

Řešič problému splnitelnosti booleovské formule (SAT) s dynamickou aktivací klauzulí (DP)

Vytvořte nástroj pro řešení problému splnitelnosti booleovské formule (SAT) využívající myšlenku dynamické aktivace klauzulí. Nejlepší cestou bude pravděpodobně modifikace stávajícího SAT řešiče, např. MiniSATu.

Řešič otestujte na tzv. obvodových SAT instancích (dodá vedoucí práce).

Literaturu dodá vedoucí práce, bližší znalost problematiky není nutná.

Programovací jazyk: nejlépe C (C++).

SAT Solver Based on Dynamic Clause Activation

Design a SAT-solver exploiting the idea of dynamic clause activation. Most probably the best way to go will be to modify some already available SAT-solver, e.g., MiniSAT.

Test the SAT-solver using provided circuit-SAT instances.

Programming language: C++.

Kompresa testu pro číslicové obvody s RAS architekturou (BP, DP)

Vytvořte nástroj pro generování komprimovaného testu (ATPG) pro číslicové obvody s „Random Access Scan“ (RAS) architekturou. Algoritmus bude založený na implicitní reprezentaci testovacích vektorů, jako instance problému splnitelnosti booleovské formule (SAT). Proveďte experimentální vyhodnocení, porovnejte se stávajícími přístupy.

Programovací jazyk: nejlépe C, vzhledem k dostupnosti SAT řešičů.

Hlubší znalosti problematiky testování číslicových obvodů nejsou nutné.

Test Compression for the RAS Architecture

Develop an automated test equipment algorithm (ATPG) for the “Random Access Scan” (RAS) architecture. The algorithm will be based on implicit representation of test vectors - by a SAT problem instance.

Conduct an experimental evaluation of the algorithm and compare the results with competitive approaches.

Programming language: C++ (preferably).

Detailed knowledge of the circuit testing problematics is not essentially necessary.

Kompresa testu pro číslicové obvody s Illinois-Scan architekturou (BP, DP)

Vytvořte nástroj pro generování komprimovaného testu (ATPG) pro číslicové obvody s "Illinois-Scan" architekturou. Algoritmus bude založený na implicitní reprezentaci testovacích vektorů, jako instance problému splnitelnosti booleovské formule (SAT). Proveďte experimentální vyhodnocení, porovnejte se stávajícími přístupy.

Programovací jazyk: nejlépe C, vzhledem k dostupnosti SAT řešičů.

Hlubší znalosti problematiky testování číslicových obvodů nejsou nutné.

Test Compression for the Illinois-Scan Architecture

Develop an automated test equipment algorithm (ATPG) for the “Illinois-Scan” architecture. The algorithm will be based on implicit representation of test vectors - by a SAT problem instance.

Conduct an experimental evaluation of the algorithm and compare the results with competitive approaches.

Programming language: C++ (preferably).

Detailed knowledge of the circuit testing problematics is not essentially necessary.

Kompresie testu pro číslicové obvody (BP, DP)

Naprogramujte zvolený algoritmus pro kompresi testu pro číslicové obvody a navrhnete HW architekturu pro dekompresi. Na výběr je mnoho různých algoritmů (RAS, Illinois scan, reseeding, EDT, ...). Vstupem algoritmu bude deterministický test (testovací vektory), výstupem komprimovaný test a dekompresní HW architektura (VHDL kód, případně implementace v FPGA).

Test Compression for Digital Circuits

Implement a selected test compression algorithm for digital circuits and design a HW architecture of the decompressor. There are many algorithms to choose from (RAS, Illinois scan, reseeding, EDT, ...). The input to the algorithm will be a deterministic test (test vectors), the output will be the compressed test and the decompression architecture (VHDL code, possibly implemented in FPGA).

Vliv malých změn v HDL kódu na syntézu (DP)

Prozkoumejte vliv malých změn v HDL (VHDL, případně Verilog) kódu na následnou syntézu komerčními nástroji. Malými změnami je rozuměna např. změna pořadí signálů v portech, reformulace jazykových konstruktů (case-if), přičemž je zachována funkční ekvivalence popisovaného obvodu. Fakt, že tyto malé změny mají vliv na výslednou velikost obvodu, zpoždění, atp., byl pozorován. Cílem práce je odhalit, do jaké míry je toto závažné, případně navrhnout doporučení. Jedná se o práci jednak implementační (naprogramování "injektoru změn" v HDL kódu), jednak experimentální.

Výstupem diplomové práce bude:

- analýza možností provedení malých změn (jaké konstrukty měnit),
- nástroj pro injekci změn do HDL kódu, konkrétně VHDL, případně Verilogu (stanoví vedoucí práce),
- důkladné experimentální vyhodnocení vlivu těchto změn na výsledek syntézy pomocí komerčních nástrojů (Xilinx XST, Altera Quartus) s použitím standardních zkušebních úloh a průmyslových návrhů (dodá vedoucí práce).

Influence of Small Changes in HDL Code on the Synthesis quality

Explore the influence of small changes in a HDL code (VHDL, Verilog) to the subsequent synthesis by commercial tools (Xilinx ISE, Vivado, Quartus). By "small changes" we understand, e.g., altering the ordering of signals definitions, using different language constructs, while the function is retained. It is a partially implementation (implementing the changes injector) and partially experimental work.

Určování míry podobnosti logických obvodů (BP, DP)

Vytvořte nástroj pro určování míry podobnosti logických obvodů. Vstupem budou dva obvody popsané logickou sítí (netlist), které jsou funkčně ekvivalentní, ale jejich struktura je odlišná. Cílem práce je navrhnout metriku "podobnosti" takovýchto log. sítí a naimplementovat algoritmus pro výpočet míry této podobnosti.

Hlubší znalost problematiky návrhu logických obvodů není nutná - jedná se spíše o řešení grafového problému.

Measuring the Similarity of Logic Circuits

Create a tool for computing of similarity of two logic circuits. The input will be two functionally equivalent (but structurally different) circuits described by a netlist. Design a metric to compute similarity of these two circuits and implement an algorithm computing it.

Resyntéza obvodu po částech v ABC (BP, DP)

Seznamte se s open-source systémem pro syntézu a optimalizaci logických obvodů ABC. V něm naimplementujte nový příkaz pro novou možnost optimalizace obvodů - resyntézu po částech.

Algoritmus bude operovat primárně nad tzv. AIG strukturou, možné je rozšíření i na jiné podporované struktury. Jazyk: C.
Není nutná zevrubná znalost problematiky syntézy logických obvodů, spíše schopnost porozumění cizímu kódu.

Circuit Resynthesis by Parts in ABC

Get acquainted with an open-source logic synthesis and optimization package ABC. Implement a new circuit optimization algorithm (implement a new command in ABC) based on resynthesis by parts. The algorithm will operate on top of the AIG data structure, which is native to ABC. However, it could be possible to use different structures too. Programming language: C.

Detailed knowledge of the logic synthesis problematics is not essentially necessary. Which is necessary, it the ability to understand the code.

Randomizace algoritmů v systému ABC (BP, DP)

ABC je interaktivní open-source nástroj pro logickou syntézu vytvářený na University of California, Berkeley. Je v něm implementována řada syntézních algoritmů (jako je minimalizace, mapování na technologii, FPGA mapování, ...). Všechny tyto algoritmy jsou plně deterministické, tj. při opakovaném spouštění produkují stejné výsledky. Na druhou stranu jsou však citlivé např. na pořadí vstupů syntetizovaného obvodu, což svědčí o přítomnosti algoritmů, do kterých by mohlo být možné vložit prvek náhodného výběru, bez narušení funkčnosti. Cílem diplomové práce je vybrané algoritmy randomizovat, tj. do míst, kde se algoritmus rozhoduje mezi "lokálně ekvivalentními" volbami, zavést náhodný výběr. Výsledné randomizované algoritmy otestujte na standardních zkušebních úlohách a porovnejte s původními deterministickými.

Randomization of Algorithms in ABC

ABC is an interactive open-source tool from UC California, Berkeley. Numerous logic synthesis and optimization algorithms implemented in it (minimization, technology mapping, etc.). All these algorithms are almost fully deterministic, i.e., they generate equal results when run repeatedly. The aim of the thesis work is to introduce randomness into some selected algorithms and perform the experimental evaluation.

Programming language: C. Detailed knowledge of the logic synthesis problematics is not essentially necessary. Which is necessary, it the ability to understand the code.

Dekompozice logických funkcí založená na binárních rozhodovacích diagramech (DP)

Vytvořte nástroj (resp. "framework") pro dekompozici logických funkcí založený na dekompozici binárních rozhodovacích diagramů (BDD). Je možno použít části dostupných zdrojových kódů podobných nástrojů. Navrhněte heuristiky pro efektivní řízení dekompozice, analyzujte různé možnosti. Vytvořený nástroj otestujte na standardních zkušebních úlohách, získané výsledky porovnejte s alternativními řešeními.

BDD-based Decomposition of Logic Functions

Design a tool for decomposition of logic functions based on binary decision diagrams (BDDs). You may re-use available source code from similar existing tools (BDS). Design heuristics for effective decomposition control, analyze different possibilities. The designed tool should be tested using standard benchmark circuits.

Generování testu pro kombinační obvody s ohledem na implementaci standardními buňkami (DP)

Cílem práce je navrhnout a naimplementovat algoritmus pro automatické generování testu (ATPG) pro kombinační obvody, s ohledem na finální implementaci pomocí standardních buněk. Poruchový model je zde odlišný od klasického (trvalá 1/trvalá 0); uvažují se poruchy přímo na úrovni tranzistorů. Klasické strukturní ATPG algoritmy pro tento poruchový model jsou známy (resp. rozšíření stávajících algoritmů je přímočaré). Tato práce by se měla zaměřit na využití principů ATPG založených na splnitelnosti booleovské formule (SAT-based ATPG) pro tento účel.

- 1) Nastudujte principy ATPG založených na splnitelnosti booleovské formule (SAT-based ATPG).
- 2) Nastudujte principy generování testu pro standardní buňky (Cell-Aware testing).
- 3) Společně s vedoucím práce navrhnete způsob, jak využít ATPG založené na splnitelnosti booleovské formule pro generování testu pro standardní buňky, případně proveďte rešerši, jestli takové přístupy již neexistují.
- 4) Výše zmíněné naimplementujte a otestujte.

Databáze konferencí a publikací IV (BP, DP)

Database of Conferences and Publications IV

Cílem práce je pokračovat v návrhu stávající aplikace pro evidenci konferencí a publikací (<https://ddd.fit.cvut.cz/PubConf/>). Použitá platforma: Nette framework, MySQL.

Aplikaci podporuje evidenci konferencí, jejich ročníků a publikací, tyto informace jsou propojené.

Mezi hlavní požadavky na rozšíření patří:

- odstranění drobných chyb, vylepšení uživatelské přívětivosti,
- vylepšení vyhledávání publikací podle klíčových slov,
- zavedení vícekritériálního vyhledávání publikací,
- vylepšení vyhledávání referencí,
- lepší správa referencí (správa „vlastních“ publikací, agregace, ...),
- sjednocení formátu pro zobrazování publikací,
- opravení BibTeX importu a exportu, hromadný BibTeX export,
- zavedení nápovědy,
- implementace nových možností aplikace (lepší našeptávače, hodnocení, agregace, lepší personalizace, hromadná správa uživatelů, zavedení uživatelských podskupin publikací, provázání s V3S, atd.).

Přesné požadavky stanoví vedoucí práce.

Veškeré provedené změny důkladně otestujte, proveďte uživatelské testy.