

ՀՏԴ 621.52

ԱՎՏՈՄԱՏԱՑՈՒՄ ԵՎ ԿԱՌԱՎԱՐՄԱՆ
ՀԱՄԱԿԱՐԳԵՐ

Կ.Ս. ԱՄԻՐՔԱՆՅԱՆ

ՀԻՇՈՂՈՒԹՅԱՆ ՆՍՈՒՇՆԵՐԻ ԿԱՌՈՒՑՎԱԾՔԱՑԻՆ ՍՈՂԵԼԻ ԱՎՏՈՄԱՏ
ԴՈՒՐՄԲԵՐՈՒՄԸ ԳՐԱՖԻԿԱԿԱՆ GDSII ՁԵՎԱԶԱՓԻ ՖԱՅԼԻՑ

Տրված է ծրագրային հոսքի (ՕՀ) նկարագրությունը (program flow), որը հնարավորություն է տալիս ավտոմատ կերպով ստանալ ստատիկ պատահական դիմումով հիշողության (ՍՊԴՀ) նմուշի կառուցվածքային մոդելը այդ հիշողության նմուշի արտադրության մեջ օգտագործվող **GDSII** (Graphic Database System) ձևաչափով ներկայացված ֆայլից: Ներկայացված են ծրագրային միջավայրի ֆունկցիոնալ, կառուցվածքային սխեմաները, մոդելի դուրսբերման առնչվող հիմնախնդիրները, բարդությունները և սահմանափակումները: Բերված են փորձերի արդյունքները:

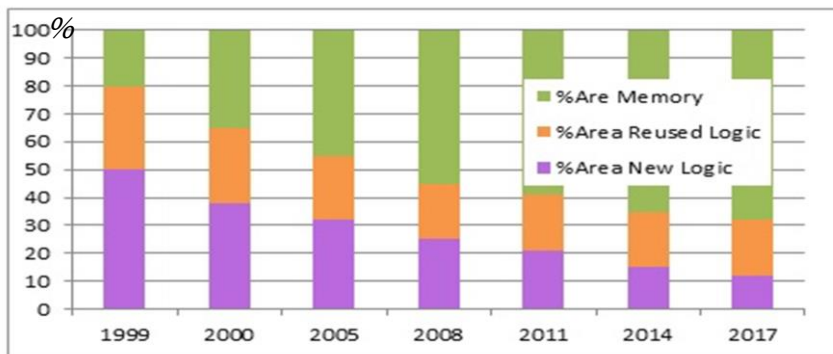
Առանցքային բառեր. կառուցվածքային մոդել, հիշողության նմուշ, նմուշի **GDSII** ձևաչափ, ծրագրային միջավայր, ՍՊԴՀ, կառուցվածքային տիպեր:

Ներածություն. Ժամանակակից կիսահաղորդչային սարքերի բնորոշ հատկություններից մեկն այն է, որ դրանք ընդգրկում են մեծ քանակությամբ ներդրված հիշողության սարքեր (ՀՄ)՝ ՍՊԴՀ-ներ: Հետագա շարադրանքում, ասելով ՀՄ, նկատի կունենանք միայն ՍՊԴՀ տիպի հիշողությունները: Այդ ՀՄ-երի քանակը կարող է հասնել հազարների, որոնք բազմաթիվ գործառնություններ են կատարում սարքավորումներում: Ներդրված հիշողության բջիջների (ՆՀԲ) գումարային քանակը շատ դեպքերում կազմում է մի քանի միլիոն: Ընդ որում, հիշողության բջիջների քանակը սարքերում տարեցտարի անընդհատ աճում է և ըստ վիճակագրական տվյալների [1]՝ 2015 թ. ՆՀԲ-երը կգրադենեն էլեկտրոնային սարքերի ֆիզիկական մակերեսի մոտ 70-80%-ը (նկ. 1): Որպեսզի առավելագույնս նվազեցվի հիշողության բջիջներով զբաղված սարքերի թիթեղի մակերեսը, այդ բջիջների կառուցվածքը՝ տոպալոգիան (layout), նախագծվում է առավելագույնս խիտ:

Այսպիսով, հիշողության բջիջների երկու առանձնահատկությունները՝ կառուցվածքի խտությունը և զբաղեցրած մակերեսը, նախագծվող սարքերի հիշողություն պարունակող տարածքում արտադրության ընթացքում առաջացած անսարքությունների (արատների) բարձր հավանականության պատճառներն են:

Հաշվի առնելով այն փաստը, որ նույնիսկ կատարյալ պայմաններում անհնար է ստանալ արտադրվող միկրոսխեմաների 100% արտադրության օգտակար ելք, դժվար չէ կռահել, որ արտադրության խտանի հիմնական պատճառը հիշողության սարքերի մասում առաջացած արտադրական և շահագործման անսարքություններն են: Հիշողության սարքերի արտադրության օգտակար ելքի ավելացման նպատակով վերջին

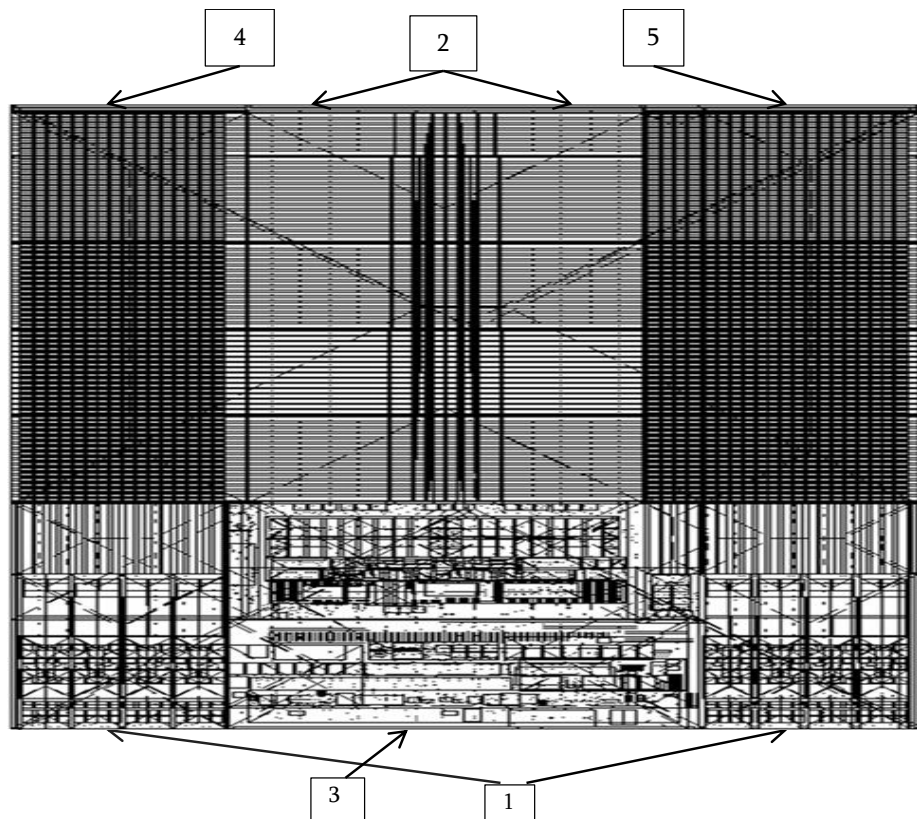
տարիներին հիշողության սարքերում տեղադրվում են լրացուցիչ այսպես կոչված, ավելցուկային հիշողության բջիջներ պարունակող տողեր և սյուներ: Հետագայում դրանք օգտագործվում են անսարք բջիջների աշխատանքը (տողերը կամ սյուները) վերականգնելու նպատակով (Built-in Repair Algorithm-BIRA):



Նկ. 1. Հիշողության մասնաբաժնի քանակը էլեկտրոնային սարքերում (կանխագուշակում՝ ըստ ITRS.net կայքի)

Նույն նպատակով հիշողության սարքերում տեղադրվում են ներկառուցված (embedded), հիշողության հետ միաժամանակ պատրաստվող, լրացուցիչ էլեկտրոնային սխեմաներ, որոնց նպատակն է հիշողության հանգույցների ինքնաթեստավորումը, անսարքությունների հայտնաբերումը (Built-in Self Test –BIST) [1-3], և, հնարավորության դեպքում, հիշողության աշխատունակության վերականգնումը:

ՀՄ-երի աշխատանքի ընթացքում ի հայտ են գալիս մի շարք ավտոմատացման խնդիրներ, ինչպիսիք են՝ հիշողության թեստավորումը, հիշողության բջիջների կոորդինատների հաշվարկը, նախագծի մեջ օգտագործվող հիշողության սարքերի օպտիմալ տեսակների և կոնֆիգուրացիաների որոշումը, նախագծի և, մասնավորապես, հիշողության սարքի արտադրության օգտակար ելքի կանխատեսման հաշվարկը և վերլուծումը: Նշված խնդիրները պահանջում են տիրապետել և համապատասխան ծրագրային միջոցներում օգտագործել հիշողության կառուցվածքի վերաբերյալ տեղեկությունները: Հիշողության կառուցվածքային տեղեկատվությունը (ՀԿՏ) գտնվում է հիշողության արտադրության մուտքային տեղեկություն պարունակող կառուցվածքային ֆայլում, որը ստեղծվում և պահվում է համակարգիչներում գրաֆիկական GDSII ձևաչափով: Նկ. 2-ում ներկայացված է հիշողության սարքի նմուշի գրաֆիկական GDSII ձևաչափի օրինակը: GDSII տեսակի գրաֆիկական ձևաչափը ծավալուն է, և այդ ֆայլի մշակումը այլ ծրագրային միջոցներով անչափ անհարմար է ու ժամանակատար:



Նկ. 2. Հիշողության սարքի նմուշի օրինակ՝ GDSII ձևաչափ.

1. միացման Մուտքային/Ելքային բջիջներ, 2. տողերի ապակողավորման սխեմա, 3. սյուների ապակողավորման սխեմա, 4. հիշողության բջիջներ՝ խմբավորված ՀՄ-ի աջ մասում, 5. հիշողության բջիջներ՝ խմբավորված ՀՄ-ի ձախ մասում

Ծրագրային այլ միջոցներով աշխատանքի ժամանակը էապես պակասեցնելու նպատակով ստեղծվում է հիշողության նմուշի GDSII ձևաչափի պարզեցված կառուցվածքային մոդելը: Մոդելը կառուցված է տեքստային TXT ձևաչափով և, համապատասխան ստուգումից հետո [4], հետագայում օգտագործվում է այլ կիրառական ծրագրերում [5,6]:

Հիշողության մոդելը բաղկացած է երկու՝ հիշողության վարքագծային և կառուցվածքային մասերից [4]: Աշխատանքում ներկայացվող ծրագրային միջավայրը հնարավորություն է տալիս ավտոմատ ձևով ստանալ *կառուցվածքային մոդելը*: Կառուցվածքային մոդելը կազմող կառուցվածքային տիպերը և դրանց հատկությունները մանրամասնորեն ներկայացված են [4]-ում: Այստեղ պարզապես կթվարկենք այդ կառուցվածքային տիպերից հիմնականները.

1. հասցեի խճողման ծածկագրում (address scrambling),

2. բանկերի ապակողավորման ֆունկցիա (banks scrambling),
3. տողերի ապակողավորման ֆունկցիա (rows scrambling),
4. սյուների ապակողավորման ֆունկցիա (columns scrambling),
5. Մուտքային/Ելքային ելուստների բաշխում (I/Os distribution),
6. True և Bar բիթային լարերի դասավորություն (True and Bar bit line ordering),
7. բիթային լարերի ոլորումների բաշխում (bit line twisting distribution),
8. սյուների ոլորումների բաշխում (column twisting distribution),
9. բառային լարերի ոլորումների բաշխում (word line twisting distribution),
10. տողերի ոլորումների բաշխում (rows twisting distribution),
11. տրամաբանական սխեմաների բաշխում (straps distribution),
12. ավելցուկային տողերի և սյուների բաշխում (redundancy rows and columns distribution),
13. դատարկ՝ պաշտպանիչ, հավասարակշռող տողերի և սյուների բաշխում (dummy rows and columns distribution):

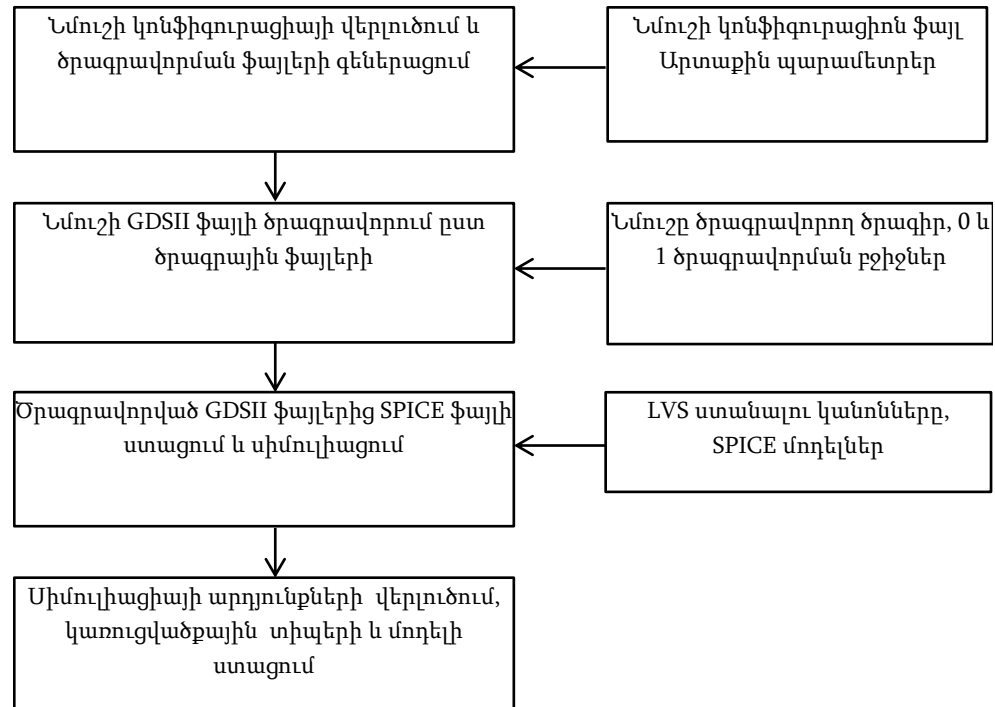
Ակնհայտ է, որ ՀՄ-երի զարգացման հետ միասին թվարկված կառուցվածքային տիպերի տեսակները ենթակա են փոփոխման և լրացման:

Կառուցվածքային մոդելի ավտոմատ դուրսբերման ծրագրային միջավայրի ֆունկցիոնալությունը և հիմնական դրույթները: Այսպիսով, կառուցվածքային մոդելը բոլոր ՆՀՄ սարքերում աշխատող ծրագրային միջոցների պարտադիր բաղադրամաս է: Մեկ այլ բարդ և կարևորագույն խնդիր է այդ մոդելի ճշտության ստուգումը: Ներկայումս էլեկտրոնային սարքերը նախագծելիս ՆՀՄ-երի կառուցվածքային մոդելը գեներացվում է ՀՄ-ն նախագծողի կողմից (manual method) կամ էլ՝ ավտոմատ կերպով մի շարք ծրագրային մեթոդներով: Կառուցվածքային մոդելի՝ նախագծողի կողմից գեներացված մեթոդի բացասական կողմերից են՝ 1. ճարտարագետից պահանջվում է ունենալ բարձր որակավորում և սպառիչ, լիարժեք տեղեկություն ՀՄ կառուցվածքի մասին, 2. ստացված մոդելը պարտադիր կերպով պետք է անցնի ստուգման (վավերացման) փուլ, որն էլ հանգեցնում է լրացուցիչ ռեսուրսների ծախսերի:

Կառուցվածքային մոդելի ավտոմատ գեներացման մեթոդը գերծ է վերոնշյալ թերություններից, բայց պահանջում է ավտոմատ ծրագրային միջոցների՝ միջավայրի ստեղծում: ՄՊԴՀ-երի հատկություններից է այդ սարքերի կանոնավոր կառուցվածքը, որը մեծ խոչընդոտ է կառուցվածքային մոդելի ավտոմատ դուրսբերման համար: Այդ խոչընդոտը շրջանցելու նպատակին է ծառայում հիշողության բջիջների «ծրագրավորման»՝ բջիջների անորոշ պարունակությունը հարկադրաբար, նախապես որոշված տրամաբանությամբ հաստատուն զրո կամ մեկ արժեքներից որևէ մեկի մեջ ձևափոխման գաղափարը [7]:

Նկ. 3-ում բերված է ՀՄ-երի նմուշի գրաֆիկական GDSII ձևաչափից կառուցվածքային մոդելը դուրս բերող ծրագրային միջավայրի ֆունկցիոնալ սխեման: ՀԿՄ-ը

ստանալու նպատակով նախագծված ծրագրային միջավայրը կատարում է հետևյալ գործողությունների հաջորդականությունը.



Նկ. 3. Կառուցվածքային մոդելը դուրս բերող ծրագրային միջավայրի ֆունկցիոնալ սխեմա

1. Նմուշի կոնֆիգուրացիայի վերլուծությունը, որի արդյունքում գեներացվում են նմուշի ծրագրավորման ֆայլերը (ՄՖ): Գեներացված ՄՖ-երն ունեն տեքստային ձևաչափ և պարունակում են տեքստային տեղեկություն հիշողության մեջ ծրագրավորվող օրինակների (ՄՕ) (programmed pattern) վերաբերյալ: Այդ օրինակների քանակը փոփոխական է և կախված է հիշողության նմուշի ֆիզիկական չափերից: Լավագույն տարբերակում այդ քանակը հավասար է երկուսի՝ զրոյական և անկյունագծային (նկ. 4), իսկ վատագույն դեպքում կարող է հասնել մինչև հինգի: Լավագույն դեպք է համարվում հիշողության նմուշի այն ֆիզիկական կառուցվածքը, երբ հիշողության բջիջներով զբաղեցված մակերեսը քառակուսի է, այսինքն՝ ֆիզիկական տողերի և սյուների քանակը նույնն է: Այս դեպքում հնարավոր է դառնում մեկ ծրագրավորող «անկյունագծային» օրինակով ծածկել հիշողության ողջ մակերեսը: Նկ. 4-ում ներկայացված են օրինակներ զրոյական և անկյունագծային ՄՕ-երը 8 տող և 8 սյուն պարունակող հիշողության նմուշի համար: Զրոյական ՄՕ-ն պարտադիր է նմուշների բոլոր տարբերակների համար: ՄՕ-երը մանրամասնորեն նկարագրված են [7]-ում:

2. Օգտագործելով ստացված ԾՕ-երը՝ ծրագրային հոսքը կատարում է նմուշի

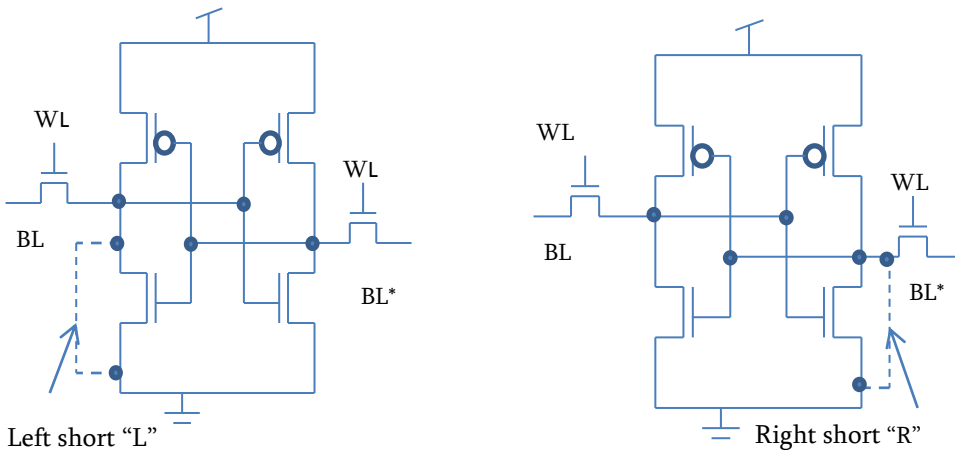
00000000	11111111	գրաֆիկական GDSII ֆայլի ծրագրավորում ծրագրավորող բջիջներով (նկ. 5), որի արդյունքում ստացվում են ԾՕ-ի քանակությամբ ծրագրավորված GDSII ֆայլեր:
00000000	01111111	
00000000	00111111	
00000000	00011111	
00000000	00001111	
00000000	00000111	
00000000	00000011	
00000000	00000001	

ա. գրայական բ. անկյունագծային

Նկ. 4. Ծրագրային օրինակները 8x8 հիշողության նմուշի համար

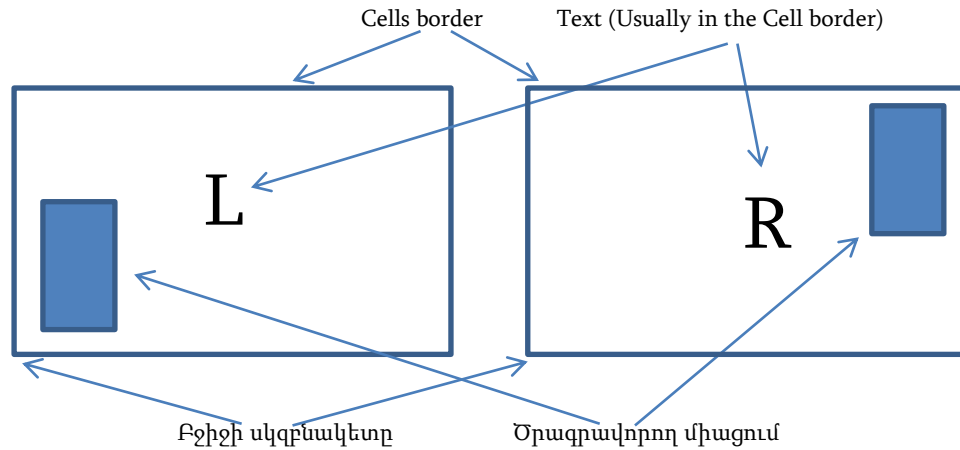
3. Օգտվելով GDSII ֆայլից՝ SPICE ձևաչափի ֆայլեր ստանալու կանոններով⁴⁶³(LVS rule deck) ԾՀ-ն դուրս է բերում համապատասխան SPICE ֆայլերը և կատարում այդ ֆայլերի մոդելավորումը: Մոդելավորման ընթացքում ԾՄ կատարում է հիշողության նմուշից «կարդալ» գործողությունը, որի արդյունքում ստացված և համապատասխան ձևով մշակված տեքստային ձևաչափով ֆայլերը

պարունակում են նմուշի մոդելը ստանալու համար անհրաժեշտ կառուցվածքային տարրերի, տիպերի մասին տեղեկություն:



Նկ. 5. Հիշողության ծրագրավորող բջիջների էլեկտրական սխեման

Հիշողության նմուշի 0 և 1 վիճակը ծրագրավորող բջիջների էլեկտրական սխեման ներկայացված է նկ. 5-ում: Հիշողության բջիջն ունի սիմետրիկ կառուցվածք, և հարկադրաբար 0 և 1 վիճակ ստանալու համար բավական է խախտել սիմետրիկությունը՝ ավելացնելով կարճ հոդակցում սխեմայի աջ կամ ձախ մասերում: Նկ. 6-ում ներկայացված է հիշողության բջիջների ծրագրավորումն ապահովող սխեմայի տուպոլոգիան: Ծրագրային հոսքը տեղադրում է ծրագրավորող բջիջները հիշողության բջիջների վրա՝ փոխելով բջիջների արժեքները, ապահովելով հիշողության ծրագրավորումը ըստ ԾՕ-ի ֆայլերի:



Նկ. 6. Հիշողության ծրագրավորման բջջերի տապալուզիան

Փորձնական արդյունքներ: Կառուցվածքային մոդելը ավտոմատ կերպով դուրս բերող ծրագրային հոսքը հաջողությամբ օգտագործվել է տարբեր տեխնոլոգիաների (45 նս, 28 նս) հիշողությունների կառուցվածքային մոդելները ստանալու նպատակով: ԾՀ-ի աշխատանքի միջին տևողությունը կես ժամ է, և կարող է տևել մինչև մի քանի ժամ: ԾՀ-ի հիմնական ժամանակի մեջ մտնում են SPICE ֆայլերի ստացումը և դրանց մոդելավորումը: SPICE ֆայլերի մոդելավորման ժամանակը ուղղակիորեն կախված է (ուղիղ համեմատական է) հիշողության նմուշի մեծությունից՝ բջջերի քանակից: Բացի այդ, միջավայրի աշխատանքի ժամանակը կախված է նմուշի երկրաչափական տեսքից՝ ԾՕ-ի քանակից: Աղյուսակում բերվում են փորձնական արդյունքներ տարբեր տեխնոլոգիաների և կոնֆիգուրացիաների հիշողության նմուշների համար:

Աղյուսակ

NW	NB	CM	PR	PC	NPP	ST	TT
2048	24	4	512	96	4	30	120
2048	32	8	256	256	2	30	60
1536	24	16	96	384	4	30	120
2048	192	4	512	768	3	30	90
1536	16	8	192	128	3	30	90
192	256	4	48	1024	5	30	150
2048	12	4	512	48	5	30	150

NW (Number of Words) – բառերի քանակը հիշողությունում,
 NB (Number of Bits) – հիշողության Մուտքերի/Ելքերի քանակը,
 CM (Column Mux) – սյուների փոխանցատիչ,
 PR, PR=NW/CM - ֆիզիկական տողերի քանակը հիշողությունում,

PC, $PC = NB \times CM$ - ֆիզիկական սյուների քանակը հիշողությունում,
 NPP (Number of Programing Patterns) - ծրագրավորման օրինակների քանակը,
 ST (Simulation Time in minutes) - նմուշի SPICE ֆայլի մոդելավորման ժամանակը,
 TT (Total Time in minutes) $TT = NPP \times ST$ - ընդհանուր ժամանակը:

Փորձերի ընթացքում նկատվեց, որ մոդելի դուրսբերման ժամանակը կախված չէ հիշողության պատրաստման տեխնոլոգիայից, և ԾՀ-ի աշխատանքի տևողությունը հիմնականում կախված է նմուշի կոնֆիգուրացիայի պարամետրներից (NW, NB, CM) և կատարվող մոդելավորման ժամանակից:

Իհարկե, ԾՀ-ում աշխատանքի ընդհանուր ժամանակը գնահատելիս անհրաժեշտ է նախապես հաշվի առնել նախապատրաստական աշխատանքների վրա ծախսված ժամանակը, որը կարող է տևել 4-ից 8 ժամ: Ավտոմատ կերպով դուրսբերման միջավայրում ստացված կառուցվածքային մոդելը վավերացված է և պատրաստ է անմիջական օգտագործման:

ԳՐԱԿԱՆՈՒԹՅԱՆ ՑԱՆԿ

1. Advanced Test Methods for SRAMs. Effective Solution for Dynamic Fault Detection in Nanoscaled Technologies / **A. Bosio, L. Dilillo, P. Girard, et al.**-Springer. 2010. -171 p.
2. **Amirkhanyan K.** Defect injection and a memory test algorithms verification flow // CSIT. - 2011 -P. 283-286.
3. Application of defect injection flow for fault validation in memories / **K. Amirkhanyan, A. Davtyan, G. Harutyunyan, et al** // EWDTS. – 2012.- P. 19-22.
4. US Patent 7786840 Memory Modeling Using an Intermediate Lave Structural Description/ **Y. Zorian, V. Vardanian, K. Aleksanyan, K. Amirkhanyan.** - August 3, 2010.
5. US patent 8112730 Various methods and apparatuses for memory modeling using a structural primitive verification for memory compilers/ **K. Amirkhanyan, K. Aleksanyan, S. Karapetyan, A. Shubat, S. Shoukourian, V. Vardanian, Y. Zorian.** - Feb.7, 2012.
6. US 8850277 B2 DETECTING RANDOM TELEGRAPH NOISE INDUCED FAILURES IN AN ELECTRONIC MEMORY/ **K. Amirkhanyan, A. Davtyan, G. Harutyunyan, T. Melkumyan, S. Shoukourian, V. Vardanian, A. Shubat, Y. Zorian.** - Sep. 30, 2014.
7. Generation of Memory Structural Model Based on Memory Layout / **K. Amirkhanyan, K. Darbinyan, A. Davtyan, G. Harutyunyan, S. Shoukourian, V. Vardanian, Y. Zorian;** US Patent application number: 20130346056 Publication date: 2013-12-26 (<http://www.faqs.org/patents/app/20130346056>)

Միևնուսիս Արմենիա ՓԲԸ: Նյութը ներկայացվել է խմբագրություն 20.05.2014:

К.С. АМИРХАНИЯН

**АВТОМАТИЧЕСКОЕ ИЗВЛЕЧЕНИЕ СТРУКТУРНОЙ МОДЕЛИ ОБРАЗЦА
ПАМЯТИ ИЗ ГРАФИЧЕСКОГО ФОРМАТА GDSII**

Представлена программная среда, которая позволяет в автоматическом режиме извлекать структурную модель образца статической памяти из используемого в производстве файла графического формата **GDSII**. Приведены функциональная и структурная блок-схемы программной среды. Рассмотрены проблемы, сложности и ограничения, связанные с извлечением структурной модели, а также некоторые результаты экспериментов.

Ключевые слова: структурная модель, образец памяти, **GDSII** формат образца, программная среда, статическая память произвольной выборки (СППВ), структурные типы.

K.S. AMIRKHANYAN

**AUTOMATIC STRUCTURAL MODEL EXTRACTION FROM THE MEMORY
INSTANCE GRAPHICAL FORMAT GDSII**

A program flow is presented which allows to automatically extract the structural model of SRAM instance from the graphical file GDSII. The functional and structural block diagrams are presented. Problems, difficulties and limitations connected with the extraction of the structural model as well as some results of experiments are considered.

Keywords: structural model, SRAM, memory instance, instance GDSII format, program flow.