

ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՅՄԱՆ
ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏ

Հայրապետյան Դավիթ Լևոնի

**ԲՅՈՒՐԵՂԻ ՎՐԱ ՍԽԱԼՆԵՐԻ ԱՌԿԱՅՈՒԹՅԱՄԲ ՀԻՇՈՂ ՀԱՆԳՈՒՅՑՆԵՐԻ
ՆԵՐՎԱՌՈՒՑՎԱԾ ԹԵՍՏԱՎՈՐՄԱՆ ԳՈՐԾԸՆԹԱՅԸ ՄՈԴԵԼԱՎՈՐՈՂ ԾՐԱԳՐԱՅԻՆ
ԳՈՐԾԻՔԻ ՀԻՄՆԱՎՈՐՈՒՄ ԵՎ ՄՇԱԿՈՒՄ**

Ե. 13.04

«Հաշվողական մեքենաների, համալիրների, համակարգերի և ցանցերի
մաթեմատիկական և ծրագրային ապահովում» մասնագիտությամբ

Տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի
հայցման ատենախոսության

ՍԵՂՄԱԳԻՐ

Երևան – 2019

ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ И
АВТОМАТИЗАЦИИ НАН РА

Айрапетян Давид Левонович

**ОБОСНОВАНИЕ И РАЗРАБОТКА ПРОГРАММНЫХ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ
МОДЕЛИРУЮЩИХ ПРОЦЕССЫ ВСТРОЕННОГО ТЕСТИРОВАНИЯ БЛОКОВ ПАМЯТИ НА
КРИСТАЛЛЕ В УСЛОВИЯХ ПРИВНЕСЕНИЯ ОШИБОК**

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

По специальности: 05.13.04 – математическое и программное обеспечение
вычислительных машин, комплексов, систем и сетей

Ереван – 2019

Ատենախոսության թեման հաստատվել է Երևանի պետական համալսարանում

Գիտական ղեկավար՝ Ֆիզ.-մաթ. գիտ. դոկտոր Ս. Կ. Շուքուրյան

Պաշտոնական ընդդիմախոսներ՝ տեխ. գիտ. դոկտոր Հ. Հ. Հարությունյան
Ֆիզ.-մաթ. գիտ. թեկնածու Վ. Ա. Վարդանյան

Առաջատար կազմակերպություն՝
Երևանի մաթեմատիկական մեքենաների գիտահետազոտական ինստիտուտ

Պաշտպանությունը կայանալու է 2019թ. հունիսի 25-ին, ժ. 17:00-ին ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում գործող 037 «Ինֆորմատիկա» մասնտագիտական խորհրդի նիստում հետևյալ հասցեով՝ Երևան, 0014, Պ. Սևակի 1:

Ատենախոսությանը կարելի է ծանոթանալ ՀՀ ԳԱԱ ԻԱՊԻ գրադարանում:
Սեղմագիրն առաքված է 2019թ. մայիսի 17-ին:

Մասնագիտական խորհրդի գիտական
քարտուղար, ֆիզ.-մաթ. գիտ.դոկտոր  Հ. Գ. Սարուխանյան

Тема диссертации утверждена в Ереванском государственном университете

Научный руководитель: доктор физ.-мат. наук С.К. Шукурян

Официальные оппоненты: доктор технических наук Г.А. Арутюнян
кандидат физ.-мат. наук В.А. Варданян

Ведущая организация:
Ереванский научно-исследовательский институт математических машин

Защита состоится 25 июня 2019г. в 17:00 часов на заседании специализированного совета 037 “Информатика” Института проблем информатики и автоматизации НАН РА по адресу: 0014, г. Ереван, ул. П. Севака 1.

С диссертацией можно ознакомиться в библиотеке ИПИА НАН РА.
Автореферат разослан 17-го мая 2019г.

Ученый секретарь специализированного
совета, доктор физ.-мат. наук



А.Г. Саруханян

ԱՏԵՆԱԽՈՍՈՒԹՅԱՆ ԸՆԴՀԱՆՈՒՐ ԲՆՈՒԹԱԳԻՐԸ

Ատենախոսության թեմայի արդիականությունը

Արդի բյուրեղի վրա համակարգերում (բյուրեղի վրա համակարգ՝ System-on-Chip, SoC) տեղադրված հիշողության IP (Intellectual Property) միավորների խտության և հզորության արագ աճը թեստավորման և վերանորոգման ծախսերի պահպանման հետ կապված նոր խնդիրներ է առաջացնում՝ մինչույն ժամանակ նվազագույնի է մոտեցնում շուկա դուրս գալու ժամանակը¹: Առաջարկված բյուրեղի վրա IP ենթակառուցվածքը² առավելագույնի է հասցնում թեստավորման և վերանորոգման արդյունավետությունը՝ օգտագործելով հիշողության նախագծման վերաբերյալ գիտելիքները և ապահովելով անսարքությունների վերաբերյալ տվյալների վերլուծությունը: Հաշվի առնելով բյուրեղի վրա համակարգի նախագծման բարդության շարունակական աճը՝ խիստ կարևորվում է որ բյուրեղի վրա տեղադրված հիշողության թեստավորման և վերականգնման լուծումները համապատասխանեն տեխնոլոգիական առաջընթացին ապահովելով SoC-ի առաջընթացին համապատասխան որակը և արդյունավետությունը²:

Տեխնոլոգիական սարքերի չափերի փոքրացմանը զուգահեռ ի հայտ են եկել հիշողության ֆիզիկական դեֆեկտների (defects) նոր տեսակներ, որոնց համար հետսիլիկոնային վերլուծության ընթացքում դիտարկվել են հիշողության թեստավորման համար անհրաժեշտ անսարքությունների (faults) նոր մոդելներ: Դա իր հերթին բերել է բոլորը-մեկում մոտեցմամբ նախագծած SoC-ներում գետեղված հիշողությունների թեստավորման և ախտորոշման (test and diagnosis) խնդիրների նոր լուծումների անհրաժեշտությանը: Մենք հետևում ենք հիշողության սարքերի ներկառուցված ինքնաթեստավորման (Memory Built-in Self-Test MBIST) համակարգերի և ծրագրային հետսիլիկոնային (post-silicon) վերլուծության ավտոմատացման գործիքների միջև աշխատանքների բաշխման մոտեցմանը³, որոնցում թեստավորման և ախտորոշման տեղեկատվական հոսքի (այսուհետ հոսք, flow) ղեկավարումը և հսկողությունը կատարվում են ծրագրի միջոցով, մինչդեռ իրական արագությամբ (at-speed) հիմքային թեստավորման և ախտորոշման գործողությունները կատարվում են հիշողության ներկառուցված ինքնաթեստավորման համակարգերի կողմից: Այս մոտեցումն արտահայտվում է մի շարք լուծումներում, որոնք լայնորեն կիրառվում են կիսահաղորդչային արդյունաբերության մեջ^{3,4}:

Այս խնդրի ամբողջական լուծումն իրականացվում է ծրագրային և ապարատային կողմերի փոխագործակցությամբ՝ ծրագրային միջավայրում թեստային նմուշների (test

¹ “International Technology Roadmap for Semiconductors”, www.itrs2.net, 2015

² Shoukourian S., Vardanian V. and Zorian, “SoC yield optimization via an embedded-memory test and repair infrastructure,” IEEE Design & Test of Computers, vol. 21, no. 3, 2004, pp. 200-207

³ Zorian Y., Shoukourian S. “Test Solutions for Nanoscale Systems-on-Chip: Algorithms, Methods and Test Infrastructure”, Selected papers of Ninth International Conference on Computer Science and Information Technologies, IEEE, pp. 1-3, 2013

⁴ Au A., Pogiel A., Rajski J., Sydow P., Tyszer J., Zawada J. “Quality assurance in memory built-in self-test tools”, 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, 2014, pp. 39-44

patterns) ստեղծման, MBIST համակարգերում ստանդարտ ինտերֆեյսերի միջոցով դրանց կիրառման և ծրագրային միջավայրում MBIST համակարգերից ձեռք բերված արդյունքների/շղթաների (chains) վերլուծության միջոցով: Մեր նկատառումները հիմնված են համապարփակ թեստավորման և ավտորոշման հոսքի⁵ վրա, որը ներառում է նմանատիպ հոսքերի երեք գլխավոր փուլեր՝ անսարքության հայտնաբերում, տեղայնացում և դասակարգում: Հոսքի յուրաքանչյուր փուլ պահանջում է, որ ստեղծվեն հատուկ թեստային նմուշներ և վերլուծվեն այնպես, որ արդյունքները փոխանցվեն հաջորդ փուլի նախապատրաստման քայլին: Յուրաքանչյուր փուլում համապատասխան թեստային նմուշների նախագծման ժամանակ հատուկ Մարշ/Մարշ տիպի (March/March-like) թեստային ալգորիթմներ են ստեղծվում և օգտագործվում, որոնք իրենց հերթին փոխանցվում են թեստային նմուշներով MBIST համակարգին՝ իրական արագությամբ կատարման համար:

Քննարկված հոսքի յուրաքանչյուր փուլի համար վերոնշյալ նմուշները հարկավոր է ամբողջովին ստուգել մինչև MBIST համակարգերում դրանց կիրառումը՝ կարևոր ժամանակային և որակական կորուստները բացառելու համար: Որքանով մեզ հայտնի է շատ քիչ աշխատություններ կան որոնք վերաբերվում են MBIST համակարգերի հետսիլիկոնային վերլուծության ավտոմատացման գործիքների ստուգման մեթոդներին: Մասնավորապես այդ գործիքների ստուգման համար առաջարկվել է մոտեցում հիմնված MBIST ղեկավարող հանգույցին հարմարեցվող հիշող սարքի մոդելի վրա⁶, որը սակայն չի ապահովում ամբողջական MBIST համակարգերի հետ ծրագրային գործիքների աշխատանքի ստուգումը: Հաշվի առնելով որ հոսքի հետսիլիկոնային օգտագործումը գործարանում ժամանակատար և ծախսատար է՝ կարևորվում է ծրագրային գործիքներում թեստավորման և ավտորոշման համար իրականացված հոսքի ճշգրտության երաշխավորումը մինչև նրա կիրառումը արտադրված SoC-ի վրա⁵: Թեստային նմուշների ստուգման համար անհրաժեշտ է MBIST համարգերում առկա անսարքություններով իրականացված միջավայր, հիշողության MBIST համակարգերի վրա թեստային նմուշների կատարման ընթացքը մոդելավորելու համար: Հիշողության հանգույցներում հնարավոր ֆիզիկական դեֆեկտներին համապատասխանող անսարքությունների ճշգրիտ մոդելներ պետք է նկարագրվեն և օգտագործվեն միջավայրում, որը պետք է անպայման բավարարի հետևյալ 2 պահանջներին.

- անսարքության մոդելի գեներացիա և ներարկում (fault injection),
- թեստային նմուշների գեներացիայի (generation) և ստացված շղթաների վերլուծության ստուգման հնարավորություն:

Հաշվի առնելով որ գոյություն ունեցող MBIST համակարգերը հիմնականում նախապես նկարագրվում են ռեգիստրային փոխանցումների մակարդակում (Register Transfer Level RTL), բնական է ստուգման միջավայրը կառուցել համակարգի համապատասխան գոյություն

⁵ Harutyunyan G., Martirosyan S., Shoukourian S., Zorian Y. “Memory Physical Aware Multi-Level Fault Diagnosis Flow”, IEEE Transactions on Emerging Topics in Computing, 2018

⁶ Au A., Pogiel A., Rajski J., Sydow P., Tyszer J., Zawada J. “Quality assurance in memory built-in self-test tools”, 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, 2014, pp. 39-44

ունեցող ապարատային նկարգրման լեզվով (Hardware Description Language HDL)⁷ Verilog⁷ կամ VHDL⁸ իրականացման վրա:

FinFET (fin field-effect transistor) տեխնոլոգիայի նորամուծությամբ առաջացել են հիշողության ֆիզիկական դեֆեկտների նոր տեսակներ: Նախորդ հիշողություններին բնորոշ անսարքությունների համար նախագծված թեստավորման և ախտորոշման հոսքերը կիրառելի չեն նոր հիշողությունների համար, քանի որ ունակ չեն ապահովել անսարքությունների հայտնաբերման անհրաժեշտ ծածկույթ (coverage), ինչպես նաև պահանջում են հոսքերի փոփոխություններ հայտնաբերման, դասակարգման և տեղայնացման փուլերում: Ակնհայտ է նաև, որ ներկա նախագծերի համար մանրամասն մշակված լուծումներն ապագայում նույն խնդրի առջև կկանգնեն հիշողության նախագծերում շարունակական փոփոխությունների պատճառով, որոնք էլ տեխնոլոգիական սարքերի չափերի փոքրացման հետևանք են: Անսարքությունների նոր տեսակների կանխատեսման, թեստավորման, ինչպես նաև ախտորոշման համար պահանջվող լուծումներում փոփոխությունների բնական պահանջ է առաջացել: Անսարքությունների հատարկումից խուսափելու համար հետագա դիտարկումներում կօգտվենք բազմաչափ կանխատեսման մեխանիզմից⁹:

Աշխատանքի նպատակն ու խնդիրները

Ատենախոսության նպատակն է հիմնավորել և ստեղծել հիշողության թեստավորման և ախտորոշման տեղեկատվական հոսքի իրականացման համար թեստային նմուշների ստուգման միջավայր (verification environment)⁷ ընդլայնելով հիշողության ներկառուցված ինքնաթեստավորման համակարգի RTL ներկայացումը սխալների նկարագրման մոդելով: Այդ միջավայրը պետք է թույլ տա անսարքությունների առկայության պարագայում վերլուծել RTL ներկայացման վրա թեստային նմուշների կիրառման արդյունքում ստացված ելքային շղթաները՝ գեներացնելով, միջավայրի մեջ ներարկելով և մոդելավորելով հիշողության հետ կապված անսարքությունները:

Հետազոտության օբյեկտները

Հետազոտության օբյեկտներն են հիշողության հետ կապված անսարքությունների ավտոմատային մոդելները, հիշողության ներկառուցված ինքնաթեստավորման համակարգերի թեստային ալգորիթմներն ու նմուշները, ինչպես նաև անսարքության մոդելի գեներացիայի, ներարկման և թեստային նմուշների ստուգման մեթոդները:

Հետազոտության մեթոդները

Օգտագործվել են ավտոմատների տեսության, հիշող սարքերի թեստավորման և ախտորոշման, ինչպես նաև օբյեկտակողմնորշված նախագծման մեթոդներ:

⁷ “IEEE Standard for Verilog Hardware Description Language” in IEEE Std 1364-2005 (Revision of IEEE Std 1364-2001), 2006, pp.1-590

⁸ “IEEE Standard VHDL Language Reference Manual” in IEEE Std 1076-2008 (Revision of IEEE Std 1076-2002), vol., no., pp.c1-626, 26 Jan. 2009

⁹ Harutyunyan G., Shoukourian S., Zorian Y. “Fault Awareness for Memory BIST Architecture Shaped by Multidimensional Prediction Mechanism”, vol. 38, no. 3, pp. 562-575, Ma 2019

Արդյունքների գիտական նորությունը

- Անսարքությունների ընդլայնվող մոդել՝ հիշողության ներքին անսարքությունների համար, ինչպես նաև դրա իրականացումը RTL HDL մոդելավորման միջավայրում:
- Հիշողության ներքին անսարքությունների մոդելների համար ավտոմատացված գեներացիայի հոսք (generation flow)՝ հիմնվելով անսարքությունների պարբերականության աղյուսակի (Fault Periodicity Table FPT) վրա:
- Թեստային նմուշների ստուգման մեթոդ առկա անսարքություններով միջավայրում՝ հիմնված ստացված շղթաների վերլուծության և անսարքությունների մոդելների շրջանցված (traversed) գրաֆերի համեմատության վրա:
- Թեստավորման և ախտորոշման հոսքի իրականացման ստուգման մոտեցում:

Ստացված արդյունքների կիրառական նշանակությունը

Առաջարկված մոտեցումները թույլ են տալիս մեծացնել հետսիլիկոնային վերլուծության ավտոմատացման գործիքներում թեստավորման և ախտորոշման հոսքի իրականացման ճշգրտությունը և հուսալիությունը: Առաջարկված անսարքության մոդելավորման մեխանիզմը թույլ է տալիս մոդելավորել անսարքությունների նոր տեսակներ RTL HDL միջավայրում ըստ իրենց պարբերական հատկանիշների:

Ներդրումներ

Ատենախոսության արդյունքները իրականացվել են «Սինոփսիս» ընկերության հետսիլիկոնային վերլուծության ավտոմատացման գործիքում, որը ներկայումս լայնորեն կիրառվում է բազմաթիվ հաճախորդների կողմից:

Պաշտպանությանը ներկայացվում են հետևյալ դրույթները

- Հիշողության ներքին անսարքությունների ընդլայնվող մոդել RTL մակարդակի համար
- Անսարքությունների ներարկման մոտեցում RTL մակարդակի մոդելավորման միջավայրում
- Գեներացիայի հոսք հիշողության ներքին անսարքությունների մոդելների համար՝ հիմնվելով անսարքությունների պարբերականության աղյուսակի վրա
- Թեստային գեներացված նմուշների ստուգման մեթոդ
- Հետսիլիկոնային վերլուծության ավտոմատացման գործիքներում թեստավորման և ախտորոշման հոսքի իրականացման ստուգման մոտեցումներ

Ստացված արդյունքների ապրոբացիան

Աշխատանքի հիմնական արդյունքները և դրույթները զեկուցվել են մի շարք միջազգային գիտաժողովներում (CSIT 2017, EWDTs 2017, EWDTs 2018) ինչպես նաև ԵՊՀ Տեղեկատվական տեխնոլոգիաների կրթական և հետազոտական կենտրոնի և ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտի ընդհանուր սեմինարներում:

Ատենախոսության արդյունքների փորձարկումը և հրապարակումները

Աշխատանքի հիմնական արդյունքները տպագրված են 4 գիտական հոդվածներում:

Ատենախոսության կառուցվածքը և ծավալը

Ատենախոսությունը բաղկացած է ներածության բաժնից, 3 գլուխներից, եզրակացությունից, նկարների ցանկից, գրականության ցանկից և 1 հավելվածից: Աշխատանքի հիմնական մասը կազմում է 103 էջ, ունի 46 նկար և 19 աղյուսակ: Գրականության ցանկը ներառում է 88 աշխատություն: Հավելվածը պարունակում է «Սինոփսիս» ընկերությունում աշխատանքի արդյունքների ներդրման ակտը:

ԱՏԵՆԱԽՈՍՈՒԹՅԱՆ ՀԱՄԱՌՈՏ ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆԸ

Ներածություն բաժնում հիմնավորվում է թեմայի արդիականությանը, ձևակերպվում է աշխատանքի նպատակը, նորությունը, կիրառական նշանակությունը, ինչպես նաև հակիրճ մատնանշվում է աշխատանքի էությունը:

Առաջին գլխում ներկայացվում է SoC-ի ներդրված հիշողությունների հետսիլիկոնային ստուգման ընթացքում օգտագործվող թեստավորման և ախտորոշման հոսքը և առանձնացվում են տվյալ ատենախոսության խնդիրները:

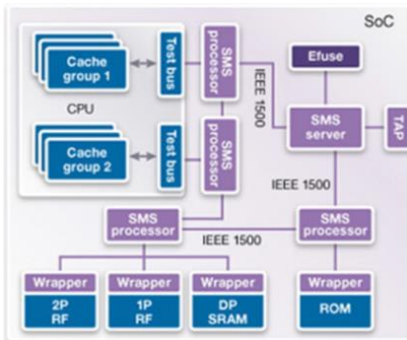
1.1 ենթագլխում դիտարկվում են հետսիլիկոնային անալիզի ծրագրային գործիքներում թեստավորման և ախտորոշման հոսքերի իրականացման խնդիրները: Ներդրված հիշողությունների համար հետսիլիկոնային թեստավորման և ախտորոշման հոսքերի հիմնական նպատակն է SoC-երի բարձր որակի և արդյունավետության (yield) երաշխավորումը: Գոյություն ունեն բազմաթիվ մոտեցումներ, որոնց նպատակն է լուծել հիմնականում հետևյալ երեք խնդիրները՝ անսարքության հայտնաբերում, տեղայնացում և դասակարգում: Առաջարկված թեստավորման և ախտորոշման լուծումները հիմնված են հիշողության ներկառուցված ենթակառուցվածքների՝ MBIST համակարգերի, վրա, որոնք գործարկվում են, վերոնշյալ երեք խնդիրները ամբողջովին ծածկող, տարաբնույթ թեստային նմուշների միջոցով: Քանի որ կիսահաղորդչային ընկերությունները ջանք չեն խնայում, որպեսզի արտադրության ժամանակ կատարվող թեստավորումը դարձնեն ֆինանսապես և ժամանակային արդյունավետ¹⁰, թեստային նմուշները հիմնականում ստեղծվում են SoC-ների նախասիլիկոնային մշակման (pre-silicon development) փուլում՝ բազմազան դեպքերի դիտարկման նպատակով: Դրանք, ըստ էության, կախված են ընտրված թեստավորման և ախտորոշման հոսքից: Մեր հետազոտությունների արդյունքում մենք հանգել ենք հետևյալ թեստավորման և ախտորոշման հոսքին¹¹, որը իր մեջ նեռարում է վերոնշյալ երեք խնդիրների լուծումները: Այն բաղկացած է հետևյալ 7 փուլերից՝

1. Անսարք (failed) հիշող հանգույցների նույնականացում (identification) MBIST համակարգի միջոցով:
2. Հիշող հանգույցների անսարքությունների տրամաբանական հասցեների ստացում:
3. Անսարքությունների ֆիզիկական հասցեների նույնականացումը, հիմնվելով տրամաբանական հասցեների և հիշողության խճողման (scrambling) վրա:

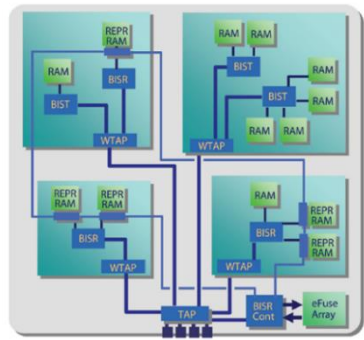
¹⁰ Lee Y., Choi I., Oh K., Ko J. J. and Kang S., "Test item priority estimation for high parallel test efficiency under ATE debug time constraints," *2017 International Test Conference in Asia (ITC-Asia)*, Taipei, 2017, pp. 150-154

¹¹ Harutyunyan, Martirosyan, Shoukourian, Zorian "Memory Physical Aware Multi-Level Fault Diagnosis Flow"

4. Անսարքության բջիջների ֆիզիկական X և Y կորդինատների նույնականացում, հիմնվելով դրանց ֆիզիկական հասցեների և հիշողության խճողման վրա:
5. Դեֆեկտների դասակարգում հիմնվելով հիշող հանգույցում դրանց տեղավորման (placement) վրա:
6. Անսարքությունների դասակարգում՝ 2-րդ քայլում ստացված անսարքությունների տրամաբանակ հասցեներով անհատական թեստավորման և թեստային սինդրոմների վերլուծության միջոցով:
7. Հիշող հանգույցներում անսարք բջիջների տեղայնացում (localization)՝ հիմնվելով հիշողության խճողման վրա, երկբջջանի անսարքությունների համար:



ա



բ.

Նկար 1. MBIST կոմբինացիոն լուծումների օրինակներ ա. DesignWare STAR Memory System (Synopsys Inc.) բ. Tessent Memory Test (Mentor, a Siemens Business)

Հետսիլիկոնային անալիզի ծրագրային գործիքներում հոսքի յուրաքանչյուր փուլ օգտագործում է նախորդ փուլերում արտաձված տվյալները: 1, 2, 5, 6, 7 քայլերը պահանջում են թեստային նմուշների ստեղծում, MBIST համակարգում դրանց կիրառում, ինչպես նաև MBIST համակարգից ստացված արդյունքների վերլուծություն:

1.2 ենթագլխում նկարագրվում են հիշողության ներկառուցված ինքնաթեստավորման (MBIST) համակարգեր: MBIST-երը ներկայացնում են հատուկ ենթակառուցվածքներ, որոնք ներկառուցված են SoC-ներում և օգտագործվում են հիշող սարքերի իրական արագությամբ թեստավորման համար: Նրանք գործարկվում են թեստային ալգորիթմների միջոցով, որոնք հրահանգում են դրանց, թե ինչպես հիշողության բջիջների վրա կատարել կարդալու և գրելու գործողությունները: MBIST-երը կարող են նախագծվել այնպես, որ օգտագործեն ապարատային (hardwired) թեստավորման ալգորիթմներ կամ դրսից ծրագրավորվելու ունակություն ունենան: Պրոցեսորի վրա հիմնված ժամանակակից MBIST համակարգերը խմբավորում են հիշող հանգույցները հիերարխիկ ցանցերում, այսուհետ՝ MBIST ցանցեր (MBIST networks), և ունակ են ապահովել հիշող հանգույցների զուգահեռ արագ թեստավորում՝ MBIST ենթաբաղադրիչների միջոցով: Հիերարխիայի բարդության

աստիճանը շարունակ մեծանում է՝ ներառելով հիերարխիայի ավելի շատ ենթաբաղադրիչներ և ավելի շատ մակարդակներ (Նկար 1):

Մեր աշխատանքում կոդիտարկվեն միկրոկոդի վրա հիմնված (microcode-based) ծրագրավորվող MBIST ցանցերը, քանի որ դրանք ապահովում են թեստավորման հավելյալ ճկունություն և անսարքությունների լուսաբանման ավելի բարձր ունակություն:

1.3 ենթազխում նկարագրվում են թեստային ալգորիթմները: Ներդրված հիշողությունները թեստավորվում են գրել/կարդալ գործողությունների հաջորդական կիրառմամբ (թեստային ալգորիթմներ)՝ MBIST բաղադրիչների կողմից հիշողության բջիջների վրա: Գոյություն ունեն թեստային ալգորիթմների տարբեր տեսակներ՝ (Մարշ թեստեր (March test), GALPAT, Walking 0/1 և այլն), սակայն Մարշ թեստերը համարվում են ամենից արդյունավետը, քանի որ դրանց բարդությունը գծային է $O(n)$: Մարշ թեստերը կազմված են Մարշ տարրերի վերջավոր թվից $M = \{M_1, M_2, \dots, M_k\}$, որտեղ յուրաքանչյուր Մարշ տարր $M_i = A_i(O_1, O_2, \dots, O_m)$ բաղկացած է՝ $A_i \in \{\uparrow, \downarrow, \updownarrow\}$ հասցեավորման ուղղությունից, $O_j \in \{R(0), R(1), W(0), W(1)\}$ – վերջավոր թվով կարդալու/գրելու գործողություններից: Մարշ տիպի ալգորիթմներ են առաջարկվում առանձին դասերի անսարքությունների հայտնաբերման և տեղայնացման համար:

Առաջարկված թեստային ալգորիթմների նոտացիան (notation) պարբերաբար ընդլայնվում է, որպեսզի ապահովվի անսարքությունների հայտաբերելիության ավելի լայն ծածկույթ¹²:

Տեխնոլոգիական սարքերի չափերի փոքրացման հետ զուգահեռ նոր Մարշ/Մարշ տիպի թեստային ալգորիթմներ են նախագծվում՝ անսարքությունների նոր տեսակների հետ կապված հիշող սարքերը թեստավորելու և ախտորոշելու համար: Թեստային ալգորիթմները ծրագրավորվում են MBIST ցանցի MBIST-երում՝ թեստային նմուշների միջոցով:

1.4 ենթազխում նկարագրվում են թեստային նմուշները և ներկայացվում ելքային շղթաների վերլուծության գործընթացը: Թեստային նմուշները պարզապես MBIST ցանցերի մուտքային և ակնկալվող ելքային շղթաների հաջորդական ներկայացումներն են: Դրանց հիմնական նպատակն է ուղարկել թեստային ալգորիթմները MBIST ցանցի համապատասխան ենթաբաղադրիչներին՝ հաղորդելով թեստային ալգորիթմը հիերարխիայի միջով: Թեստավորման սարքերը կամ ծրագրային գործիքները գործարկում են MBIST ցանցը ստանդարտացված թեստային ինտերֆեյսերի միջոցով (IEEE-1687, IEEE-1500, IEEE 1149.1 ստանդարտներ): Մուտքային շղթաները ներմուծվում են MBIST ցանց՝ օգտվելով ծայրամասային սկանավորման (boundary scan) մեխանիզմից: Արդյունքում ցանցից ստացվում է նույն երկարության ելքային շղթա:

Ծրագրային գործիքները շարունակ փոփոխության են ենթարկվում, որպեսզի աջակցեն նոր թեստային նմուշներին՝ ստեղծված նոր թեստային ալգորիթմների և MBIST ցանցերի բարդության աճին համապատասխան: Ավելին, այդ գործիքների վերլուծության և

¹² Aleksanyan K., Amirkhanyan K., Shoukourian S., Zorian Y., “Memory modeling using an intermediate level structural description”, US Patent 07768840, issued on August 3, 2010

հաղորդման ընթացակարգերը նույնպես անհրաժեշտ է բարելավել՝ անսարքությունների նոր տեսակների հետ աշխատանքը ապահովելու համար:

Հետևաբար թեստային նմուշների մուտքային շղթաների ձևավորման և ելքային շղթաների վերլուծության ընթացակարգերը պետք է ստուգվեն, նախքան դրանք կկիրառվեն իրական բյուրեղի վրա: Ստուգումը պահանջում է հիշողության հանգույցներում դեֆեկտներ/անսարքություններ որոշելու ունակությամբ ճշգրիտ MBIST մոդել:

1.5 ենթագլխում ցույց են տրվում հիշողության ներքին դեֆեկտների և անսարքությունների մոդելավորման առկա մոտեցումները: Մենք դիտարկում ենք SoC-ներում ներկառուցված պատահական մուտքի հիշողությունները (Random Access Memory RAM) քանի որ դրանք ամենատարածվածն են: RAM հիշող սարքերը կազմված են տարբեր բաղադրիչներից՝ որոնցից է նաև հիշող բջիջների զանգվածը (memory cell array): Հիշողության բջիջները՝ կախված կիրառությունից իրենց հերթին նույնպես բաղկացած են տարբեր կոմպոնենտներից՝ օրինակ SRAM սարքերի հիշողության բջիջները հիմնականում կազմված են 6 տրանզիստորներից: Այդուհանդերձ, անկախ կիրառությունից, բջիջները պետք է ունակ լինեն պահել/պահպանել 0 կամ 1 արժեքներ, որը կարող կարողացվել համապատասխան գործողությամբ և/կամ փոփոխվել գրելու/վերականգման (reset) գործընթացներով:

Չիպի արտադրության ընթացքում, հիշողության բջիջներում ժամանակ առ ժամանակ առաջանում են դեֆեկտներ որոնք հանգեցնում են բջիջների փոփոխված սխալ վարքին՝ հիմնված դեֆեկտի տեսակի և տեղակայման վրա: Հիշողության բջիջների նման վարքի աբստրակցիա է առաջարկվել անսարքության պրիմիտիվների (Fault Primitives FP) միջոցով՝ օգտագործելով համապատասխան $\langle S/F/R \rangle$ նոտացիան¹³, որտեղ S-ը հիշողության բջիջների վրա կիրառվող, անսարքությունը գրգռող (sensitizing) գործողությունների հաջորդականությունն է, F - ը՝ զոհ բջջի վիճակի արժեքն է, այն բանից հետո երբ անսարքությունը ակտիվացել է, R - ը հիշողության զոհ բջջի վրա դիտարկվող արժեքն է, եթե S-ի վերջին գործողությունը «կարդալ» է: Սխալ վարքը կարող է նաև ներառել երկու հիշողության բջիջներ՝ երկբջջանի (coupling) անսարքություններ: Դիտարկվել է նաև անսարքությունների ավելի բարդ վարքագիծ՝ որը ներառում է բազմաթիվ անսարքության պրիմիտիվներ, որոնք ազդելով միևնույն հիշողության բջջի վրա քողարկում են միմյանց սխալ վարքը՝ կապակցված (linked) անսարքություններ:

1.6 ենթագլխում բերվում է ատենախոսության խնդրի ձևակերպումը: MBIST-երի թեստավորման և ախտորոշման գործընթացի ծրագրային գործիկներում իրականացումը ստուգող ներկա դրությամբ սակավաթիվ աշխատանքներ կան: Մասնավորապես ստուգվում է MBIST-երի թեստավորման և ախտորոշման գործընթացը¹⁴, սակայն MBIST-երը չեն դիտարկվում ամբողջական համակարգի շրջանակներում:

¹³ van de Goor A. J. and Al-Ars Z., "Functional memory faults: a formal notation and a taxonomy," Proceedings 18th IEEE VLSI Test Symposium, Montreal, Quebec, Canada, 2000, pp. 281-289.

¹⁴ Au A., Pogiel A., Rajski J., Sydow P., Tyszer J., Zawada J. "Quality assurance in memory built-in self-test tools", 17th International Symposium on Design and Diagnostics of Electronic Circuits & Systems, 2014, pp. 39-44

Ամփոփելով նոր թեստային նմուշների և համապատասխան թեստային ալգորիթմների ստեղծման անհրաժեշտությունը, հաղորդման և վերլուծությունների նոր ընթացակարգերը, MBIST ցանցերի աճող բարդությունը և հիշողության հետ կապված անսարքությունների նոր տեսակները, ստուգման միջավայրի կառուցման անհրաժեշտություն է առաջանում, որը թույլ կտա տվյալ անսարքության մոդելի կամ անսարքության մոդելների բազմության ներարկում RTL մոդելում, որը նկարագրում է համապատասխան SoC-ին, հնարավոր է դարձնում ստուգել թեստային նմուշի կատարումը և մուտքային/ելքային շղթաների վերլուծությունը: Հաջորդող գրվածները նվիրված են թվարկված խնդիրների համար լուծումներ գտնելուն:

Երկրորդ գլխում ներկայացվում է անսարքությունների պրիմիտիվների համար անսարքությունների ավտոմատային մոդելների գեներացիայի ընթացակարգերը՝ միաբջիջ (single-cell), երկբջջանի և կապակցված անսարքությունների համար: Պարամետրացված FDT-ն, ինչպես նաև պարամետրացված FDT-երի գեներացիայի ավտոմատացված հոսքը անսարքությունների մոդելների համար ներկայացվում են հետագայում: Ի վերջո մատնանշվում են MBIST ցանցի HDL ներկայացման մեջ անսարքության մոդելի իրականացման համար անհրաժեշտ պահանջները:

2.1 ենթագլխում ներկայացվում է հիշողության անսարքությունների ավտոմատային մոդելը: Անսարքության մոդելի ներկայացումը հիմնված է դետերմինացված վերջավոր ավտոմատի¹⁵ վրա (Deterministic Finite Automata DFA): DFA-ն սահմանվում է որպես հնգյակ՝

$$A = (Q, \Sigma, \delta, q_0, F)$$

որում՝ Q -ն DFA վիճակների վերջավոր բազմությունն է, Σ -ն մուտքային սիմվոլների վերջավոր բազմությունն է, $\delta(q, a) = q'$ անցումների ֆունկցիան է, q_0 սկզբնական վիճակն է, F վերջնական վիճակների բազմությունն է և $F \in Q$:

Հնգյակի տարրերը սահմանվում են անսարքության մոդելի համար հետևյալ կերպ՝

1. քանի որ FP-ն սահմանվում է n հիշողության բջիջների համար՝ $n-1$ ագրեսոր (aggressor) բջիջներով և 1 զոհի (victim) բջջով, յուրաքանչյուր $q \in Q$ ներկայացվում է որպես $(n+1)$ -յակ $q := (R, S_1, \dots, S_n)$, որում $n-1$ FP-ի կողմից ներգործվող բջիջների թիվն է, R -ը զոհ բջջի վրա «կարդալ» գործողությամբ վերադարձվող արժեքն է, եթե այն կիրառվում է q վիճակում,
2. յուրաքանչյուր $a \in \Sigma$ գործողություն է, որը կարող է կիրառվել n բջիջներից մեկի վրա: Եթե FP-ի կողմից երկու կամ ավել բջիջներ են ներգործվում, ապա գործողությունները անհրաժեշտ է տարբերակել: Օրինակ, R կարդալու գործողությունը կարող է համարակարգվել n բջիջներից յուրաքանչյուրի համար Σ -ում հետևյալ կերպ՝ R_1, R_2, \dots, R_n ,
3. $\delta(q, a) = q'$ անցում է $(n+1)$ -յակ q -ից q' n բջիջներից մեկի վրա կիրառվող a գործողության արդյունքում,
4. q_0 ցույց է տալիս DFA-ի նախնական վիճակը, երբ բոլոր n բջիջները դեռ սկզբնառժեքավորված չեն,

¹⁵ Sipser M. "Introduction to the Theory of Computation", 2nd edition, 2006

5. F բազմությունը բաղկացած է մեկ վերջնական վիճակից, որը համապատասխանում է անսարքության ակտիվացմանը:

Մոդելի ընդհանուր կառուցվածքը կարելի է բաժանել չորս մասերի՝ սկզբնական վիճակ, սկզբնաթժեքավորման բլոկ, անսարքության վարքային բլոկ և անսարքության ակտիվացման վիճակ:

Ստորև ներկայացվում FDT աղյուսակը (Աղյուսակ 1), որը գեներացվում է վերջավոր ավտոմատների աղյուսակային ներկայացման հիման վրա: Աղյուսակի համար կօգտագործվի երկուական ներկայացում, որպեսզի հեշտացվի անսարքության մոդելի հետագա կիրառումը:

Աղյուսակ 1. Անսարքությունների Նկարագրության Աղյուսակ (FDT)

R	S ₁	...	S _n	a ₁	...	a _k
R ⁰	S ₁ ⁰	...	S _n ⁰	$\delta(q_0, a_1)_{bin}$...	$\delta(q_0, a_k)_{bin}$
...
R ^m	S ₁ ^m	...	S _n ^m	$\delta(q_m, a_1)_{bin}$...	$\delta(q_m, a_k)_{bin}$

Ներկայացված DFA մոդելը կարող է հեշտորեն փոխակերպվել Միլիի վիճակների ավտոմատի (Mealy state machine)՝ ավելացնելով $\Omega = \{-, 0, 1\}$ արտածման այբուբեն (output alphabet), և λ արտածման ֆունկցիա (output function) $\lambda: Q \times \Sigma \rightarrow \Omega, \forall q \in Q \text{ և } \forall a \in \Sigma$ համար:

Միլիի վիճակների ավտոմատի ներկայացումը կօգտագործվի RTL-ում անսարքության մոդելի իրականացման ընթացքում:

2.2 ենթաբաժնում նկարգվում են միաբջիջ, երկբջջանի և կապակցված անսարքությունների համար մոդելների գեներացիայի մշակված ընդացակարգերը:

ա) Միաբջիջ անսարքությունների համար անսարքության ավտոմատային մոդելը հիմնված է $\langle S/F/R \rangle$ նոտացիայի վրա: Ենթադրելով որ գործողությունների թիվը S -ում n է, անսարքությունների մոդելը ընդհանուր դեպքում կարող է ներկայացվել հետևյալ կերպ՝

$$DFA_{\text{միաբջիջ}} = (Q, \Sigma, \delta, q_0, F)$$

որում՝ $Q = q_0 \cup \{q_i := (R\alpha, S\alpha) : \alpha \in \{0, 1\}, i = 1, \dots, n+1\}$ անսարքության վարքային-բլոկ $\cup \{q \text{ անսարքության-ակտիվացման-վիճակ} := (R\alpha, S\beta) : \alpha \in \{0, 1\}, \beta \in \{0, 1\}\}$, $\Sigma = \{W(0), W(1), R, \text{Reset}\}$, $\lambda: Q \times \Sigma \rightarrow Q, \forall q \in Q$ և $\forall a \in \Sigma$ համար, $q_0 := (RX, SX)$, $F = \{q \text{ անսարքության-ակտիվացման-վիճակ}\}$:

Մոդելում DFA-ի յուրաքանչյուր վիճակ զույգ է (R, S) , որտեղ R -ը հիշողության բջջի վրա կիրառվող «կարդալու» գործողության վերադարձվող արժեքն է, մինչդեռ S -ն այդ վիճակում հիշողության բջջի իրական արժեքն է:

Ներկայացվում է FDT-ի գեներացիայի ընթացակարգը՝ $\langle S/F/R \rangle$ նոտացիայով միաբջիջ FP-ի համար: Ընթացակարգը FDT-ում ավելացնում է նոր տողեր՝ S -ում յուրաքանչյուր գործողության համար, ինչպես նաև վերակարգավորում է նախորդող տողերում անցումները անսարքության վարքը ճշգրիտ պատկերելու համար:

բ) Երկբջջանի անսարքության ավտոմատային մոդելը հիմնված է $\langle S_a; S_v/F/R \rangle$ նոտացիայի վրա: Ենթադրելով որ S_a և S_v -ում գործողությունների ընդհանուր թիվը n է, անսարքությունների մոդելը ընդհանուր դեպքում կարող է ներկայացվել հետևյալ կերպ՝

$$DFA_{\text{երկբջջանի}} = (Q, \Sigma, \delta, q_0, F)$$

որում $Q = q_0 \cup \{(RX, S_0, S_iX), (RX, S_01, S_iX), (R1, S_0X, S_i1), (R0, S_0X, S_i0)\}$ սկզբնավորման-բյուր $\cup \{q_i := (R\alpha, S_0\beta, S_i\gamma) : \alpha \in \{0, 1\}, \beta \in \{0, 1\}, i = 1, \dots, n+4\}$ անսարքության-վարչային-բյուր $\cup \{q_{անսարքության-ակտիվացման-վիճակ} := (R\alpha, S_0\beta, S_i\gamma) : \alpha \in \{0, 1\}, \beta \in \{0, 1\}, \gamma \in \{0, 1\}\}$, $\Sigma = \{W_0(0), W_0(1), R_0, W_i(0), W_i(1), R_i, Reset\}$, $\lambda: Q \times \Sigma \rightarrow Q, \forall q \in Q \text{ և } \forall a \in \Sigma \text{ համար, } q_0 := (RX, S_0X, S_iX), F = \{q_{անսարքության-ակտիվացման-վիճակ}\}$:

DFA-ի յուրաքանչյուր վիճակ ներկայացված է եռյակի տեսքով (R, S_0, S_i) , որում R -ը զոհ բջջի վրա կիրառվող «կարդալ» գործողության կողմից վերադարձվող արժեքն է, S_0 -ն ագրեսոր բջջի արժեքն է, S_i -ը DFA-ի ընթացիկ վիճակում զոհ բջջի իրական արժեքն է: Ագրեսոր բջջի «կարդալ» արժեքը ներառված չէ, քանի որ այն նույն S_0 -ն է:

Նկարագրվում է FDT-ի գեներացիայի ընթացակարգը, որը սակայն ունի որոշ տարբերություններ միաբջիջ անսարքության ընթացակարգի համեմատ:

գ) Արդի հիշողության նախագծերում առկա դիտարկված իրական կապակցված անսարքությունները ներառում են առավելագույնը 3 բջիջ, որտեղ FP_1 և FP_2 -ը կամ միաբջիջ են կամ երկբջջանի: Յույց է տրվում որ A_{FP_1} -ն և A_{FP_2} -ն որոշ փոփոխություններից հետո հնարավոր է միավորել՝

$$DFA \text{ Ակապակցված} = A_{FP_1} \cup A_{FP_2}$$

Արտածված ավտոմատային մոդելի FDT գեներացիայի ընթացակարգը և համապատասխան Միլլիի ավտոմատի աշխատանքի մեկնաբանումը նույնպես ներկայացված է:

Աղյուսակ 2. Պարամետրացված FDT $\langle x/\sim x/\rangle$ համար

R^s	S^s	R	W(t)	W(T)	Reset
X	X	0010	0010	0010	0010
t	T	0001	0001	0010	0001
T	T	0010	0001	0010	0010

Հիմնվելով FPT-ում այն դիտարկման վրա, որ FP-ները համաչափ (symmetric) հատկություններ ունեն⁶ (այն է՝ $\langle 0/1/\rangle$ և $\langle 1/0/\rangle$ համար համաչափ նոտացիան $\langle x/\sim x/\rangle$ է), առաջարկվում է պարամետրացված (parametrized) FDT աղյուսակային ներկայացումը: Միաբջիջ և երկբջջանի անսարքությունների համար FDT-երի գեներացիայի ընթացակարգերը մի փոքր փոփոխվում են՝ համաչափ FP-ների համար ընդհանուր պարամետրացված FDT կառուցման նպատակով, այն է՝ վերոնշյալ օրինակի համար FDT-ն ցուցադրված է Աղյուսակ 2-ում: Ըստ այդմ, օգտագործելով փոփոխված ընթացակարգերը հնարավոր է դառնում գեներացնել պարամետրացված FDT աղյուսակ նաև կապակցված անսարքությունների համար: Պարամետրացված FDT-երի օգտագործումը էականորեն նվազեցնում է գեներացվող ավտոմատային մոդելների քանակը, նամանավանդ համաչափ բազմաբջջանոց անսարքությունների դեպքում:

2.3 ենթադրում նկարագրվում է պարամետրացված FDT-երի գեներացիայի ավտոմատացված հոսքը: Ներկայացվում է անսարքության մոդելավորման համար

¹⁶ Harutyunyan G., Shoukourian S., Zorian Y. "Fault Awareness for Memory BIST Architecture Shaped by Multidimensional Prediction Mechanism", vol. 38, no. 3, pp. 562-575, Ma 2019

ընտրույթի (exhaustion) նվազեցմանն ուղղված պարամետրացված FDT գեներացիայի ավտոմատացված հոսքը: Հոսքը հիմնված է FPT-ի վրա, և, այսպիսով, քննարկում է միայն հայտնի իրական անսարքությունները: Ի հավելումն, պարամետրացված FDT-երը կարող են պահպանվել հետագա օգտագործման համար:

Նախ և առաջ, կապակցված անսարքությունները խմբավորվում են գոյություն ունեցող FPT ներկայացման մեջ, որպեսզի օգտագործվեն հոսքում: Կապակցված և չկապակցված (unlinked) անսարքությունների համար դիտարկվում են երկու ենթահոսք (sub-flows): Երկու ենթահոսքերն էլ օգտագործում են համաչափ FP-երի լուծման (parsing) ինտերակտիվ ընթացակարգը, որը որպես մուտք ստացած FPT աղյուսակի վանդակի համար կոնկրետացնում է անսարքության նոտացիան հիմնվելով որոշ լրացուցիչ մուտքային տվյալների վրա:

Հոսքը օգտագործումը նկարագրվում է հաջորդ գլխում՝ պարամետրացված FDT-երի գեներացիայի, պահպանման և վերստին օգտագործման համար:

2.4 ենթագլխում բացահայտվում են հիշողության ներկառուցված ինքնաթեստավորման ցանցի HDL ներկայացման մեջ անսարքության մոդելի իրականացման նկատմամբ պահանջները: Իրական հիշողության RTL ներկայացումը միշտ չէ, որ առկա է MBIST ցանցի նախագծման ընթացքում: Փոխարենն օգտագործվում են հիշող հանգույցների վիրտուալ մոդելներ: Քանի որ անսարքության մոդելը չի կարող օգտագործվել՝ ուղղակիորեն հիշողության բջիջը փոխարինելու համար, կարելի է այն տեղադրել հիշող սարքից դուրս՝ կապելով հիշող հանգույցի մուտքերի/ելքերի հետ: Անսարքության մոդելը պետք է հնավորություն ունենա գործարկել FDT-ն: Անսարքության մոդելի իրականացման հետ կապված պահանջները հստակեցվում են:

Երրորդ գլխում ներկայացվում է անսարքությունների ներարկման հոսքը: Այնուհետև կառուցվում է ստուգման միջավայրը և ցույց է տրվում դրա կիրառությունը թեստավորման և ախտորոշման հոսքի ստուգման համար: Միջավայրը փորձնականորեն հիմնավորվում է: Վերջապես, հիմնվելով բազմաչափ կանխատեսման մեխանիզմի վրա, առաջարկվում են լավացումներ թեստավորման և ախտորոշման հոսքի իրականացման համար, որոնք հեշտացնում են նաև այդ իրականացման ստուգումը:

3.1 ենթագլխում նկարագրվում են անսարքության մոդելի ղեկավարող սարքը (controller) և անսարքության ներարկման հոսքը: Քանի որ անսարքության մոդելը ներկայացվում է FDT-ի միջոցով, պահանջվում է ղեկավարող սարք որը կկազմակերպի աղյուսակի վիճակների անցումները: Ղեկավարող սարքը կապվում է հիշողության փիների (memory pins) հետ և աշխատում է այն պահին, երբ դիմում է կատարվում հիշողության փիների միջոցով ներարկված անսարքության հասցեով: Հիշող հանգույցի վրա անսարքության հասցեով «կարդալ» գործողության կատարման դեպքում անսարքության ղեկավարող սարքը հիշողության ելքային փիներին վերագրում է անսարքության ավտոմատային մոդելի ընթացիկ վիճակի համատասխան արժեքը:

Անսարքության ներդրման հոսքը հենվում է ավտոմատ պարամետրացված FDT-երի գեներացիայի հոսքի վրա, որպեսզի գեներացնի համաչափ անսարքությունների համար պարամետրացված FDT-երի շտեմարան: FDT-երի մեջ գեներացիայի փուլում ավելացվում են

լրացուցիչ սյունակներ՝ դրոշներ (F), անսարքության մոդելի վարքի հետագա վերլուծության համար (Աղյուսակներ 3, 4):

Աղյուսակ 3. Երկբջջանի անսարքության փոփոխված տող

R ^s	S ^s _a	S ^s _v	R _a	F	W _a (t ₁)	F	W _a (T ₁)	F	R _v	F	W _v (t ₂)	F	W _v (T ₂)	F	Reset	F
----------------	-----------------------------	-----------------------------	----------------	---	----------------------------------	---	----------------------------------	---	----------------	---	----------------------------------	---	----------------------------------	---	-------	---

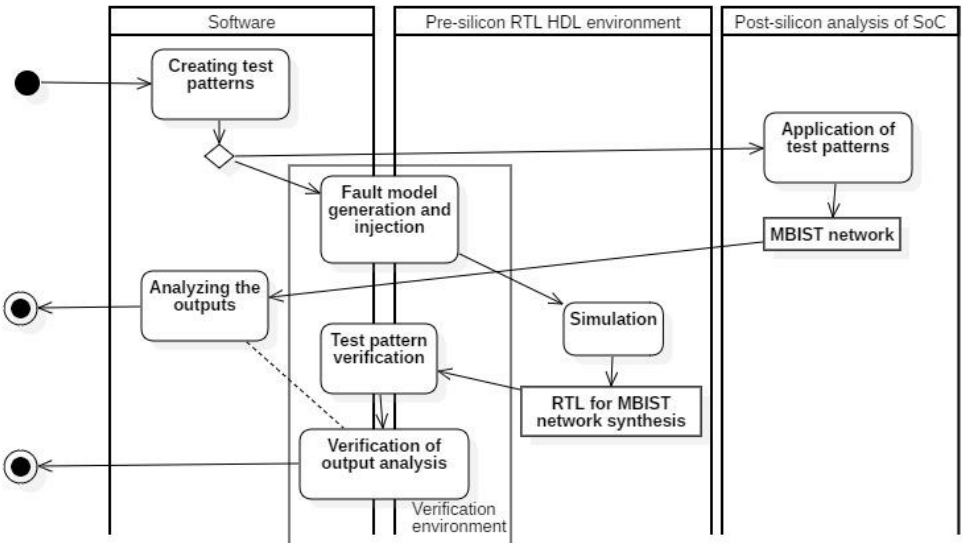
Աղյուսակ 4. Միաբջիջ անսարքության փոփոխված տող

R ^s	S ^s	R	F	W(t)	F	W(T)	F	Reset	F
----------------	----------------	---	---	------	---	------	---	-------	---

Ներարկրման հոսքի շրջանակում՝ հիմնվելով անսարքության ներարկրման վերաբերյալ տեղեկության վրա, համապատասխան անսարքության պարամետրացված FDT-ն փնտրվում է շտեմարանում: Պարամետրացված FDT-ն հայտնաբերելու դեպքում այն արժեքավորվում է (initialized) անսարքության համապատասխան: Հակառակ դեպքում անսարքության համար գեներացվում է FDT՝ համապատասխան ընթացակարգի օգտագործման հիման վրա: Հիշող հանգույցի կոնֆիգուրացիայի (configuration) տեղեկության հիման վրա կազմվում է անսարքության մոդելի ղեկավարող սարքը և ներարկվում RTL HDL-ում:

Սկարագրված անսարքության ներարկման հոսքը հետագայում օգտագործվում է թեստավորման և ախտորոշման ընթացակարգի կիրառման ստուգման միջավայրում:

3.2 ենթագլխում նկարագրվում է ստուգման միջավայրի իրականացումը: Այն (Նկար 2) օգտագործում է MBIST ցանցի RTL մոդելը՝ որը սովորաբար ստեղծվում է համակարգի մշակման ժամանակ և բաղկացած է 3 մասից՝ անսարքության մոդելի գեներացիա և ներարկում, թեստային նմուշների ստուգում և շղթաների վերլուծության ստուգում:



Նկար 2. Ստուգման միջավայր

Անարքության մոդելների գեներացիայի և ներարկման համար օգտագործվում է անսարքությունների ներարկման վերոնշյալ հոսքը:

Հաշվի առնելով, որ թեստային նմուշները հրահանգում են հիշողության ներկառուցված ինքնաթեստավորման համակարգին աշխատեցնել Մարշ/Մարշ տիպի թեստային ալգորիթմները հիշող հանգույցների վրա, ստուգվում է՝ արդյոք ճիշտ է կազմվել ալգորիթմի նկարագրման շղթան և արդյոք նպատակային հիշող հանգույցի վրա է այն կիրառվել: Թեստային նմուշի ստուգման ժամանակ օգտագործվում է անսարքության մոդելի տվյալ թեստային ալգորիթմի կողմից ավտոմատի նախապես շրջանցված գրաֆը՝ MBIST ցանցի RTL իրականացումից անկախ ծրագրային միջավայրում, համեմատելով մոդելավորման արդյունքում ստացված գրաֆի շրջանցման հետ:

Թեստային նմուշի կիրառման արդյունքում ստացված ելքային շղթաների վերլուծության ստուգումը կազմակերպելու համար օգտագործվում է անսարքությունների ներարկման վերաբերյալ տեղեկությունը՝ անսարքությամբ հիշող հանգույցի ցանցում տեղը, անսարքության տեսակը՝ նոտացիան, և հասցեն: Թեստավորման և ախտորոշման հոսքի յուրաքանչյուր փուլում ելքային շղթաների վերլուծությունը կատարվում է յուրովի:

3.3 ենթազվխում նկարագրվում է թեստավորման և ախտորոշման հոսքի իրականացման ստուգումը: Քանի որ այս բազմափուլ հոսքը օգտագործվում է հետսիլիկոնային անալիզի մեջ, և յուրաքանչյուր փուլում թեստային նմուշի ստեղծումը կախված է նախորդ փուլում թեստային նմուշի կատարման արդյունքներից, ապա այս հոսքի իրականացման ստուգումը էական է: Ներկայացվում են հոսքի անսարքությունների հայտնաբերման, տեղայնացման, դասակարգման և անսարքությունների ագրեսոր բջջի տեղայնացման փուլերի ստուգումը ստեղծված միջավայրում:

3.4 ենթազվխում կատարվում է ստուգման միջավայրի աշխատանքի հիմնավորումը: Ստուգման միջավայրի աշխատանքը հիմնավորվել է որոշակի թեստային ալգորիթմների օգնությամբ, որոնք հայտնաբերում են արդի հիշող սարքերի հայտնի բոլոր ներքին անսարքությունները: Միջավայրի հիմնավորումը կատարվել է տարբեր բարդության MBIST ցանցերի ավելի քան 600 կոնֆիգուրացիաների վրա՝ օգտագործելով վիրտուալ հիշող սարքերի 760 նկարագրություն: Հիմնվելով անսարքությունների պարբերական աղյուսակի վրա՝ մոդելավորվեցին, ներարկվեցին և հաջողությամբ նույնականացվեցին հիշողության ներքին անսարքությունները:

3.5 ենթազվխում բերվում թեստավորման և ախտորոշման հոսքի իրականացման լավացումները: Թեստավորման և ախտորոշման հոսքի իրականացման մեջ փոփոխություններ են առաջարկվել թեստային ալգորիթմի շաբլոնի օգտագործման միջոցով՝ թեստավորման և ախտորոշման փուլերի, ինչպես նաև անսարքությունների մասնակի դասակարգման համար: Այս փոփոխությունների նպատակն է թեթևացնել թեստավորման և ախտորոշման համար նոր թեստային նմուշների ստեղծումը՝ բազմաչափ կանխատեսման մեխանիզմը օգտագործելով: Նոր թեստային նմուշները հնարավոր է ստուգել առաջարկված միջավայրում:

ԵԶՐԱԿԱՑՈՒԹՅՈՒՆ

Ատենախոսությունում ստացվել են հետևյալ արդյունքները՝

1. Անսարքությունների պրմիտիվների հիման վրա միաբջիջ և երկբջջանի անսարքությունների համար առաջարկվել են մոդելներ՝ համապատասխան FDT աղյուսակների գեներացիայի ընթացակարգերի հետ մեկտեղ [1]:
2. Կապակցված անսարքությունների համար արտաձվել են անսարքության մոդելը և համապատասխան FDT գեներացիայի ընթացակարգը [2]:
3. Մշակվել է պարամետրացված FDT գեներացիայի հոսք, որն ընդգրկում կառուցված ընթացակարգերը [4]:
4. Առաջարկվել է միաբջիջ, երկբջջանի և կապակցված անսարքությունների համար մոդելների ներարկման հոսք RTL HDL միջավայրի մեջ [3]:
5. Հիմնավորվել և կառուցվել է իրականացման ստուգման միջավայրը: Կատարվել են միջավայրի փորձարկումներ, որոնք ցույց են տալիս միջավայրի կիրառելիությունը հիշողությունների բազմազան կոնֆիգուրացիաների ու տարբեր անսարքությունների համար: Առաջարկվել են նաև հետսիլիկոնային վերլուծության ավտոմատացման գործիքների թեստավորման և ախտորոշման հոսքի իրականացման որոշ լավացումներ [4]:

ԱՏԵՆԱԽՈՍՈՒԹՅԱՆ ԹԵՄԱՅՈՎ ՀՐԱՏԱՐԱԿՎԱԾ ԱՇԽԱՏՈՒԹՅՈՒՆՆԵՐ

- [1] Hayrapetyan D., Manukyan A. “Modeling dynamic single-cell and coupling faults via automata models”, Computer Science and Information Technologies (CSIT), Armenia, 2017, pp. 65-68
- [2] Hayrapetyan D., “Modeling linked faults via automata models”, IEEE East-West Design and Test Symposium (EWDTS), Serbia, 2017, pp. 237-241
- [3] Hayrapetyan D., Manukyan A., Tshagharyan G. “Implementation of Memory Static, Coupling and Dynamic Fault Models at the Register Transfer Level”, IEEE East-West Design and Test Symposium (EWDTS), Russia, 2018, pp. 744-748
- [4] Hayrapetyan D. “Verification of Test and Diagnosis Flow Implementation in Software Post-Silicon Analysis Automation Tools”, Reports of the National Academy of Sciences and the State Engineering University of Armenia, Vol. 2, Series of Technical Sciences, Yerevan, 2019, pp. 239-249

RESUME

DAVIT LEVON HAYRAPETYAN

JUSTIFICATION AND DEVELOPMENT OF A SOFTWARE TOOL MODELING FAULT-INCLUSIVE SILICON BUILT-IN TEST FLOW FOR EMBEDDED MEMORY COMPONENTS

Introduction

With technology shrinking the necessity arises of developing new test and diagnosis solutions for silicon embedded memory cores, based on their increased number present in a system-on-chip (SoC) and new types of memory defects and related faults being observed.

The approach of task distribution between hardware (HW) memory built-in self-test (MBIST) network and software (SW) automation tools, where the management and control of test and diagnosis flow are implemented via SW, while actual at-speed basic test and diagnostic procedures are performed by the components of MBIST network is broadly used in the semiconductor industry. The interactions between SW and HW parts are made via test patterns, with corresponding input/output chains, generated at SW part, which increase in complexity with development of new test algorithms, continuously growing MBIST network hierarchy and new requirements in test and diagnosis flow considering new types of memory faults. Although patterns need to be thoroughly verified before applying to the MBIST networks for excluding essential time and quality losses, to the best of our knowledge there are few publications on methods of verification used for MBIST networks post-silicon analysis automation tools. The studies are mainly focused on verification of test execution on MBIST engine rather than complete networks.

Since it is crucial to ensure the correctness of test and diagnosis flow implementation before it is applied to a manufactured SoC, a justified fault-inclusive and adequate to MBIST network implementation in SoC environment for patterns verification is required for modeling test pattern execution on the MBIST network. Accurate models of memory faults corresponding to defects that might be present in the memory cores should be described and used within the environment. The considered environment should cover necessarily the following 2 requirements:

- fault model generation and injection,
- verification of test pattern generation and output chain analysis.

The aim of the thesis

The thesis aims to justify and develop a verification environment for memory test and diagnosis flow basing on extension of memory built-in self-test network RTL representation with fault models. The environment will allow to analyze the received output chains of test patterns applied to the fault-inclusive RTL representation via generation, injection and modelling the memory faults.

Scientific novelty

- Extendable fault model for memory internal faults and its implementation in the RTL HDL simulation environment.
- Automated generation flow for memory internal fault models based on fault periodicity table (FPT).
- Method of test pattern verification in fault-inclusive environment via resulting chain analysis and traversed fault model graph comparison.
- Approach for verification of test and diagnosis flow implementation.

Practical value and implementation

The proposed approach allows to increase reliability of test and diagnosis flow implementation in post-silicon analysis test automation tools. The suggested fault modeling mechanism can be effectively extended for new types of faults.

The results of the thesis were implemented in Synopsys Inc. software post-silicon analysis automation tool which is currently used by multiple customers.

The following topics are presented for defence

- Extendable fault model for memory internal faults at the Register Transfer Level [1] [2].
- Approach for fault model injection in the RTL simulation environment [3].
- A generation flow for memory internal fault models via fault periodicity table [4].
- Method of generated test patterns verification [4].
- Approach for verification of test and diagnosis flow implementation [4].

ОБОСНОВАНИЕ И РАЗРАБОТКА ПРОГРАММНЫХ ИНСТРУМЕНТАЛЬНЫХ СРЕДСТВ МОДЕЛИРУЮЩИХ ПРОЦЕССЫ ВСТРОЕННОГО ТЕСТИРОВАНИЯ БЛОКОВ ПАМЯТИ НА КРИСТАЛЛЕ В УСЛОВИЯХ ПРИВНЕСЕНИЯ ОШИБОК

Введение

С уменьшением размеров технологических устройств, возрастает необходимость в разработке новых решений для тестирования и диагностирования компонентов памяти встроенных в кристаллы, в связи с их увеличенным количеством в системах на кристалле (СнК SoC), наблюдаемыми новыми видами неисправностей устройств памяти и связанными с ними ошибками.

В полупроводниковой промышленности широко используется подход распределения задач между аппаратной (HW) встроенной самотестирующейся сетью компонентов памяти (MBIST) и программными автоматизированными инструментами. Управление и контроль информационного потока тестирования и диагностики реализованы программно, в то время как базовые процедуры тестирования и диагностики выполняются компонентами MBIST сети на максимальной частоте (*at-speed*). Взаимодействие между программной и аппаратной частями данного подхода производится с помощью тестовых наборов (*test patterns*) с соответствующими последовательностями (*chains*) входных/выходных данных которые создаются в программной части и становятся более сложными по мере разработки новых алгоритмов тестирования, растущей иерархии MBIST сетей и новых требований к потоку тестирования и диагностики предусматривающих новые виды ошибок устройств памяти. Несмотря на то что тестовые наборы должны тщательно проверяться перед выполнением на MBIST сети во избежание потери ценного времени и качества продукта, в меру наших знаний очень мало публикаций посвящено методам проверок используемых в инструментах автоматизированного постсиликонового (*post-silicon*) анализа MBIST сетей. В основном исследования сосредоточены вокруг методов проверки выполнения тестирования отдельными MBIST компонентами, а не MBIST сетью в целом.

Чтобы удостовериться в правильности реализации потока тестирования и диагностики перед тем как он будет применен к изготовленной СнК, необходима, отвечающая требованиям реализации MBIST сети в СнК, компетентная среда с возможностью привнесения ошибок для моделирования исполнения тестовых наборов в MBIST сетях. Точные модели ошибок соответствующих неисправностям встречающихся в блоках памяти должны быть описаны и использованы в упомянутой среде. Рассматриваемая среда обязана соответствовать двум требованиям:

- возможность создания и внесения ошибок,

- проверка создания тестовых наборов и анализа последовательностей выходных данных.

Цели работы

Целью работы является обоснование и разработка проверочной среды для тестирования и диагностирования памяти. Среда основана на расширении RTL (уровень регистровых передач) представления встроенной самотестирующейся сети для блоков памяти с возможностью привнесения ошибок и должна допускать анализ выходных последовательностей тестовых наборов применяемых к RTL представлению, пользуясь созданными, привнесенными и смоделированными ошибками памяти.

Научная новизна

- Расширяемая модель внутренних ошибок памяти и ее реализация в моделирующей RTL HDL среде.
- Автоматизированный поток создания моделей внутренних ошибок памяти основанный на периодической таблице ошибок (FPT).
- Метод проверки тестовых наборов в среде с привнесенными ошибками с помощью анализа последовательностей выходных данных и сравнения пройденных (traversed) графов модели ошибок.
- Подход к проверке реализации потока тестирования и диагностирования.

Практическая ценность и реализация

Предложенный подход дает возможность улучшить надежность реализации потока тестирования и диагностирования в автоматизированных инструментах постсиликонового анализа. Представленный механизм моделирования ошибок может быть эффективно расширен учитывая новые виды ошибок.

Результаты работы были реализованы в автоматизированном программном инструменте постсиликонового анализа Synopsys Inc., которым на данный момент пользуются многочисленные клиенты.

На защиту представлены следующие положения

- Расширяемая модель внутренних ошибок памяти для RTL [1] [2].
- Подход к привнесению модели ошибок в моделирующую RTL среду [3].
- Поток создания моделей внутренних ошибок памяти учитывая периодическую таблицу ошибок [4].
- Метод проверки созданных тестовых образцов [4].
- Подход к проверке реализации потока тестирования и диагностики [4].