové, iba doteraz neboli aplikované na asynchrónne obvody. Ich výhodou je, že pokrytie porúch v obvode pre jeden vstupný testovací vektor sa získa jedným simulačným prechodom, pri ktorom sa šíria zoznamy porúch postupne cez všetky logické členy na primárny výstup podľa definovaných pravidiel [13]. To je výhoda oproti sériovej metóde, ktorá využíva injekciu porúch do štruktúry obvodu, teda simulácia pre jeden testovací vektor musí prebehnúť toľkokrát, koľko porúch uvažujeme pokryť v obvode.

Pravidlá šírenia zoznamu porúch sú známe pre jednoduché logické členy, logické členy s ľubovoľným počtom vstupov [16], kombinačné bloky opísané jazykom FDL (*function definition language*) [17] a sekvenčné bloky opísané Moorovým automatom [18]. Pre zložité logické členy rýchlostne nezávislých

asynchrónnych obvodov zatiaľ takéto pravidlá neboli definované.

Fsimac je ďalším sériovým simulátorom pre asynchrónne obvody, ktorý taktiež používa 13-hodnotovú logiku. Tento simulátor je možný používať iba pre špeciálnu podtriedu asynchrónnych obvodov, pre Huffmanove obvody [19].

Výhodou používania modelu rýchlostne nezávislých obvodov je možnosť reprezentácie obvodov ostatných tried asynchrónnych obvodov pod týmto modelom. Nástroj pre rýchlostne nezávislé obvody je možné použiť na ľubovoľnú triedu asynchrónnych obvodov [4].

3 Ciele dizertačnej práce

V súčasnosti najviac používané obvody nie sú typu DI, preto poruchy oneskorení u nich môžu spôsobovať aj chybnú funkciu. Tento smer vývoja si vyžaduje, aby sa zaoberalo podrobnejšie s problematikou porúch oneskorení asynchrónnych obvodov. Značné množstvo výskumu existuje pre generovanie testovacích vektorov a návrh pre testovateľnosť so zameraním sa na trvalé poruchy, avšak iba niekoľko na poruchy oneskorení [20].

Preto cieľom dizertačnej práce bude zaoberať sa s problémami testovania porúch oneskorení v asynchrónnych obvodoch. Snahou bude prispieť k metódam generovania testovacích vektorov pre poruchy oneskorení. Existujúce metódy generovania testu pre model trvalých porúch budú analyzované z hľadiska rozšíriteľnosti pre testovanie porúch oneskorení. Metódy návrhu pre testovateľnosť, ktoré boli navrhnuté na podporu testovania trvalých porúch asynchrónnych obvodov budú taktiež modifikované na zvýšenie testovateľnosti porúch oneskorení.

Z analýzy problémov testovateľnosti asynchrónnych obvodov boli definované ciele dizertačnej práce, ktoré by mali prispieť k zlepšeniu testovateľnosti asynchrónnych obvodov a sú zamerané na algoritmy generovania deterministického testu pre poruchy oneskorení pre väčšinu typov asynchrónnych obvodov a návrh vhodnej architektúry SCAN pre ich aplikáciu. Dosiaľ boli navrhnuté iba podporné algoritmy generovania testov a poruchovej simulácie pre model trvalých porúch, ktoré budú základom pre testy porúch oneskorení.

4 Dosiahnuté výsledky

Súčasný stav problematiky testovateľnosti porúch oneskorení v asynchrónnych obvodoch bol spracovaný v [20]. Boli analyzované všetky existujúce metódy zamerané na poruchy oneskorení asynchrónnych obvodov a bola zdôvodnená nutnosť hľadať vylepšenia v existujúcich metódach.

Model trvalých porúch je ten istý ako pre kombinačné a synchronne sekvenčné logické obvody, teda testy sú generované pre poruchy trvalej nuly a poruchy trvalej jednotky na všetkých vodičoch v obvode. Ďalej je používané len označenie — trvalé poruchy.

Ako základ pre ďalšiu prácu s poruchami oneskorení bol implementovaný sériový poruchový simulátor a bol navrhnutý a implementovaný deduktívny poruchový simulátor pre trvalé poruchy v rýchlostne nezávislých asynchrónnych obvodoch [21, 22, 23]. Testovacie vektory pre trvalé poruchy môžu byť použité neskôršie na zostavenie testu pre poruchy oneskorení.

Na otestovanie poruchových simulátorov bol implementovaný jednoduchý náhodný generátor testovacích vektorov, ktorý nuluje obvod s pravdepodobnosťou vypočítanou podľa počtu hazardov v obvode. Vzorec na výpočet tejto pravdepodobnosti bol podstatne zjednodušený v porovnaní s dosiaľ používaným výpočtom, čím sa znížil výpočtový čas bez degradácie pokrytia porúch [22].

Sériový poruchový simulátor bol implementovaný podľa publikovaného simulátora SPIN-SIM. Ich porovnania sú uvedené neskôršie. Dôvodom implementácie bolo, aby jeho výkonnosť mohla byť porovnaná s navrhnutým deduktívnym poruchovým simulátorom v tom istom výpočtovom prostredí. Výsledky týchto dvoch poruchových simulátorov boli porovnané na množine rýchlostne nezávislých skúšobných obvodov. Tieto obvody obsahovali okrem logických členov C aj preklápacie obvody D. Pre preklápacie

obvody D neexistovala náhradná bezhazardná množina pseudo-logických členov, preto bola navrhnutá a publikovaná v [22].

Návrh deduktívneho poruchového simulátora bolo umožnené implementovaním algoritmu na prešírenie zoznamu porúch cez zložité logické členy asynchrónnych obvodov. Navrhnutý algoritmus spracuje Boolovskú funkciu, ktorá reprezentuje funkciu zložitého logického člena. Zoznamy porúch z jednotlivých vstupov ukladá do dočasných vnútorných zoznamov, pomocou ktorých vygeneruje zoznam porúch pre výstup logického člena. Deduktívny poruchový simulátor obsahuje niekoľko implementačných zlepšení pomocou ktorých simulačný čas bol ďalej znížený (napr. reprezentácia porúch, generovanie týchto reprezentácií, efektívna práca so zoznamami). Deduktívny poruchový simulátor používa 13-hodnotovú logiku a časovacie pečiatky na detekovanie hazardov, oscilácií a súčasných zmien signálov [23].

Poruchové simulátory boli implementované v jazyku C++. Testovanie bolo vykonané na osobnom počítači (s procesorom AMD Athlon™64 X2 Dual Core 4400+ a s 2 GB pamäťou) pod OS Linux 2.6.24.

Tabuľka 1 obsahuje porovnanie implementovaných poruchových simulátorov z hľadiska pokrytia trvalých porúch. Výsledky poruchového simulátora SPIN-SIM sa nachádzajú v 4. stĺpci tabuľky (ako sú uvedené v [10]). V 3. stĺpci sú výsledky predchodcu SPIN-SIMu. 5. riadok tabuľky obsahuje výsledky, ktoré boli dosiahnuté poruchovým simulátorom implementovaným podľa publikovaného SPIN-SIM. Pokrytia trvalých porúch nie sú úplne rovnaké, lebo skúšobné rýchlostne nezávislé obvody neobsahujú taký istý počet logických členov, lebo boli znova syntetizované. 6. stĺpec tabuľky obsahuje výsledky navrhnutého a implementovaného deduktívneho poruchového simulátora. Výsledky v tabuľke pre implementované poruchové simulátory boli dosiahnuté s počtom náhodne generovaných vektorov 200. Možno konštatovať, že pokrytie porúch pre implementované simulátory sú rovnaké a v najhoršom prípade pokrytie trvalých porúch je 94.12%.

Tabuľka 2 porovnáva poruchové simulátory trvalých porúch z hľadiska potrebného výpočtového času a pamäte. Výpočtové časy sa nachádzajú pre sériový poruchový simulátor v 2. stĺpci a pre deduktívny v 3. stĺpci tabuľky. Vo väčšine prípadov deduktívny poruchový simulátor je o 60% – 80% rýchlejší. Pre niektoré obvody (*chu133*, *converta* a *half*), sériový poruchový simulátor je rýchlejší, lebo poruchy, ktoré boli detekované predchádzajúcimi testovacími vektormi, sériový poruchový simulátor už nesimuluje, kým deduktívny poruchový simulátor pracuje vždy so všetkými poruchami. Tento problém je možné odstrániť použitím deterministického generátora testovacích vektorov.

Pamäťové požiadavky sa nachádzajú pre sériový poruchový simulátor v 5. stĺpci a pre deduktívny v 6. stĺpci tabuľky 2. Pamäťové nároky v niektorých prípadoch (*chu150* a *nak_pa*) pre deduktívny poruchový simulátor sú nižšie ako pre sériový. Táto nepresnosť merania môže byť spôsobená napríklad krátkymi výpočtovými časmi, alebo veľkosťou testovacích obvodov (malé obvody, málo porúch a malé zoznamy porúch). Zvýšenie pamäťových nárokov pre deduktívny poruchový simulátor je v každom prípade pod 14%.

5 Záver

Pre rýchlostne nezávislé asynchrónne obvody bol navrhnutý a implementovaný deduktívny poruchový simulátor trvalých porúch. Bol implementovaný aj sériový poruchový simulátor z dôvodu porovnania výsledkov v tom istom výpočtovom prostredí. Navrhnutým deduktívnym poruchovým simulátorom sa znížil čas simulácie o 60% až 80%, pričom pamäťové nároky sa zvýšili maximálne iba o 14%. V niektorých prípadoch deduktívny poruchový simulátor bol pomalší ako sériový. Tento problém bude odstránený návrhom a implementáciou deterministického generátora testu.

Deduktívna metóda je efektívna aj pre asynchrónne obvody s porovnaním so sériovou poruchovou simuláciou a ani časové pečiatky nespôsobili spomalenie procesu poruchovej simulácie. Ukázalo sa, že veľkosť pamäte sa podstatne nezvýšila, čo je teda veľkou výhodou použitia deduktívnej metódy pre poruchovú simuláciu aj pre asynchrónne obvody.

Ďalším cieľom práce bude zaoberať sa s problémami testovania porúch oneskorení v asynchrónnych

Tabuľka 1: Porovnanie poruchových simulátorov z hľadiska pokrytia porúch

Meno obvodu	Počet porúch	Pokrytia porúch			
		[24]	[10]	sériový	deduktívny
alloc_outbound	58	92%	100.0%	100.00%	100.00%
chu133	60	97%	96.9%	98.33%	98.33%
chu150	40	82%	97.1%	95.00%	95.00%
converta	56	46%	91.9%	96.43%	96.43%
dff	34	79%	85.7%	100.00%	100.00%
ebergen	46	N/A	95.7%	100.00%	100.00%
half	34	N/A	100.0%	94.12%	94.12%
hazard	40	86%	97.0%	100.00%	100.00%
master_read	132	46%	97.7%	95.45%	95.45%
mp_forward_pkt	66	95%	100.0%	100.00%	100.00%
mr1	152	40%	93.5%	—	—
nak_pa	76	91%	100.0%	100.00%	100.00%
nowick	50	98%	100.0%	100.00%	100.00%
ram_read_sbuf	84	89%	100.0%	100.00%	100.00%
rcv_setup	36	93%	100.0%	100.00%	100.00%
rpdf	26	92%	100.0%	100.00%	100.00%
sbuf_ram_write	82	78%	100.0%	100.00%	100.00%
sbuf_send_ctl	66	49%	94.9%	98.48%	98.48%
seq4	96	47%	95.2%	—	—

Tabuľka 2: Porovnanie poruchových simulátorov z hľadiska výpočtového času a pamäťových nárokov

Meno obvodu	Čas [s]		Zníženie času	Pamäť [kB]		Zvýšenie pamäte
	sériový	deduktívny		sériový	deduktívny	
alloc_outbound	0.19	0.07	63%	1668	1700	2%
chu133	0.23	0.29	-26%	1680	1780	6%
chu150	0.09	0.07	22%	1660	1552	-7%
converta	0.18	0.26	-44%	1560	1744	12%
dff	0.05	0.00	100%	1548	1672	8%
ebergen	0.07	0.01	86%	1552	1700	10%
half	0.08	0.13	-63%	1548	1680	9%
hazard	0.05	0.01	80%	1552	1692	9%
master_read	0.70	0.39	44%	1732	1964	13%
mp_forward_pkt	0.24	0.04	83%	1584	1608	2%
nak_pa	0.32	0.08	75%	1684	1612	-4%
nowick	0.14	0.05	64%	1664	1704	2%
ram_read_sbuf	0.25	0.06	76%	1688	1756	4%
rcv_setup	0.07	0.02	71%	1660	1664	0%
rpdf	0.04	0.00	100%	1536	1640	14%
sbuf_ram_write	0.15	0.01	93%	1688	1748	2%
sbuf_send_ctl	0.19	0.10	47%	1684	1728	3%

obvodoch. Snahou bude prispieť k metódam generovania testovacích vektorov a návrhu pre testovateľnosť porúch oneskorení v asynchrónnych obvodoch.

Základnou myšlienkou spôsobu testovania oneskorení pri rýchlostne nezávislých obvodoch bude skúmanie vlastností grafovej špecifikácie STG, ktorá špecifikuje správanie sa obvodu.

Literatúra

- [1] Al-Assadi, W. K. a Kakarla, S.: Testing of asynchronous NULL conventional logic (NCL) circuits in synchronous-based designs. Proceedings of the 22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems, str. 215–222, 2007.
- [2] Fant, K. M. a Brandt, S. A.: NULL convention logic: A complete and consistent logic for asynchronous digital circuit synthesis. Proceedings of the 1996 IEEE International Conference on Application Specific Systems, Architectures and Processors, str. 261–273, 1996.
- [3] Koppad, D., Bystrov, A., a Yakovlev, A.: Off-line testing of asynchronous circuits. Proceedings of the 18th International Conference on VLSI Design held jointly with 4th International Conference on Embedded Systems Design, str. 730–735, 2005.
- [4] Shi, F., Makris, Y., Nowick, S. M., a Singh, M.: Test generation for ultra-high-speed asynchronous pipelines. Proceedings of the IEEE International Conference on Test, str. 1018, 2005.
- [5] Sparsø, J. a Furber, S.: Principles of asynchronous circuit design: A systems perspective, kapitola 1–8. Kluwer Academic Publishers, 2001.
- [6] Frištacký, N., Kolesár, M., Kolenička, J., a Hlavatý, J.: Logické systémy. Alfa, 1986.
- [7] Shi, F. a Makris, Y.: Testing delay faults in asynchronous handshake circuits. Proceedings of the 2006 IEEE/ACM International Conference on Computer Aided Design, str. 193–197, 2006.
- [8] Cortadella, J., Kishinevsky, M., Kondratyev, A., Lavagno, L., a Yakovlev, A.: Petrify: a tool for manipulating concurrent specifications and synthesis of asynchronous controllers. IEICE Transactions on Information and Systems, E80-D(3):315–325, 1997.
- [9] Hauck, S.: Asynchronous design methodologies: an overview. Proceedings of the IEEE, 83(1):69–93, 1995.
- [10] Shi, F. a Makris, Y.: SPIN-SIM: Logic and fault simulation for speed-independent circuits. Proceedings of the 2004 International Test Conference, str. 597–606, 2004.
- [11] Shi, F. a Makris, Y.: SPIN-TEST: Automatic test pattern generation for speed-independent circuits. Proceedings of the 2004 International Conference on Computer Aided Design, str. 903–908, 2004.
- [12] Chakraborty, T. J., Agrawal, V. D., a Bushnell, M. L.: Delay fault models and test generation for random logic sequential circuits. Proceedings of the 29th ACM/IEEE Conference on Design Automation, str. 165–172, 1992.
- [13] Novák, O., Gramatová, E., Ubar, R., a kol.: Handbook of Testing Electronic Systems. Czech Technical University Publishing House, 2005.
- [14] Lee, H. K. a Ha, D. S.: On the generation of test patterns for combinational circuits. Technická Správa 12_93, Dep't of Electrical Eng., Virginia Polytechnic Institute, 1993.
- [15] Russel, S. J. a Norvig, P.: Artificial Intelligence - A Modern Approach. Prentice-Hall, Inc., New Jersey, 1995.
- [16] Jha, N. a Gupta, S.: Testing of Digital Systems. Cambridge University Press, 2003.
- [17] Menon, P. R. a Chappell, S. G.: Deductive fault simulation with functional blocks. IEEE Transactions on Computers, C-27(8):689–695, 1978.
- [18] Walczak, K.: Deductive fault simulation for sequential module circuits. IEEE Transactions on Computers, 37(2):237–239, 1988.
- [19] Sur-Kolay, S., Roncken, M., Stevens, K., Chaudhuri, P.P., a Roy, R.: Fsimac: a fault simulator for asynchronous sequential circuits. Proceedings of the 9th Asian Test Symposium, str. 114–119, 2000.
- [20] Dobai, R.: Design for delay testability of asynchronous digital circuits. Počítačové architektury a diagnostika 2008, str. 11–16, Liberec, Česko, 2008. ISBN 978-80-7372-378-1.
- [21] Dobai, R.: Testing of delay faults in asynchronous circuits. Informal Digest of Papers of the 14th IEEE European Test Symposium, Sevilla, Španielsko, 2009.
- [22] Dobai, R.: Test generation for asynchronous circuits. IIT.SRC 2009: Student Research Conference, str. 284–291, Bratislava, Slovensko, 2009. ISBN 978-80-227-3052-5.
- [23] Dobai, R. a Gramatová, E.: Deductive fault simulation for asynchronous sequential circuits. 12th EUROMICRO Conference on Digital System Design, Patras, Grécko, 2009. prijaté na publikovanie.
- [24] Shi, F. a Makris, Y.: Fault simulation and random test generation for speed-independent circuits. Proceedings of the 2004 Great Lakes Symposium on VLSI, str. 127–130, 2004.