

# APPLICATION OF EVOLUTIONARY ALGORITHM IN CREATION OF REGRESSION TESTS

**Michaela Belešová**

Master Degree Programme (2), FIT BUT

E-mail: xbeles00@stud.fit.vutbr.cz

Supervised by: Marcela Šimková

E-mail: isimkova@fit.vutbr.cz

**Abstract:** This paper proposes the application of an evolutionary algorithm for the creation of regression tests. The main goal is to reduce the number of test vectors obtained from functional verification, while the reduced vector set must maintain the functional coverage of the original set. For realization, the UVM verification methodology and the simulation tool ModelSim are used.

**Keywords:** functional verification, UVM, SystemVerilog, regression tests, genetic algorithms

## 1 ÚVOD

So zväčšovaním komplexnosti hardvérových obvodov rastie náročnosť ich verifikácie a testovania. Preto vznikajú nové metodiky a optimalizácie, ktoré tieto procesy zjednodušujú. Zaujímavou oblasťou na optimalizovanie je regresné testovanie [1]. Regresné testovanie znamená uchovanie sady testov pre vybraný obvod/systém a ich opätovné spúšťanie v stanovených intervaloch. Pravidelný beh regresných testov je nevyhnutný, keďže i jemné modifikovanie kódu je veľmi náchylné na zavádzanie chýb. Regresné testovanie je však drahé, pretože často pracuje s príliš veľkým počtom redundantných vektorov overujúcich tie isté vlastnosti.

Zaujímavým a zatiaľ takmer nepreskúmaným zdrojom pre vytvorenie sady regresných testov je funkčná verifikácia [2]. Dôvod záujmu je ten, že funkčná verifikácia umožňuje získanie sady vektorov, ktoré tzv. pokrývajú (angl. *coverage*) vlastnosti obvodu definované špecifikáciou, a teda všetky kľúčové vlastnosti obvodu. Snahou je dosiahnuť 100 % pokrytie. Redundancia vektorov je však v tomto prípade výrazná, čo vyplýva z generovania pseudonáhodných vektorov. Generované dáta môžu byť náhodné, splňajú formát daný špecifikáciou, ale vektory môžu overovať už raz overené vlastnosti. Práve preto je potrebné generované vektory optimalizovať.

Silným nástrojom na riešenie širokého spektra optimalizačných úloh sú evolučné algoritmy [3], ktoré zahŕňajú niekoľko heuristických stochastických metód inšpirovaných prírodou. Evolučné algoritmy sa javia ako dobrá možnosť pre optimalizáciu regresných testov.

Tento príspevok sa venuje návrhu evolučného algoritmu na redukcii počtu vektorov v sade regresných testov, získaných v procese funkčnej verifikácie obvodu, pri zachovaní pokrytia. Príspevok tiež rieši integráciu navrhnutého evolučného algoritmu s už pripraveným verifikačným prostredím.

## 2 NAMERANÁ REDUNDANCIA VEKTOROV V KONKRÉTNOM OBVODE

Problém redundancie vektorov je vhodné demonštrovať na konkrétnom obvode. Pre tieto účely bola zvolená aritmeticko-logická jednotka (ALU) so 404 nadefinovanými vlastnosťami. Aby sa dosiahlo maximálne pokrytie týchto vlastností je potrebné overiť každú z nich aspoň raz. V experimente s pseudonáhodným generovaním postupnosti vektorov v procese funkčnej verifikácie, bolo na dosiahnutie 100 % pokrytia potrebné vytvoriť 4000 vektorov. Pri použití týchto vektorov bolo prevedených až 15818 preverení definovaných vlastností, teda  $15818 - 404 = 15414$  overení bolo zbytočných. To poukazuje na približne 39-násobnú redundanciu a poskytuje tak veľa priestoru na optimalizácie.

### 3 NÁVRH OPTIMALIZÁCIE

Štruktúra prostredia, v ktorom má optimalizačný proces prebiehať, je znázornená na obrázku 1. Znázornené optimalizačné prostredie sa dá rozdeliť na dve časti.

1. Časť zakreslená čiernou farbou predstavuje vopred pripravené verifikačné prostredie. Toto prostredie je realizované pomocou metodiky Universal Verification Methodology (angl. *UVM*) [4]. Medzi základné UVM komponenty, ktoré sú súčasťou verifikačného prostredia, patria:

**uvm\_sequencer:** slúžiaci na generovanie vstupných vektorov,

**uvm\_driver:** privádzajúci vygenerované vektory na vstupy verifikovaného obvodu,

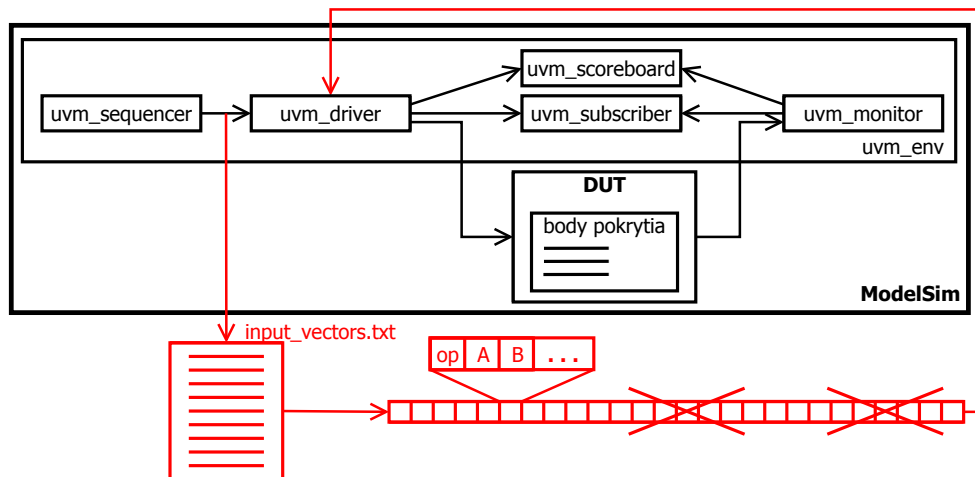
**uvm\_monitor:** čítajúci výstupy verifikovaného obvodu,

**uvm\_scoreboard:** overujúci správnosť výstupov verifikovaného obvodu voči výstupom z referenčného modelu,

**uvm\_subscriber:** slúžiaci pre meranie pokrytia vybraných vlastností obvodu a iných štatistík, napr. pokrytia kódu.

Komponent DUT (angl. *Device Under Test*) označuje verifikovaný obvod. Pre beh verifikácie a meranie pokrytia vlastností (body pokrytia na obrázku) sa využíva simulačný nástroj ModelSim spoločnosti Mentor Graphics.

2. Časť znázornená červenou farbou predstavuje cieľ mojej práce. Táto časť zahŕňa samotný evolučný algoritmus a jeho integráciu s verifikačným prostredím.



Obrázek 1: Integrácia evolučného algoritmu a verifikačného prostredia

Aplikácia evolučného algoritmu pri tvorbe optimálnych regresných testov prebehne nasledovne. Z verifikačného prostredia (na obrázku 1 znázorneného čiernou farbou) sa získa sada vstupných vektorov, pomocou ktorých sa podarilo v procese funkčnej verifikácie dosiahnuť maximálne pokrytie vybraných vlastností obvodu. Ako už však bolo vysvetlené, je vysoko pravdepodobné že získaná sada bude obsahovať redundantné vektory. Preto cieľom tejto práce je **redukovať** túto sadu evolučným algoritmom pri zachovaní rovnakého pokrytia sledovaných vlastností. Po aplikácii evolučného algoritmu bude nutné zmerať pokrytie vlastností redukovanou sadou vektorov. Toto sa bude realizovať privedením novozískanej sady vektorov do verifikačného prostredia na vstup komponenty `uvm_driver` namiesto vstupu od komponenty `uvm_sequencer` a opätovným spustením verifikácie.

### 4 EVOLUČNÝ ALGORITMUS A KANDIDÁTNE RIEŠENIA

Táto časť príspevku je zameraná na popis navrhnutého evolučného algoritmu. Ako už bolo spomenuté, evolučné algoritmy[3] zahŕňajú niekoľko optimalizačných metód. Pre svoje vlastnosti a oblasť využitia je pre problém optimalizácie sady regresných testov najvhodnejší genetický algoritmus (GA).

Pri návrhu GA pre optimalizáciu sady regresných testov však treba čeliť zásadnému problému. Ten spočíva v rôznej dĺžke chromozómov vyplývajúcej zo snahy o ich skracovanie, čo pre genetické algoritmy nie je typické. Na to je potrebné prihliadať pri navrhovaní procesu mutácie a kríženia, kde je nutné existujúce prístupy prispôbiť. Navrhovaný GA potom vyzerá nasledovne:

**Kandidátne riešenie:** skladá sa zo sady vstupných vektorov pre DUT.

**Počiatkové kandidátne riešenie:** nie je vygenerované náhodne, ale je získavané z funkčnej verifikácie ako sada vstupných vektorov dosahujúca najvyššie pokrytie vybraných vlastností.

**Mutácia:** je navrhovaná dvomi spôsobmi:

1. vypustenie niektorého z testovacích vektorov kandidátneho riešenia,
2. vzájomná výmena poradia dvoch vektorov v rámci chromozómu kandidátneho riešenia.

**Kríženie:** je možné aplikovať známe prístupy ako napríklad jednobodové či uniformné kríženie.

Avšak možnosť zmeny dĺžky chromozómov umožňuje realizovať aj všeobecnejší prístup a to taký, pri ktorom sú medzi dvomi jedincami zamenené bloky testovacích vektorov rôznej dĺžky a rôzneho umiestnenia v rámci chromozómu. Príklad uvedeného všeobecného kríženia chromozómov  $X$  a  $Y$ , kde vzniknú nové dva chromozómy  $X'$  a  $Y'$  je nasledujúci:

$$X : v_1^1 v_1^2 | v_1^3 v_1^4 v_1^5 | v_1^6 v_1^7, Y : v_2^1 | v_2^2 | v_2^3, X' : v_1^1 v_2^2 v_2^1 v_1^7, Y' : v_2^1 v_1^3 v_1^4 v_1^5 v_2^3$$

Pričom  $v_i^j$  označuje  $j$ -ty vektor  $i$ -teho chromozómu.

**Hodnotiaca funkcia:** vhodne kombinuje pokrytie a počet testovacích vektorov obsiahnutých v kandidátnom riešení

**Podmienka ukončenia výpočtu:** dlhodobá stagnácia ohodnotenia kandidátnych riešení, pretože nie je možné inak určiť mieru optimálnosti výsledného riešenia

GA majú veľký počet nastaviteľných parametrov. Medzi ne patria napr. pravdepodobnosť mutácie a kríženia, ako aj použitý algoritmus selekcie. Keďže neexistuje spôsob, ako je možné jednoznačne a efektívne určiť najlepšie nastavenie parametrov GA, je nutné ich vhodné nastavenie zistiť experimentálne. V budúcnosti by mohol byť ako prídavok implementovaný tzv. ostrovný GA [5] v jeho sekvenčnej verzii. Ostrovný algoritmus obvykle v porovnaní so základným GA nachádza lepšie výsledky a stačí mu menej krokov výpočtu.

## 5 ZÁVER

Tento príspevok popisuje nový zdroj získania regresných testov ako vektorov vygenerovaných v procese funkčnej verifikácie hardvérového obvodu a zároveň navrhuje genetický algoritmus ako vhodnú možnosť optimalizácie týchto testov. Popísaný genetický algoritmus pracuje s variabilnou dĺžkou chromozómov. Z toho dôvodu sú navrhnuté špeciálne metódy kríženia a mutácie. Vhodné nastavenie parametrov genetického algoritmu je však nutné zistiť experimentálne. Možné budúce vylepšenie návrhu je použitím ostrovného algoritmu.

## POĎAKOVANIE

Príspevok vznikol za podpory projektu VUT v Brne FIT-S-14-2297.

## REFERENCE

- [1] T. L. Graves, M. J. Harrold, J.-M. Kim, aj.. An Empirical Study of Regression Test Selection Techniques. In: ACM Transactions on Software Engineering and Methodology, 2001.
- [2] Meyer, A.. Principles of Functional Verification. USA: Elsevier Science, 2014. ISBN 0-7506-7617-5
- [3] Aliev, R. A., Aliev, A. R.. Soft Computing and Its Applications. USA: World Scientific Publishing Co., Inc., 2001. ISBN 9810247001
- [4] Accellera Systems Initiative. Universal Verification Methodology. 2014. URL: <http://www.accellera.org/community/uvm/>
- [5] Whitley, D., Rana, S., Heckendorn, R.. The Island Model Genetic Algorithm: On Separability, Population Size and Convergence. USA: Colorado State University, 1998.