ISSN 0002-306X. ՀԳԱԱ և ՀԱՊՀ Տեղ. Տեխն. գիտ. սերիա. 2017. Հ. LXX, N 1.

ՀՏԴ 681.3.07

ՀԱՇՎՈՂԱԿԱՆ ՏԵԽՆԻԿԱ ԵՎ ԻՆՖՈՐՄԱՏԻԿԱ

Գ.Է. ՀԱՐՈՒԹՅՈՒՆՅԱՆ

ՀԱՄԱՊԻՏԱՆԻ ՆԵՐԿԱՌՈՒՑՎԱԾ ԹԵՍՏԱՎՈՐՄԱՆ ՀԱՄԱԿԱՐԳ՝ ՀԻՄՆՎԱԾ ԱՆՍԱՐՔՈՒԹՅՈՒՆՆԵՐԻ ՊԱՐԲԵՐԱԿԱՆ ԱՂՅՈՒՍԱԿԻ ԵՎ ԹԵՍՏԱՅԻՆ ԱԼԳՈՐԻԹՄՆԵՐԻ ՇԱԲԼՈՆԻ ՎՐԱ

Առաջարկվում է համապիտանի ներկառուցված թեստավորման համակարգ (ՆԹՀ), որն ապահովում է թեստային ալգորիթմների ծրագրավորման առավելագույն մակարդակ: Առաջարկված ՆԹՀ-ն հնարավորություն է տալիս ընդլայնել հայտնաբերվող անսարքությունների բազմությունը՝ առանց փոխելու ՆԹՀ-ի ճարտարապետությունը:

Առանցքային բառեր. անսարքություն, թեստային ալգորիթմ, հիշող սարք, ներկառուցված թեստավորման համակարգ, թեստային ալգորիթմի ռեգիստր, անսարքությունների պարբերական աղյուսակ:

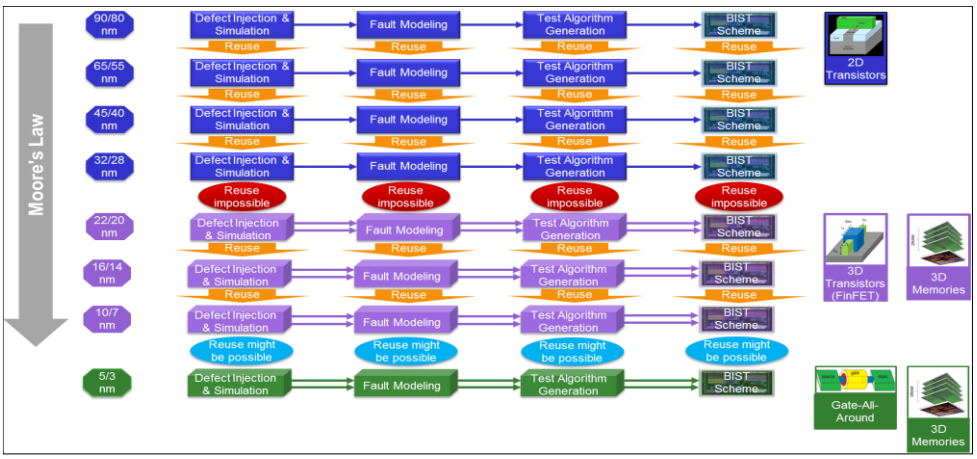
Ներածություն. Ներկառուցված թեստավորման համակարգերը (ՆԹՀ) լայնորեն կիրառվում են հիշող սարքերի թեստավորման համար: Արտադրության փուլում ՆԹՀ-ն հիշող սարքի վրա կիրառում է գրել-կարդալու հաջորդականություններ, որպեսզի ստուգվի հիշող սարքի աշխատունակությունը: Գրել-կարդալու նշված հաջորդականություններն անվանվում են թեստային ալգորիթմներ [1]: Գոյություն ունեն տարբեր տիպի ՆԹՀ-երի ճարտարապետություններ, որոնցից ամենատարածվածներից է միկրոկոդի վրա հիմնված ծրագրավորվող ՆԹՀ-ն [2]: Այն ունի թեստային ալգորիթմի ռեգիստր (ԹԱՌ), որի մեջ նախապես հայտնի ֆորմատով պահվում է թեստային ալգորիթմը: ԹԱՌ-ը հասանելի է արտաքին միջավայրից, որը թույլ է տալիս դրսից ՆԹՀ-ում ծրագրավորել թեստային նոր ալգորիթմներ: Սակայն այս տիպի ՆԹՀ-երում թեստային ալգորիթմի բաղադրիչները (հասցեավորման կարգերը, գրել/կարդալու գործողությունները և հիշող սարքում գրվող/կարդացվող տվյալները) հիմնականում լինում են նախապես հայտարարված, ինչը կարող է հանգեցնել տրված թեստային ալգորիթմի կառուցման անհնարինությանը: Հետևաբար, կարիք է լինում թեստային ալգորիթմի բաղադրիչները նույնպես դարձնել ծրագրավորվող՝ապահովելով թեստային ալգորիթմի ծրագրավորման առավելագույն ճկունություն [3]:

Սակայն գոյություն ունեցող ՆԹՀ-երը չեն երաշխավորում, որ հնարավոր լինի նույն ՆԹՀ-ն առանց փոփոխելու կիրառել նաև ապագա տեխնոլոգիաների հիշող սարքերում: Ուստի կարևոր է մշակել համապիտանի ՆԹՀ, որը թույլ կտա

ծրագրավորել առավելագույն ճկունությամբ թեստային ալգորիթմներ, ինչպես նաև թեստավորել ոչ միայն ընթացիկ, այլ նաև ապագա տեխնոլոգիաների հիշող սարքերի անսարքությունները՝ առանց փոխելու ՆԹՀ-ի ճարտարապետությունը: Նման ՆԹՀ իրականացնելու համար նախ պետք է հնարավոր դարձնել ապագա տեխնոլոգիաների հնարավոր անսարքությունների կանխատեսումը: Այս աշխատանքում այդ խնդիրը լուծված է անսարքությունների պարբերական աղյուսակի միջոցով, որը թույլ է տալիս կանխատեսել ապագա տեխնոլոգիաների անսարքությունները և թեստային ալգորիթմների շաբլոնի միջոցով կառուցել արդյունավետ թեստային ալգորիթմներ՝ այդ անսարքությունները թեստավորելու համար:

1. Նանոչափական հիշող սարքերի զարգացումը. Նանոչափական հիշող սարքերի չափերի աննախադեպ փոքրացումը հանգեցնում է նախագծման էական բարդությունների, և դրան զուգընթաց՝ էական առաջընթաց է նկատվում ներդրված թեստավորման լուծումներին ներկայացվող պահանջներում՝ նանոչափական բյուրեղների (չիպերի) բարձր արտադրողականություն և որակ ապահովելու համար:

Նկ. 1-ում բերված է նանոչափական հիշող սարքերի զարգացման շղթան: Ինչպես երևում է նկարից՝ 90-ից մինչև 28 նանոմետրանոց հիշող սարքերում օգտագործվում էին երկչափ (2D) տրանզիստորներ, և մեկ տեխնոլոգիայից մյուսին անցնելիս գոյություն ունեցող թեստավորման լուծումները հնարավոր էր լինում վերաօգտագործել: Հիմնական պատճառն այն էր, որ անցման ժամանակ փոփոխվում էին միայն հիշող սարքի չափերը, և կառուցվածքային որևէ փոփոխություն տեղի չէր ունենում:

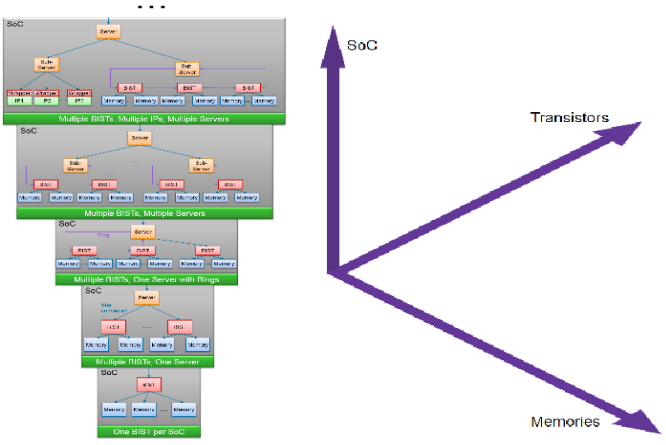


Նկ. 1. Նանոչափական տեխնոլոգիաների զարգացումը

Հետագա հետազոտությունները ցույց տվեցին, որ 22 նանոմետրից սկսած՝ հիշող սարքերի արագագործության բարելավումը հնարավոր չէ իրականացնել գոյություն ունեցող մեխանիզմներով, և հիշող սարքերում տեղի ունեցան կառուցվածքային նոր փոփոխություններ. երկչափ (2D) տրանզիստորից անցում կատարվեց եռաչափ (3D) տրանզիստորի կառուցվածքին [4], ինչպես նաև երկչափ (2D) հիշող սարքերից անցում կատարվեց եռաչափ (3D) հիշող սարքերին [5]:

Նկ. 1-ում պատկերված հիշող սարքերի զարգացման շղթան բավարարում է «Ինտել» ընկերության համահիմնադիր Գ. Մուրի կողմից բերված օրենքին (կոչվում է «Մուրի օրենք»): 1965թ. Մուրը կանխատեսել է, որ հիշող սարքի միավոր մակերեսում տրանզիստորներն ամեն 18 ամիսը մեկ կկրկնապատկվեն [6]: Հատկանշական է, որ այս օրենքը գործում է մինչ օրս:

Բացի տրանզիստորների և հիշող սարքերի կառուցվածքային փոփոխություններից, փոփոխվում են նաև բյուրեղները. դրանք գնալով մեծանում են՝ առաջ բերելով թեստավորման նոր խնդիրներ: Նկ. 2-ում նկարագրված են համակարգ բյուրեղի վրա և դրա զարգազման շղթան: Եթե փոքր բյուրեղներում մեկ թեստավորման համակարգը բավարար էր, ապա արդի բյուրեղներում, որտեղ հանդիպում են բազում հիշող սարքեր և զանազան անալոգային և խառը ազդանշանային բլոկներ, անհրաժեշտ է լինում կառուցել մեկից ավելի թեստավորման ենթահամակարգեր, իսկ այնուհետև, օգտագործելով այդ ենթահամակարգերը, կառուցել համակարգ բյուրեղի վրա՝ պահպանելով բյուրեղի հիերարխիայի տրամաբանությունը:



Նկ. 2. Համակարգ բյուրեղի վրա և դրա զարգացումը

2. Անսարքությունների պարբերական աղյուսակ. [7-10] աշխատանքներում առաջարկվել է հիշող սարքերի անսարքությունների ներկայացման նոր ձև՝ ան-

սարքությունների պարբերական աղյուսակի տեսքով (տե՛ս աղ.): Աղյուսակում $FG_i(x, S)$ նշանակում է.

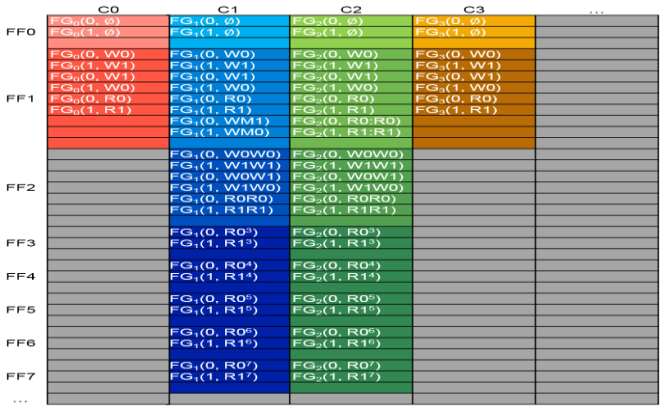
- i - անսարքություններին մասնակցի բջիջների քանակը,
- x - բջջի արժեքը՝ նախքան դրանում անսարքության ակտիվացումը,
- S - անսարքությունն ակտիվացնող գործողությունների հաջորդականությունը:

Անսարքությունների պարբերական աղյուսակն ունի հետևյալ առավելությունները.

- Թույլ է տալիս հեշտությամբ վերլուծել և հիշել մեծ թվով անսարքություններ մեկ աղյուսակում:
- Հնարավորություն է տալիս անսարքությունները ներկայացնել համակարգված ձևով, և բացառում է որևէ անսարքության բացթողումը դիտարկումից:
- Չկա որևէ սահմանափակում տողերի և սյուների քանակների վրա, և դրանք կարող են ընդլայվել՝ ապագա տեխնոլոգիաներն անսարքություններն ընդգրկելու համար:
- Թույլ է տալիս կանխատեսել նոր անսարքություններ՝ հիմնվելով անսարքությունների պարբերական հատկությունների վրա:
- Հնարավորություն է տալիս գնահատել տրված թեստային ալգորիթմով հայտնաբերվող անսարքությունների ծածկույթը:
- Թույլ է տալիս կառուցել համապիտանի ՆԹՀ, որն ապահովում է թեստային ալգորիթմի ծրագրավորելիությունը՝ առանց սահմանափակումների:

Աղյուսակ

Անսարքությունների պարբերականություն



3. Թեստային ալգորիթմների շաբլոն. Գոյություն ունեն թեստային ալգորիթմների կառուցման տարբեր մեթոդներ [11, 12]: Դրանց որոշ մասը հիմնված է փնտրման մեթոդների վրա. առաջարկվում է կատարել թեստային ալգորիթմների հատարկում՝ սկսած ամենակարճ երկարությամբ թեստային ալգորիթմից, այնուհետև մեծացնելով դրա երկարությունը: Այդ մեթոդի առավելությունն այն է, որ հայտնաբերված ալգորիթմը նվազագույնն է, իսկ թերությունն այն է, որ այս տիպի մեթոդները հաճախ պահանջում են երկար ժամանակ (օրեր, ամիսներ)՝ որոնվող թեստային ալգորիթմը գտնելու համար:

Մեկ այլ տարածված մոտեցում է թեստային ալգորիմների կառուցման էվրիստիկական մեթոդը, երբ թեստային ալգորիթմների փնտրումը կատարվում է ոչ լրիվ հատարկման հիման վրա, այլ հաշվի առնելով հիշող սարքերում հանդիպող անսարքություններից բխող որոշակի պայմաններ: Այս տիպի մեթոդների առավելությունն այն է, որ փնտրվող թեստային ալգորիթմը կարող է գտնվել կարճ ժամանակում (միլիվայրկյանների, վայրկյանների ընթացքում), իսկ թերությունն այն է, որ գտնված թեստային ալգորիթմը կարող է լինել ոչ արդյունավետ, այսինքն՝ երկարությամբ շատ տարբեր լինել նվազագույն թեստային ալգորիթմի երկարությունից:

Թեստային ալգորիթմների շաբլոնը թույլ է տալիս արագ և պարզ ձևով, առանց ալգորիթմների կառուցման գործիք օգտագործելու, կառուցել թեստային ալգորիթմներ, ինչը հիմնականում կատարվում է փնտրման կամ էվրիստիկական մեթոդներով:

Ներկայացնենք թեստային ալգորիթմների MTT(x, S) շաբլոնը.

$$\Uparrow(W(\sim D_k)); \Uparrow([R(\sim D_k)], [W(x)], S); \Uparrow([R(D_k)], [W(\sim x)], \sim S);$$

$$\Downarrow([R(\sim D_k)], [W(x)], S); \Downarrow([R(D_k)], [W(\sim x)], \sim S); \Downarrow(R(\sim D_k)),$$

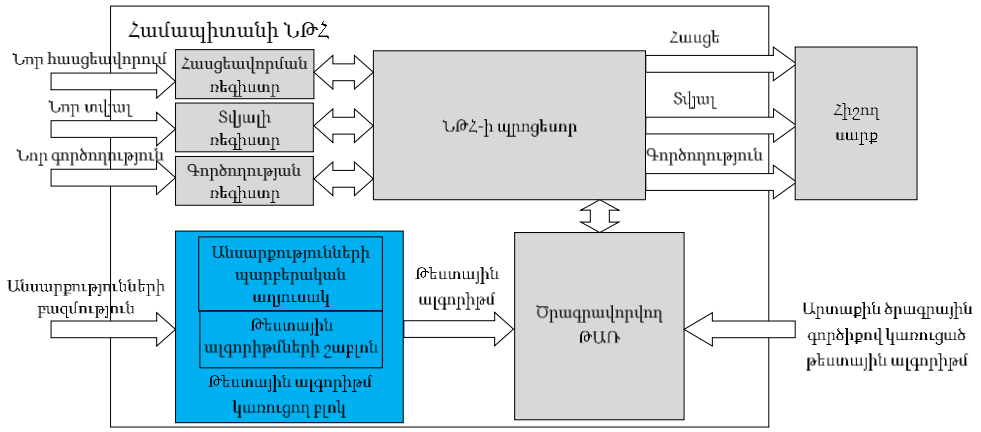
որտեղ՝

- $S=OP1D1, \ldots, OPkDk$, $k\geq 0$, $x\in\{0, 1\}$,
- եթե $k\geq 1$, ապա $\sim S=OP_1(\sim D_1), \ldots, OP_k(\sim D_k)$, և եթե $S=\varnothing$, ապա $\sim S=\varnothing$,
- $[W(x)]$ և $[W(\sim x)]$ բացակայում են MTT(x, S)-ի կառուցվածքում, եթե $S \neq \varnothing$ և $x=\sim D_k$, հակառակ դեպքում՝ դրանք առկա են,
- $[R(D_k)]$ և $[R(\sim D_k)]$ բացակայում են MTT(x, S)-ի կառուցվածքում, եթե $S \neq \varnothing$, $x=\sim D_k$, $OP_1=R$, հակառակ դեպքում՝ դրանք առկա են,
- եթե $S=\varnothing$, ապա $D_k=x$:

4. Համապիտանի ներկառուցված թեստավորման համակարգ. Նկ. 3-ում ցույց է տրված առաջարկված համապիտանի ՆԹՀ-ի ճարտարապետությունը, որը բաղկացած է հետևյալ բլոկներից.

- *Ծրագրավորվող ԹԱՌ.* թեստային ալգորիթմի պահպանման համար նախատեսված ռեգիստր: Ռեգիստրում հնարավոր է ծրագրավորել թեստային նոր ալգորիթմ, որը հաճախ կառուցվում է արտաքին ծրագրային գործիքի միջոցով:
- *Հասցեավորման, տվյալի* և *գործողության ռեգիստրներ.* նախատեսված են համապատասխանաբար *նոր հասցեավորում, նոր տվյալ* և *նոր գործողություն* սահմանելու համար, որոնք հետագայում կարող են օգտագործվել թեստային ալգորիթմներ ծրագրավորելիս:
- *ՆԹՀ-ի պրոցեսոր. ծրագրավորվող ԹԱՌ-ում* գոյություն ունեցող թեստային ալգորիթմի հիման վրա գեներացնում են *հասցե*, *տվյալ*, *գործողություն* եռյակները, որոնք կիրառվում են *հիշող սարքի* վրա:
- *Թեստային ալգորիթմ կառուցող բլոկ.* պարունակում է *անսարքությունների պարբերական աղյուսակը* և *թեստային ալգորիթմների շաբլոնը*: ՆԹՀ-ն օգտագործողը հնարավորություն ունի սահմանելու անսարքությունների բազմություն, որի հիման վրա կառուցվում է *թեստային ալգորիթմ* անմիջապես ՆԹՀ-ում, որն իր հերթին ծրագրավորվում է ԹԱՌ-ում:

Ի համեմատություն գոյություն ունեցող ՆԹՀ-երի՝ համապիտանի ՆԹՀ-ի բոլոր բլոկներում իրականացվել են լավացումներ, իսկ *թեստային ալգորիթմ կառուցող բլոկը* հատուկ է միայն այս ճարտարապետությանը: Թեստային ալգորիթմների շաբլոնի առանձնահատկությունները թույլ են տալիս թեստային ալգորիթմների կառուցումը ծրագրային գործիքի մակարդակից տանել դեպի ապարատային իրականացումը, ինչը առաջարկված համապիտանի ՆԹՀ-ի ճարտարապետության հիմնաքարերից մեկն է:



Նկ. 3. Համապիտանի ՆԹՀ-ի ճարտարապետությունը

Առաջարկված համապիտանի ՆԹՀ-ի հիմնական առավելություններն են.

- Թեստային ալգորիթմների ծրագրավորման առավելագույն ճկունությունը՝ օգտագործելով ինչպես թեստային ալգորիթմի ծրագրավորելիության, այնպես էլ թեստային ալգորիթմի յուրաքանչյուր բաղադրիչի (հասցեավորում, տվյալ, գործողություն) ծրագրավորելիության հնարավորությունը:
- ՆԹՀ-ի ընդլայնման հնարավորությունը՝ ապագա տեխնոլոգիաների հիշող սարքերի անսարքությունները թեստավորելու համար, առանց փոխելու ՆԹՀ-ի ճարտարապետությունը:
- Օգտագործողի հնարավորությունը՝ սահմանելու միայն անսարքությունների բազմությունը ՆԹՀ-ն օգտագործելու համար՝ առանց թեստային ալգորիթմ կառուցող ծրագրային գործիքի:

Նշենք, որ առաջարկված ՆԹՀ-ի միջոցով ստացվել են հետևյալ արդյունքները.

1. Եռաչափ տրանզիստորներով կառուցված հիշող սարքերի թեստավորում: Փորձարկումների արդյունքում պարզվել է, որ եռաչափ տրանզիստորներին հատուկ անսարքությունների որոշ մասն արդեն գոյություն ունի պարբերական աղյուսակում, իսկ մյուսները՝ ոչ: Արդյունքում՝ պարբերական աղյուսակի որոշ դատարկ վանդակներ լրացվել են հայտնաբերված նոր անսարքություններով [10]: Այնուհետև թեստային ալգորիթմների շաբլոնով կառուցվել է March FF թեստային ալգորիթմը, որն ունի 24N բարդություն, որտեղ N-ը հիշող սարքի հասցեների քանակն է: Հիմնավորվել է, որ March FF թեստային ալգորիթմն ունի նվազագույն բարդություն:

2. Եռաչափ հիշող սարքերի թեստավորում: Այս դեպքում պարզվել է, որ անհրաժեշտ է ընդլայնել անսարքությունների պարբերական աղյուսակը, որպեսզի հետազոտությունների արդյունքում հայտնաբերված եռաչափ հիշող սարքերին հատուկ անսարքությունները տեղավորվեն այդ աղյուսակի ընդլայնված մասում [10]:

3. Ներկառուցված թեստավորման հիերարխիական համակարգ: Նոր բյուրեղները կարող են պարունակել մեկից ավելի ՆԹՀ-եր: Գոյություն ունեն զանազան տիպի գործոններ, որից կախված՝ նույն բյուրեղի շրջանակներում կարիք է լինում տարբեր ՆԹՀ-երի համար կառուցել տարբեր թեստային ալգորիթմներ: Առաջարկված ՆԹՀ-ի առավելությունների հաշվին հնարավոր եղավ կառուցել տրված գործոնները բավարարող թեստավորման հիերարխիկական համակարգ:

Եզրակացություն. Առաջարկվել է համապիտանի ներկառուցված թեստավորման համակարգ, որն ապահովում է թեստային ալգորիթմների ծրագրավորելիության առավելագույն մակարդակ: Այն հիմնված է անսարքությունների պարբերական աղյուսակի և թեստային ալգորիթմների շաբլոնի վրա: Առաջարկված

ՆԹՀ-ն թույլ է տալիս հայտնաբերել նոր տիպի անսարքություններ՝ առանց փոխելու ՆԹՀ-ի ճարտարապետությունը: Դա նշանակում է, որ գոյություն ունեցող ՆԹՀ-ն հնարավոր կլինի օգտագործել ապագա տեխնոլոգիայի հիշող սարքերի թեստավորման համար:

ԳՐԱԿԱՆՈՒԹՅԱՆ ՑԱՆԿ

1. **Van de Goor A.J.** Testing semiconductor memories: Theory and Practice. - John Wiley & Sons, 1991.
2. **Youn D., Kim T., Park S.** A microcode-based memory BIST implementing modified march algorithm // IEEE Asian Test Symposium. - 2001. - P. 391-395.
3. **Hakhumyan A., Harutyunyan G.** Implementation of a Flexible BIST Architecture Based on Programmability of Test Operations, Patterns and Algorithms // Computer Science and Information Technologies (CSIT). - 2011. - P. 287-290.
4. **Cheng H.-W., Li Y.** 16-nm Multigate and Multifin MOSFET Device and SRAM Circuits // International Symposium on Next-Generation Electronics. - 2010. - P. 32-35.
5. **Patti R.S.** Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs // Proceedings of the IEEE. - 2006. - Vol. 94, No. 6. - P. 1214–1224.
6. **Moore G.E.** Cramming More Components Onto Integrated Circuits // Electronics. - 1965. - Vol. 38, No. 8. - P. 114-117.
7. **Harutyunyan G., Shoukourian S., Zorian Y.** Fault and Test Algorithm Periodicity Hypothesis in Memory Devices and Its Application to Memory BIST Processor Architecture // Reports of National Academy of Sciences of Armenia. - 2012. - Vol. 112, No. 3. - P. 229-238.
8. **Harutyunyan G., Shoukourian S., Vardanian V., Zorian Y.** An Effective Solution for Building Memory BIST Infrastructure Based on Fault Periodicity // IEEE VLSI Test Symposium. - 2013. - P. 71-76.
9. **Harutyunyan G., Shoukourian S., Vardanian V., Zorian Y.** Extending Fault Periodicity Table for Testing Faults in Memories under 20nm // IEEE East-West Design and Test Symposium. - 2014. - P. 12-15.
10. **Harutyunyan G.** An efficient test approach for modern nanoscale memory devices // Reports of National Academy of Sciences of Armenia. - 2017. - Vol. 117, No. 1.
11. **Benso A., Bosio A., Di Carlo S., Di Natale G., Prinetto P.** Automatic March Tests Generation for Static and Dynamic Faults in SRAMs // IEEE European Test Symposium. - 2005. - P. 122–127.
12. **Wu C.-F., Huang C.-T., Cheng K.-L., Wu C.-W.** Fault Simulation and Test Algorithm Generation for Random Access Memories // IEEE TCAD on Integration Circuits and System. - 2002. - Vol. 21, No. 4. - P. 480–490.

Երևանի պետական համալսարան: Նյութը ներկայացվել է խմբագրություն 30.06.2016:

Г.Э. АРУТЮНЯН

УНИВЕРСАЛЬНАЯ ВСТРОЕННАЯ ТЕСТОВАЯ СИСТЕМА НА ОСНОВЕ ПЕРИОДИЧЕСКОЙ ТАБЛИЦЫ НЕИСПРАВНОСТЕЙ И ШАБЛОНА ТЕСТОВОГО АЛГОРИТМА

Предлагается универсальная встроенная тестовая система (ВТС), которая обеспечивает максимальный уровень программируемости тестовых алгоритмов. Предлагаемая ВТС дает возможность расширить множество обнаруживаемых неисправностей без изменения архитектуры ВТС.

Ключевые слова: неисправность, тестовый алгоритм, запоминающее устройство, встроенная тестовая система, регистр тестового алгоритма, периодическая таблица неисправностей.

G.E. HARUTYUNYAN

A UNIVERSAL BUILT-IN TEST SYSTEM BASED ON A FAULT PERIODICITY TABLE AND A TEST ALGORITHM TEMPLATE

A universal built-in self-test (BIST) system is proposed which provides a maximum level of test algorithm programmability is proposed. The proposed BIST system allows to extend the set of detected faults without changing the BIST architecture.

Keywords: fault, test algorithm, memory device, built-in self-test system, test algorithm register, fault periodicity table.