

Návrh metód generovania BIST pre vnorené pamäte v systémoch na čipe

Juraj Šubín

2. ročník, denná forma

Školiteľ: Elena Gramatová

Slovenská technická univerzita, Fakulta informatiky a informačných technológií

Ilkovičova 2, 84216 Bratislava 4, SR

juraj.subin@stuba.sk

Abstrakt—Príspevok opisuje návrh nového systému automatického generovania architektúry BIST pre testovanie vnorených pamätí integrovaných v systémoch na čipe. Existuje niekoľko podobných systémov, no niektoré z nich sú špecificky zamerané na konkrétne technológie, využívané v návrhárskych firmách, iné neberú do úvahy niektoré dôležité parametre. Chýba teda systém, dostatočne flexibilný k požiadavkám používateľa (návrhára), ktorý by bol dostupný pre použitie najmä na akademickej úrovni. Úlohou navrhovaného systému je rozdeliť pamäte, integrované v systéme na čipe, do testovacích okolí, ktoré testujú pamäte sériovo alebo paralelne, v závislosti od rôznych typov zadaných požiadaviek, ako napr. čas testovania, navýšenie plochy na čipe, spotreba energie. Výstupom systému má byť syntetizovateľný VHDL kód každého bloku vygenerovanej architektúry BIST. Vstupy od používateľa a jeho interakcia so systémom sú potrebné na to, aby bol systém schopný vygenerovať vyhovujúcu architektúru BIST spĺňajúcu požiadavky používateľa.

Kľúčové slová—systém na čipe; vnorená pamäť; testovanie; BIST; VHDL.

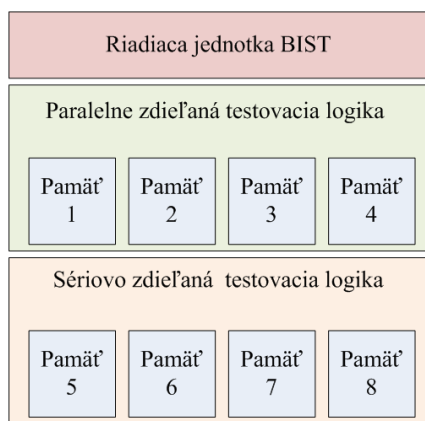
I. ÚVOD

Systémy na čipe (SoC – *system on chip*) sú integrované do každodenne používaných elektronických zariadení, ktoré je potrebné testovať po ich výrobe resp. počas ich životnosti. Súčasná technológia poskytuje možnosti vývoja zložitejších systémov s väčšou funkcionalitou, čo prispieva aj k zložitosti ich testovania. Veľká pozornosť je sústredená na testovanie pamätí vnorených v SoCs, pretože v súčasnosti zaberajú najväčší podiel z ich plochy (cca 86 %) [1]. Na jednom SoC môže byť integrovaných stovky až tisícky pamätí rôzneho typu a veľkosti, pričom všetky je potrebné otestovať. Vstavané samočinné testovanie (BIST – *built-in self-test*) je stále vhodným prístupom k testovaniu veľkého počtu vnorených pamätí. Architektúra BIST je navrhovaná ako jednoúčelová testovacia logika, ktorá je integrovaná v SoC spolu so všetkými ostatnými komponentami, a preto je každá architektúra BIST cielene navrhovaná pre potreby daného SoC. Schopnosť detegovania porúch v pamäti závisí od voľby testovacieho algoritmu. V súčasnosti najrozšírenejšie sú algoritmy typu march, ktorých hlavnými výhodami sú lineárna zložitnosť, dostatočné pokrytie porúch a používanie

pravidelných testovacích vzorov, preto sú aj vhodné pre použitie v BIST. Výber konkrétneho algoritmu march závisí od žiadaného pokrytia porúch a dĺžky trvania testu. Iným dôležitým aspektom, ktorý je potrebné vziať do úvahy pri testovaní pamätí, je výber adresnej schémy pre testovací algoritmus. Je známych niekoľko špeciálnych kódov, z ktorých je možné vyberať v závislosti od cieľu testovania. Napríklad, lineárna adresná schéma je jednoduchá a využívaná prevažne na detekciu porúch v jednej bunke alebo porúch spárených buniek v pamäťovej matici. Komplementárna adresná schéma je užitočná pre stresovanie adresného dekódera a pre detekciu dynamických porúch zapríčinených rýchlymi zmenami viacerých signálov adresného dekódera v jednom čase. Grayov kód je dobre známa adresná schéma, ktoré má využitie najmä pri testovaní s nízkou spotrebou energie, nakoľko po sebe idúce adresy majú Hammingovu vzdialenosť vždy rovnú 1 [2].

V súčasnosti existuje niekoľko prístupov k testovaniu veľkého počtu pamätí vnorených v SoC. Najzákladnejším prístupom je použitie obvodu BIST pre každú pamäť v SoC. Tento prístup je náročný na plochu na čipe a môžu nastať problémy s riadením testovania a zbieraním informácií o detegovaných poruchách. Mierne vylepšený prístup je použitie nezdieľaných testovacích okolí (každá pamäť má pridané vlastné testovacie okolie), riadených centralizovanou riadiacou jednotkou. Testovacie okolie pozostáva z logiky potrebnej na otestovanie pamäte (adresný generátor, údajový generátor, riadiaci blok, komparátor a komunikačné rozhranie). Nakoľko testovacie okolia sú v tomto prípade nezdieľané, problém s vysokým navýšením plochy na čipe pretrváva. Najnovší prístup využíva výhody vyplývajúce zo zdieľania testovacích okolí, ktoré sú riadené centralizovanou riadiacou jednotkou (obr. 1). Pamäte, ktoré zdieľajú testovacie okolie, môžu byť testované buď sériovo, alebo paralelne. Sériové testovanie je efektívne pre dosiahnutie nízkej spotreby energie počas testovania. Výhodou paralelného testovania je dosiahnutie krátkeho času testovania. Využitím hybridnej BIST architektúry (kombinácia sériového a paralelného testovania) je možné nájsť kompromis medzi časom testovania, navýšením plochy na čipe a spotrebou energie. V súčasnosti sa hybridné testovanie javí ako najlepší prístup k

testovaniu veľkého počtu pamätí integrovaných v SoC. Avšak, návrh takejto hybridnej BIST architektúry pre konkrétny SoC nie je jednoduchá úloha. Je potrebné prihliadať na viacero parametrov, ako napr. počet, typ a veľkosť pamätí, spotreba energie, navýšenie plochy na čipe, aplikácia testu, dĺžka testovania atď. Ručný návrh optimálnej architektúry BIST je časovo náročný proces a nájdenie optimálneho riešenia nemusí byť garantované. Z uvedeného dôvodu môže byť systém pre automatické generovanie architektúry BIST veľmi užitočný a efektívny. Existuje niekoľko podobných riešení, no niektoré z nich sú špecificky zamerané na konkrétne technológie, využívané v návrhárskych firmách, iné na druhej strane neberú do úvahy niektoré dôležité parametre. To je hlavná motivácia na návrh novej metódy generovania architektúry BIST na testovanie vnorených pamätí v SoC, ktorá bude flexibilná k požiadavkám používateľa (návrhára).



Obr. 1. Všeobecná schéma hybridnej BIST architektúry.

Článok je rozdelený do piatich častí vrátane úvodu a záveru. Druhá časť opisuje existujúce prístupy a systémy automatického generovania BIST, tretia časť je venovaná opisu novej metódy generovania architektúry BIST, štvrtá časť je zameraná na ciele a tézy dizertačnej práce a piata časť je záverom.

II. SYSTÉMY AUTOMATICKÉHO GENEROVANIA BIST

V súčasnosti sú známe štyri systémy automatického generovania architektúry BIST pre vnorené pamäte na čipe [3] – [6]. Niektoré z nich boli vyvinuté pre špecifické potreby návrhárskych firiem a sú nedostupné na výskum. V tabuľke 1 sa nachádza porovnanie týchto systémov podľa troch zvolených hľadísk. Porovnávané systémy sú pre jednoduchosť označené ako BIST_1 až BIST_4 s ich citáciou, nakoľko v publikáciách nemajú tieto systémy priradený špecifický názov. Hlavnou nevýhodou systému BIST_1 je, že všetky pamäte sa začínajú testovať paralelne v rovnakom čase. Tento fakt bráni systému dosiahnuť optimálne riešenie z pohľadu všetkých požiadaviek uvedených v tabuľke. Systém BIST_2 nepodporuje zdieľané sériové testovacie okolie, preto nemožno optimalizáciu architektúry zacieliť na spotrebu energie. Vstupné údaje systému BIST_3 nie sú v publikácii uvedené, preto nemožno povedať, či je parameter spotreby energie braný do úvahy pri

snahe vytvoriť čo najmenší počet skupín pamätí. Architektúru v systéme BIST_4 možno optimalizovať len z pohľadu navýšenia plochy na čipe. Žiadny z uvedených systémov nepodporuje všetky základné typy pamätí – SRAM, DRAM, ROM. Ďalej chýba možnosť voľby testovacieho algoritmu a použitej adresnej schémy. Formát výstupu takisto neuvádza žiaden zo systémov. Uvedené nedostatky sú motiváciou pre návrh nového systému pre automatické generovanie architektúry BIST pre testovanie vnorených pamätí v SoC s novo navrhnutými metódami a algoritmi, kde nebudú použité štandardy IEEE1149.1 [7] alebo IEEE1687 [8].

TABUĽKA I. POROVNANIE EXISTUJÚCICH SYSTÉMOV

Prístup / Porovnanie	Požiadavky	Druhy testovacích okolí	Poradie testovania pamätí
BIST_1 [3]	plocha na čipe spotreba energie čas testu	nezdieľané, zdieľané paralelné & sériové	test pre všetky pamäte štartuje v rovnakom čase
BIST_2 [4]	plocha na čipe čas testu dĺžky prepojení	zdieľané paralelné	1 skupina pamätí testovaná v jednom čase
BIST_3 [5]	area overhead čas testu dĺžky prepojení najmenší počet skupín	zdieľané paralelné & sériové	získavané výpočtom
BIST_4 [6]	plocha na čipe	zdieľané paralelné & sériové	získavané výpočtom
navrhovaný systém	plocha na čipe spotreba energie čas testu	nezdieľané, zdieľané paralelné & sériové	získavané výpočtom

III. METÓDY GENEROVANIA ARCHITEKTÚRY BIST

Navrhovaný systém generovania architektúry BIST je založený na hybridnej metóde testovania a mal by byť flexibilný a škálovateľný vzhľadom na požiadavky používateľa. Vnorené pamäte budú testované cez testovacie okolia pozostávajúce z adresného a údajového generátora, riadiaceho bloku, komparátora a komunikačného rozhrania. Testovacie okolia budú riadené centralizovanou riadiacou jednotkou BIST. Tento prístup je použitý aj v iných BIST architektúrach, ale zložitost' a komplexnosť testovacích okolí a riadiacej jednotky tu závisí od požiadaviek používateľa a návrhových obmedzení spojených s konkrétnym SoC. Architektúra BIST môže byť prepojená aj na blok vstavanej samočinnej opravy (BISR – *built-in self-repair*). Preto je viac efektívne generovať architektúru BIST podľa požiadaviek používateľa ako použiť štandardnú architektúru a integrovať ju na SoC, pričom by sa nebrali do úvahy niektoré špecifické parametre. Pre vývoj takéhoto systému je potrebné navrhnuť niekoľko metód pre vygenerovanie optimálnej architektúry BIST vzhľadom na rôzne požiadavky, a to metóda voľby adresných schém, metóda rozdelenia pamätí do testovacích okolí, metóda zostavenia riadiaceho bloku na riadenie BIST všetkých pamätí vnorených v SoC, metóda ohodnotenia žiadaných parametrov, podľa ktorých sa bude riadiť

rozdelenie pamätí a pod. Vygenerovaná architektúra BIST bude jedinečná pre danú kombináciu žiadaných parametrov, typy pamätí integrovaných v SoC. Predpokladá sa využitie týchto parametrov, rozdelených do troch skupín:

I. Parametre pamätí

- počet a veľkosť pamätí,
- typ pamätí (DRAM, SRAM, ROM),
- rozhranie pamätí,
- frekvencia operácií zápisu a prečítania,
- zdieľané / nezdieľané testovacie okolie.

II. Parametre testovania

- typ testu (rôzne algoritmy march),
- adresné schémy (napr. lineárna, komplementárna, atď.),
- čas testu,
- obmedzenia v spotrebe energie.

III. Špecifické parametre

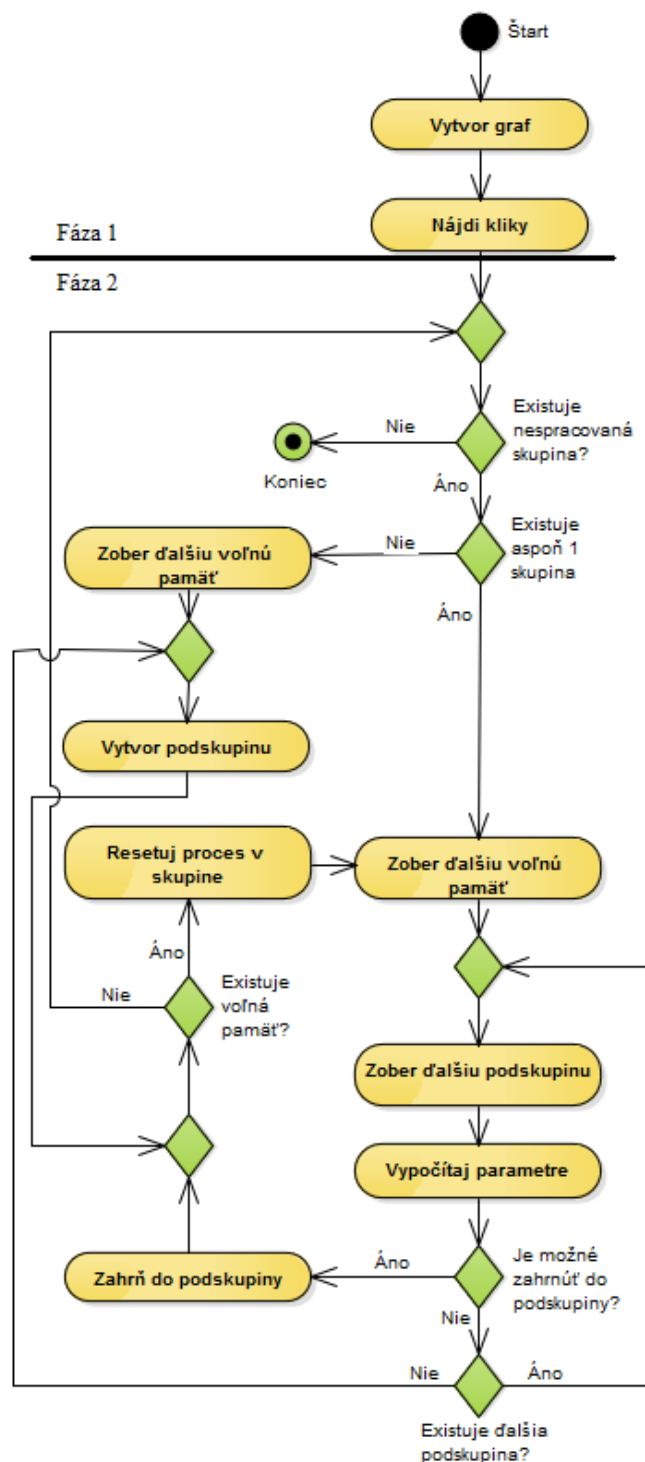
- požiadavka optimalizácie architektúry,
- obmedzenia v navýšení plochy na čipe,
- prepojenie na BISR,
- používateľsky definované skupiny pamätí.

Základom systému je rozdelenie pamätí do testovacích okolí, ktoré si vyžaduje jednak návrh algoritmu na ich rozdelenie v závislosti od typu pamätí a zvoleného cieľu optimalizácie architektúry, návrh metód ohodnotenia parametrov spotreby energie, pridanej plochy a času testovania, ktoré budú súčasťou tohto algoritmu. Jedna pamäť môže byť priradená len pod jedno testovacie okolie. Počet pamätí pod jedným testovacím okolím a typ testovacieho okolia budú vždy ovplyvnené hraničnými hodnotami troch parametrov: časom testu (ak je zadaný), spotrebou energie a navýšením plochy na čipe. Nový algoritmus pre nájdenie optimálneho rozdelenia pamätí do testovacích okolí využíva grafové metódy a je zobrazený na obr. 2. Algoritmus pozostáva z dvoch fáz. Prvá fáza je tzv. inicializačná fáza, ktorej cieľom je vytvoriť neorientovaný graf reprezentujúci kompatibility pamätí z hľadiska ich testovania. Každá pamäť je reprezentovaná vrcholom grafu. Ak sú 2 vrcholy grafu prepojené hranou, znamená to, že dané pamäte sú kompatibilné z hľadiska testovania a môžu byť priradené pod jedno testovacie okolie, či už sériové, alebo paralelné. V navrhovanom algoritme sú dve pamäte považované za kompatibilné z hľadiska testovania v prípade, ak:

- sú rovnakého typu,
- pracujú na rovnakej operačnej frekvencii,
- budú testované rovnakým testom march,
- bude použitá rovnaká adresná schéma.

Ďalšou časťou fázy 1 je nájdenie najväčších kompletných podgrfov (kliky) grafu a vytvorenie skupín pamätí kompatibilných pre testovanie.

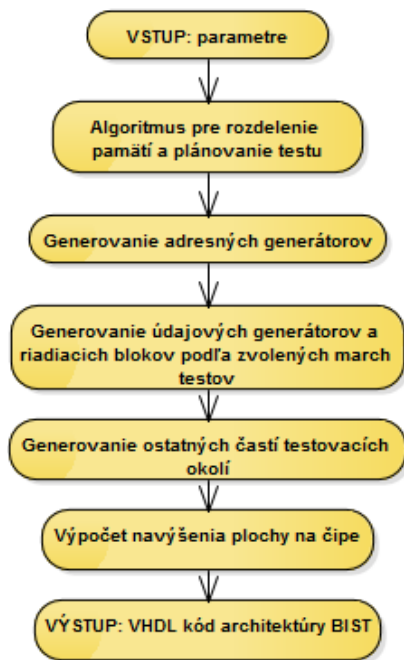
V druhej fáze algoritmu sa vykonávajú výpočty a prerozdelenie rozhodnutia. Cieľom tejto fázy je analyzovať vytvorené skupiny pamätí jednu po druhej a rozdeliť ich ďalej



Obr. 2. Algoritmus rozdelenia pamätí do testovacích okolí.

do podskupín, ak je to potrebné. Metóda na takéto rozdelenie musí akceptovať špecifické parametre, uvedené vyššie, a zadané obmedzenia pre vytváranie podskupín. Po ukončení tejto fázy sú už pamäte rozdelené do finálnych podskupín, pre ktoré budú vygenerované zdieľané testovacie okolia. Prioritou algoritmu je dosiahnuť optimálnosť generovanej architektúry z pohľadu zvoleného parametra (parametrov) optimalizácie.

Výsledkom systému automatického generovania BIST je optimálna architektúra BIST s týmito výstupmi: VHDL model blokov BIST, prepojenie na BISR, ak je žiadané, optimálny plán testu, odhad spotreby energie počas testovania, výpočet navýšenia plochy. Implementáciou metódy vznikne nový akademický systém pre automatické generovanie architektúry BIST – MBISTGen s grafickým používateľským rozhraním. Základná schéma systému MBISTGen je zobrazená na obr. 3.



Obr. 3. Všeobecná schéma systému MBISTGen.

IV. CIELE DIZERTAČNEJ PRÁCE

Cieľ dizertačnej práce je návrh metód na automatické generovanie architektúry BIST pre vnorené pamäte v SoC. Vstupom sú požiadavky používateľa (spotreba energie, dĺžka testu a pridaná plocha v SoC), parametre pamätí, existujúce metódy testovania porúch v pamätiach. Výsledná architektúra BIST bude opísaná v jazyku VHDL. Preto tézy dizertačnej práce sú:

- Špecifikácia požiadaviek a obmedzení na architektúru BIST na návrh optimálneho BIST na veľký počet vnorených pamätí.
- Návrh a vývoj blokov BIST, ich modelovanie vo VHDL, flexibilných z pohľadu počtu a rôznorodosti pamätí vnorených do čipu, ako aj testov march či aplikácie testov pri použití rôznych adresných schém.
- Návrh optimálnych architektúr sériovo-paralelných BIST, škálovateľných na počet pamätí, čas testovania a minimalizáciu spotreby energie pri testovaní.
- Návrh novej metódy na rozdelenie pamätí integrovaných v SoC na sériové a paralelné testovanie v závislosti od požiadaviek návrhára na plochu, čas testovania a spotrebu energie počas testovania.

- Overenie navrhutej metódy na automatické generovanie BIST vo VHDL pre vybrané SoC a zadané požiadavky na testovanie pamätí s ich rôznymi zostavami a typov.

V. ZÁVER

Stúpajúca miera integrácie čoraz väčšieho počtu jadier a funkčných blokov na čipe si vyžaduje nové prístupy k riešeniu ich testovania, a to so zameraním sa na testovanie pamätí, ktoré zaberajú najväčšiu časť plochy čipu. Návrh vstavanej architektúry vykonávajúcej samočinné testovanie vysokého počtu pamätí nie je triviálna úloha. Z analýzy existujúcich systémov automatického generovania BIST pre pamäte v SoC bol identifikovaný priestor pre vývoj nového systému automatického generovania testovacej architektúry ponúkajúceho navyše oproti existujúcim prístupom napríklad možnosť voľby testovacieho algoritmu, adresnej schémy, alebo poskytovacieho výstup vo formáte VHDL. Ďalšia práca bude sústredená na implementáciu navrhnutého algoritmu a vhodných blokov BIST.

POĎAKOVANIE

Táto práca bola podporovaná Agentúrou na podporu výskumu a vývoja na základe národného projektu VEGA 1/0616/14.

LITERATÚRA

- [1] International Technology Roadmap for Semiconductors (ITRS). 2011 Edition. Test and Test Equipment. <http://www.itrs.net>.
- [2] A.J. van de Goor, H. Kukner, S. Hamdioui, "Optimizing Memory BIST Address Generator Implementations," in IEEE International Conference on Design & Technology of Integrated Systems in Nanoscale Era, 2011, p. 6.
- [3] L. Zaourar, Y. Kieffer, A. Wenzel, "A Complete Methodology for Determining Memory BIST Optimization under Wrappers Sharing Constraints," in Asia Symposium on Quality Electronic Design (ASQED), 2011, pp. 46-53.
- [4] Ch. Tzuo-Fan, Ch. Wen-Chi, L. Chien-Mo, Ch. Yao-Wen, L. Kuan-Yu, Ch. Ming-Tung, T. Min-Hsiu, T. Chih-Mou, "BIST Design Optimization for Large-Scale Embedded Memory Cores," in IEEE/ACM International Conference on Computer-Aided Design - Digest of Technical Papers, 2009, pp. 197-200.
- [5] A.B. Kahng, I. Kang, "Co-Optimization of Memory BIST Grouping, Test Scheduling, and Logic Placement," in IEEE Design, Automation and Test in Europe Conference and Exhibition (DATE), 2014, p. 1-6.
- [6] M. Miyazaki, T. Yoneda, H. Fujiwara, "A Memory Grouping Method for Sharing Memory BIST Logic," in Asia and South Pacific Conference on Design Automation, 2006, pp. 671-676.
- [7] Standard Test Access Port and Boundary-Scan Architecture, IEEE Std 1149.1, 2008. [online] Available at: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=938734> [Accessed 11 Jan. 2016].
- [8] Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device, IEEE Std 1687, 2014. [online] Available at: http://www.techstreet.com/ieee/products/vendor_id/3931 [Accessed 11 Jan. 2016].