ՀՀ ԿՐԹՈՒԹՅԱՆ ԵՎ ԳԻՏՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

ԵՐԵՎԱՆԻ ՊԵՏԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

ՍԵՐԳԵՅ ՀԱԿՈԲԻ ՄԱՆՈՒԿՅԱՆ

ԵՌԱՉԱՓ ՏՐԱՆԶԻՍՏՈՐՆԵՐՈՎ ՄԻԿՐՈԷԼԵԿՏՐՈՆԱՅԻՆ ՍԽԵՄԱՆԵՐԻ ՆԱԽԱԳԾՄԱՆ ՄԻՋՈՑՆԵՐԻ ՄՇԱԿՈՒՄԸ

ԱՏԵՆԱԽՈՍՈՒԹՅՈՒՆ

Ե.27.01 «Էլեկտրոնիկա, միկրո և նանոէլեկտրոնիկա»
մասնագիտությամբ տեխնիկական գիտությունների
թեկնածուի գիտական աստիճանի հայցման համար

Գիտական ղեկավար՝ ՀՀ ԳԱԱ թղթակից անդամ,
ՀՀ գիտության վաստակավոր գործիչ,
տ.գ.դ., պրոֆ. Վ. Շ. Մելիքյան

ԵՐԵՎԱՆ 2017

ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆ

ՆԵՐԱԾՈՒԹՅՈՒՆ

<u>Թեմայի արդիականությունը:</u> Ինտեգրալ սխեմաների (ԻՍ) զարգացման ողջ ընթացքն ուղեկցվել է դրանցում տարբեր տեսակի տրանզիստորների օգտագործմամբ և վերջիններիս երկրաչափական չափերի փոքրացմամբ՝ մասշտաբավորմամբ: Մասշտաբավորման հիմնական նպատակը ԻՍ-երի ֆունկցիոնալ հնարավորությունների և դրանցում կիսահաղորդչային սարքերի ինտեգրման աստիճանի մեծացումն է: Միկրոէլեկտրոնային սխեմաներում առաջինն օգտագործվել են երկբևեռ տրանզիստորները, որոնք հայտնի են իրենց բարձր արագագործությամբ: Սակայն վերջիններիս ներկայացվող անհրաժեշտ պահանջները չբավարարող էներգասպառումը և բարդ մասշտաբավորվելիությունը խոչընդոտում էին դրանցով կառուցվող ԻՍ-երի հետագա զարգացումը:

Երկբևեռ տրանզիստորները փոխարինվեցին մետաղ-օքսիդ-կիսահաղորդիչ (ՄՕԿ) տրանզիստորներով, որոնք երկբևեռների համեմատ ավելի դանդաղագործ էին, սակայն ունեին ԻՍ-երին անհրաժեշտ ցածր էներգասպառում ապահովելու հնարավորություն, իսկ հեշտ մասշտաբավորվելիության շնորհիվ հնարավոր դարձրին միկրոէլեկտրոնային սխեմաներում տարրերի ինտեգրման աստիճանի շարունակական բարձրացումը: Այդ տրանզիստորների երկու տեսակները՝ nՄՕԿ եւ pՄՕԿ, ընկած են կոմպլեմենտար ՄՕԿ (ԿՄՕԿ) ԻՍ-երի կառուցման հիմքում, որոնք երկար տարիներ հանդիսացել են սխեմաների կառուցման արդյունավետ տեխնոլոգիական լուծումներ: ԿՄՕԿ ԻՍ-երի մասշտաբավորման ընթացքում դրանց հիմնական մակրոբնութագրերի՝ արագագործության, ցածր էներգասպառման և կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի բարելավմանը խոչընդոտում են փոքր չափերով ՄՕԿ տրանզիստորներում ի հայտ եկող կարճ-հոսքուղային երևույթները: Դրանք հանգեցնում են հոսքուղու երկարությունից կախված շեմային լարման, լիցքակիրների՝ հագեցման արագություն ձեռք բերելու կարճ տևողության, արտաբերի հոսանքի՝ հոսքուղու երկարությունից հակադարձ համեմատականությունը կորցնելուն և տրանզիստորի՝ ժամանակի ընթացքում դեգրադացվող բնութագրերին: Նշված երևույթներն էական ազդեցություն ունեցան 22 *նմ*-ից փոքր չափերով տրանզիստորների դեպքում, երբ ենթաշեմային կորստի հոսանքները ձեռք բերեցին օգտակար հոսանքներին համեմատելի արժեքներ, իսկ ՄՕԿ սարքերի հետագա մասշտաբավորումը դարձավ աննպատակահարմար: Դրանց փոխարինման նպատակով մշակվել են սիլիցիումը մեկուսիչի վրա (ՍՄՎ) տրանզիստորները, որոնց արտադրման բարդությունները, սակայն, խոչընդոտում են հանդիսանում դրանց լայն կիրառությունը:

ՄՕԿ տեխնոլոգիային փոխարինելու համար ստեղծվեցին եռաչափ բազմափական տրանզիստորներ (ԵՏ), որոնք, ՄՕԿ տրանզիստորներից ունեցած կառուցվածքային էական տարբերությունների և աշխատանքի բոլորովին այլ սկզբունքի շնորհիվ, հնարավոր են դարձնում ԻՍ-երի՝ 22 *նմ*-ից հետո չափերի մասշտաբավորումը՝ ապահովելով էներգասպառման և արագագործության անհրաժեշտ մակարդակ:

Տարբեր հեղինակների կողմից մշակվել են ԵՏ-երով սխեմաների նախագծման մի շարք միջոցներ, որոնք գերազանցապես օգտագործում են միափական ՄՕԿ սարքերին բնութագրերով համանման, կարճ միացված փականներով եռաչափ սարքեր: Իրենց կառուցվածքի պատճառով այդ սարքերի բնութագրերի ղեկավարելիությունը խիստ սահմանափակ է: Ժամանակակից ԻՍ-երին առաջադրվող էներգասպառման, արագագործության և կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի խիստ պահանջները բավարարելու նպատակով առաջանում է անկախ փականային միացմամբ եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների մշակման ծայրահեղ անհրաժեշտություն:

<u>Հետազոտության առարկան:</u> Եռաչափ տրանզիստորներով սխեմաների նախագծման սկզբունքներն ու բարդությունները, ստատիկ հիշողության բջիջների, դոմինո տրամաբանական փականների և ստատիկ տրիգերների նախագծման եղանակներն ու առանձնահատկությունները, անկախ փականային միացմամբ եռաչափ տրանզիստորներով վերջիններիս նախագծման միջոցների մշակման սկզբունքները:

<u>Աշխատանքի նպատակը:</u> Կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի աննշան մեծացմամբ՝ հնարավորինս բարձր արագագործություն և ցածր էներգասպառում ունեցող եռաչափ տրանզիստորներով ԻՍ-երի նախագծման միջոցների մշակումն ու հետազոտումը:

<u>Հետազոտության մեթոդները:</u> Ատենախոսության կատարման ընթացքում օգտագործվել են եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների կառուցման տեսությունները, կարճ միացված և անկախ փականներով կառուցվածքների առանձնահատկությունները, սխեմատեխնիկական նկարագրման լեզուներով սխեմաների նկարագրությունները, ծրագրավորման օբյեկտակողմնորոշված լեզվով ծրագրային գործիքների իրականացման սկզբունքները:

<u>Գիտական նորույթը.</u>

- Ձևակերպվել են եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների մշակման միջոցներին առաջադրվող պահանջները, որոնց բավարարումն ապահովում է արագագործության, էներգասպառման և կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի անհրաժեշտ փոխզիջումային արժեքները:
- Առաջարկվել է եռաչափ տրանզիստորներով ստատիկ հիշողության բջիջների նախագծման 4 մեթոդ, որոնք, օգտագործելով դրանց անկախ փականներ ունենալու հատկությունը, հայտնի լուծումների համեմատ ապահովում են բջջի ստատիկ աղմկակայունության պաշարի անհրաժեշտ մեծ արժեք և փոքր ստատիկ էներգասպառում՝ կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի ոչ էական մեծացման հաշվին:
- Մշակվել է եռաչափ տրանզիստորներով դոմինո տրամաբանության դինամիկ հանգույցում լիցքի պաշտպանման միջոց, որը, փականի զբաղեցրած մակերեսի աննշան ավելացման հաշվին, առկա մոտեցումների համեմատ ապահովում է էապես ցածր էներգասպառում և անհրաժեշտ բարձր արագագործություն:
- Ստեղծվել է ստատիկ տրիգերների՝ էներգասպառման չնչին ավելացման հաշվին ստատիկ աղմկակայունության պաշարը և արագագործությունը բարելավող մեթոդ, որի հիման վրա տրիգերի հետադարձ կապի շրջիչն իրականացվել է անկախ փականներով եռաչափ տրանզիստորներով:

<u>Պաշտպանության են ներկայացվում հետևյալ դրույթները.</u>

- եռաչափ տրանզիստորներով ստատիկ հիշողության բջիջների նախագծման 4 մեթոդը
- եռաչափ տրանզիստորներով դոմինո տրամաբանության դինամիկ հանգույցում լիցքի պաշտպանման միջոցը
- ստատիկ տրիգերների՝ էներգասպառման չնչին ավելացման հաշվին ստատիկ աղմկակայունության պաշարը և արագագործությունը բարելավող մեթոդը

Աշխատանքի գործնական արժեքը: Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծումը առաջարկվող մեթոդների ու միջոցների օգնությամբ իրագործվել է 3D-Compiler ծրագրային գործիքի կողմից, որի շնորհիվ հնարավոր է դառնում ատենախոսությունում առաջարկված սխեմաների մշակումն իրականացնել հաշված վայրկյաններից մինչև մի քանի րոպեի ընթացքում: Առաջարկված մեթոդների կիրառմամբ իրագործված հիշողության սխեմաների էներգասպառումը, կարճ փականային միացմամբ եռաչափ տրանզիստորներով նախագծված սխեմաների համեմատ, բարելավվել է մոտ 36 – 66,7% -ով՝ կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի 2,38-7,66% վատթարացմամբ: Դոմինո փականների նախագծման առաջարկվող միջոցը ԿՓԵՏ կառուցվածքի համեմատ ապահովում է էներգասպառման 31,5-44,4% նվազեցում, ԿԲ-ի վրա զբաղեցրած մակերեսի 11-13,8% մեծացման հաշվին: Ստատիկ տրիգերների հետադարձ կապի շրջիչի նախագծման մեթոդը, 3D-Compiler ծրագրային գործիքով իրագործվելով, ԿՓԵՏ միացմամբ տրանզիստորների համեմատ ապահովում է տրիգերների շղթայի հապաղման 31,67-33,5% բարելավում՝ կորստի էներգասպառման 2,04-5,93% ավելացման հաշվին:

Գիտական դրույթների հավաստիությունը հիմնավորված է եռաչափ տրանզիստորներով սխեմաների նախագծման մշակված միջոցներով ԻՍ-երում լայնորեն կիրառվող սխեմաների նախագծմամբ, երբ այդ միջոցներով նախագծված սխեմաների համար ստացված արագագործության, էներգասպառման և կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի ցուցանիշները համընկնում են տեսական վերլուծություններով ակնկալվող արժեքներն:

Ներդրումը: 3D-Compiler ծրագրային միջոցը ներդրված է «Սինոփսիս Արմենիա» ՓԲԸ-ում: Այն օգտագործվում է ստատիկ աղմկակայունության մեծ պաշար պահանջող հիշողության մատրիցների, բարձր արագագործությամբ դոմինո փականների և հուսալի ստատիկ տրիգերների նախագծման համար, երբ անհրաժեշտ է ապահովել սխեմայի մակրոբնութագրերի փոխզիջումային արժեքներ:

Աշխատանքի փորձարկումը: Ատենախոսության հիմնական գիտական և գործնական արդյունքները զեկուցվել են՝

- 2-րդ միջազգային «IcETRAN: International Conference on Electrical, Electronic and Computing Engineering» գիտաժողովում (Սերբիա 2015թ.),
- 10-րդ միջազգային «CSIT: Computer Science and Information Technologies» գիտաժողովում (Երևան, Հայաստան, 2015թ.),
- 10-րդ միջազգային «Semiconductor Micro and Nanoelectronics» գիտաժողովում (Երեւան, Հայաստան, 2015թ.),
- 14-րդ միջազգային «EWDTS: East-West Design & Test» սիմպոզիումում (Երևան, Հայաստան, 2016թ.),
- «Սինոփսիս Արմենիա» ՓԲԸ գիտական սեմինարներում (Երեւան, Հայաստան, 2015-2017թթ.):

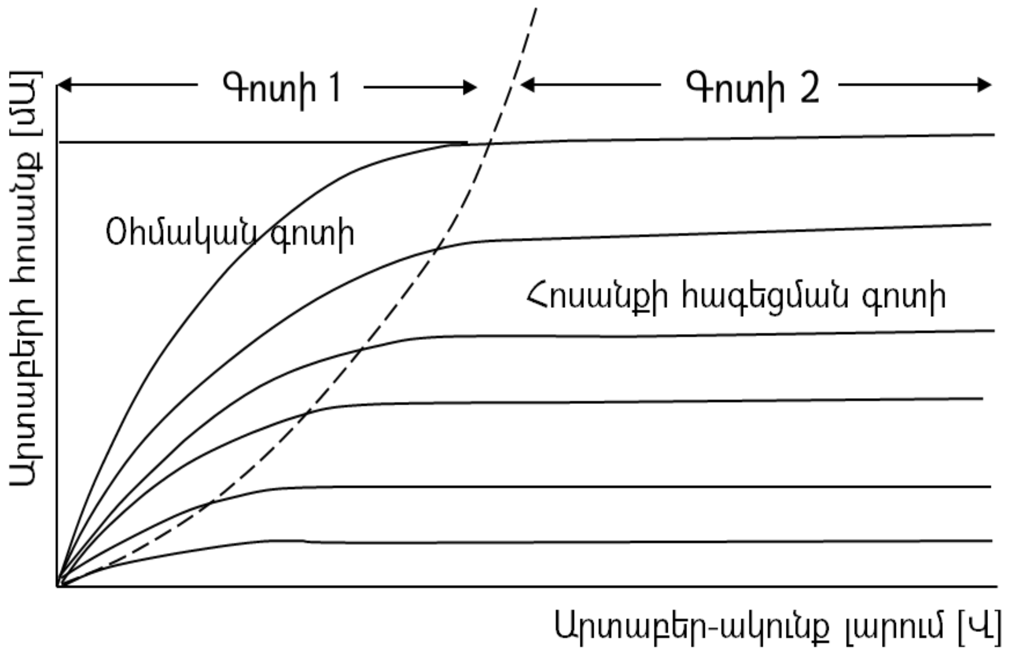
Հրապարակումներ: Ատենախոսության հիմնական դրույթները ներկայացված են 7 գիտական աշխատանքներում:

Ատենախոսության կառուցվածքը և ծավալը: Ատենախոսությունը բաղկացած է 3 գլուխներից, եզրահանգումից, 128 անուն գրականության ցանկից և 6 հավելվածներից: 1-ին հավելվածում ներկայացված է ատենախոսության ներդրման ակտը, երկրորդում՝ մշակված սխեմաների SPICE նկարագրությունները: 3-րդ հավելվածում բերված են մշակված ծրագրային միջոցի Python ծրագրավորման լեզվով նկարագրությունից հատվածներ, իսկ չորրորդում, հինգերորդում և վեցերորդում՝ ատենախոսությունում բերված նկարների, աղյուսակների և հապավումների ցանկերը: Հիմնական տեքստը կազմում է 115 էջ, որում ընդգրկված են 87 նկար և 10 աղյուսակ: Աշխատանքի ընդհանուր ծավալը, հավելվածների հետ միասին, կազմում է 145 էջ:

ԳԼՈՒԽ 1. ԵՌԱՉԱՓ ՏՐԱՆԶԻՍՏՈՐՆԵՐՈՎ ՄԻԿՐՈԷԼԵԿՏՐՈՆԱՅԻՆ ՍԽԵՄԱՆԵՐԻ ՆԱԽԱԳԾՄԱՆ ՄԻՋՈՑՆԵՐԻ ՄՇԱԿՄԱՆ ԸՆԴՀԱՆՈՒՐ ՀԱՐՑԵՐԸ

1.1. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման միջոցների մշակման անհրաժեշտությունը

Վերջին երեսուն տարիներին կիսահաղորդչային արդյունաբերության զարգացումը պայմանավորված էր ինտեգրալ սխեմաների (ԻՍ) արտադրության տեխնոլոգիաների զարգացմամբ [1-4], որի շարժիչ ուժը տրանզիստորների երկրաչափական չափերի շարունակական փոքրացումը՝ մասշտաբավորումն է [2,3]: Ժամանակից ԻՍ-երին [5] առաջադրվող տարրերի տեղաբաշխման մեծ խտության, սխեմաների բարձր արագագործության և ցածր էներգասպառման պահանջներն էլեկտրոնային սարքավորումների զարգացման հիմնական խթաններն են [6,7]: ԻՍ-երում հիմնական ակտիվ տարրը տրանզիստորն է [8-10], որով իրագործվում են նաև սխեմայի պասսիվ տարրերը՝ դիմադրությունները և ունակությունները [9, 11]: Վերջիններս, տրանզիստորի համեմատ, կիսահաղորդչային բյուրեղի վրա զբաղեցնում են ավելի մեծ մակերես [11] և դրսևորում են պարազիտային զգալի հատկություններ [12,13], ինչը, դրանց օգտագործման դեպքում, խոչընդոտում է ԻՍ-ում տարրերի ինտեգրման աճին [14] և սահմանափակում է վերջիններիս բարդության ավելացումը [15]: Մասնավորապես, ժամանակակից ԻՍ-երում լայնորեն կիրառվող մետաղ-օքսիդ-կիսահաղորդիչ (ՄՕԿ) դաշտային տրանզիստորը [16-20], գործելով իր վոլտ-ամպերային բնութագծի (նկ. 1.1) [18] գծային հատվածում, կարող է ծառայել որպես դիմադրություն: Դրա պատճառն այն է, որ նրանում տրանզիստորի արտաբերի հոսանքի կախվածությունն արտաբեր-ակունք լարումից գծային է [17,18], իսկ կանալի r դիմադրությունը համարժեք է նույն լարման անկմամբ և հոսանքով r պասսիվ դիմադրությանը [18], որի մեծությունը խիստ կախված է փական-ակունք լարումից [18]: ԻՍ-երում տրանզիստորների ունեցած դերի շնորհիվ [21]՝ միկրոէլեկտրոնիկայի զարգացումն ուղղակիորեն կապված է վերջիններիս տեխնոլոգիական առաջընթացի հետ:

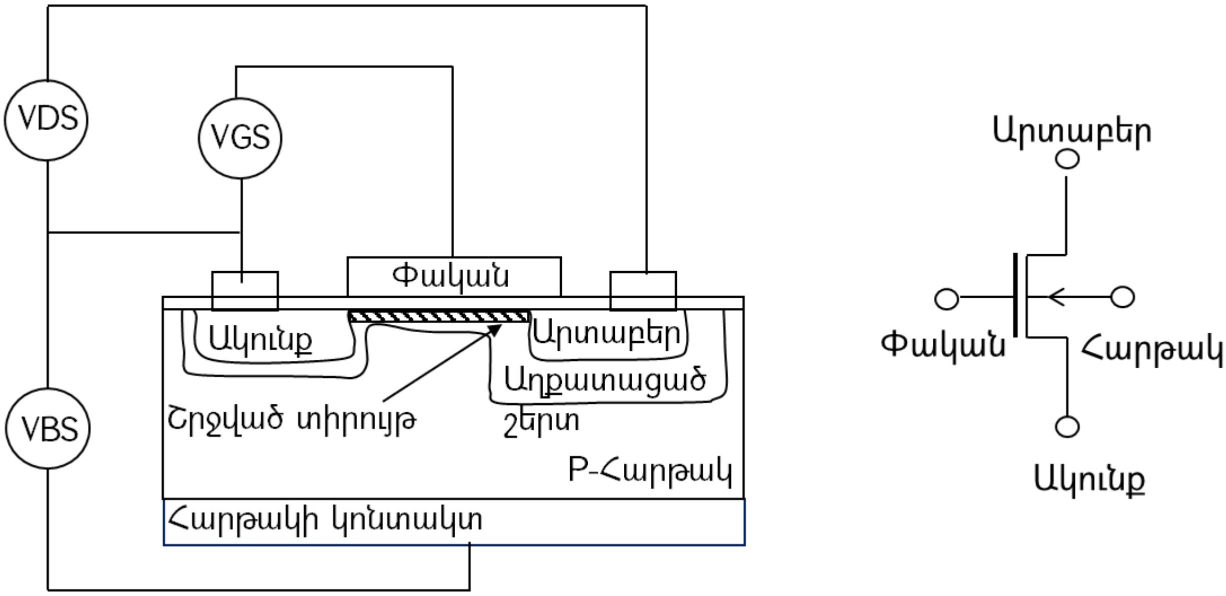


Նկ. 1.1. ՄՕԿ տրանզիստորի վոլտ-ամպերային բնութագիծը

Օգտագործված տրանզիստորների տեսակից [22] կախված՝ ԻՍ-երը լինում են՝

- երկբևեռ տրանզիստորների վրա հիմնված երկբևեռ ԻՍ-եր [23],
- ՄՕԿ դաշտային տրանզիստորների վրա հիմնված միաբևեռ ԻՍ-եր [24]:

Երկբևեռ տրանզիստորներն ԻՍ-երում սկսել են օգտագործվել 1960-ական թվականներից [23]: Դրանք առանձնանում են իրենց բարձր արագագործությամբ [25], սակայն ունեն մեծ էներգասպառում և բարդ մասշտաբավորվող են [26], ինչն էապես սահմանափակում է տրանզիստորի չափերի հետագա փոքրացումը և, հետևաբար՝ ԻՍ-ի ինտեգրման աստիճանի բարձրացումը [14]: Տարրերի տեղաբաշխման խտության մեծացումը հնարավոր դարձնելու համար ստեղծվեցին նոր տիպի տրանզիստորներ [16]. ԻՍ-երում երկբևեռ տրանզիստորներին փոխարինելու եկան դաշտային ՄՕԿ տրանզիստորները [17,18], որոնք ավելի դանդաղագործ են, քան երկբևեռները [27], սակայն, իրենց հեշտ մասշտաբավորելիության շնորհիվ [28] կարող են ապահովել ինտեգրման համեմատաբար բարձր աստիճան [14], ունեն ցածր էներգասպառում և արտադրության ավելի պարզ տեխնոլոգիական գործընթաց [29-31]: ՄՕԿ տրանզիստորը, որն անվանվում է նաև մեկուսացված փականով դաշտային տրանզիստոր (նկ. 1.2) [19] ունի 4 ելուստ՝ փական, արտաբեր, ակունք և հարթակ [17]:



Նկ. 1.2. ՄՕԿ տրանզիստորի լայնական կտրվածքը և սխեմատեխնիկական սիմվոլը

Փականի տակ առկա է մեկուսիչ շերտ [32], որը մեկուսացնում է փականն ակունքից և արտաբերից: Վերջիններս իմպլանտացված են հարթակի կիսահաղորդչի հիմնական լիցքակիրներին հակառակ լիցքակիրներով [33]՝ դրա հետ ձևավորելով pn անցում [34]: Փականի և ակունքի էլեկտրոդների միջև կիրառված պոտենցիալների տարբերությունը ղեկավարում է արտաբերի և ակունքի միջև հոսող հոսանքը [35]: Եթե փականի վրա կիրառված լարումը տրանզիստորի շեմային լարմանը մոտ արժեք ունի, բայց փոքր է դրանից, հարթակի հիմնական լիցքակիրները մղվում են դեպի հարթակի հատակը՝ փականի տակ առաջացնելով աղքատացված շերտ [36,37]: Երբ փականի վրա կիրառվում է շեմայինից բարձր լարում, ուղղահայաց էլեկտրական դաշտի շնորհիվ [38], հարթակի ոչ հիմնական լիցքակիրները հավաքվում են փականի տակ՝ առաջացնելով շրջված տիրույթ՝ հոսքուղի [39,40]: Ակունքի ու արտաբերի միջև պոտենցիալների տարբերության առկայության դեպքում՝ հոսքուղու լիցքակիրներն ուղղորդված շարժում են կատարում ակունքից արտաբեր՝ ձևավորելով արտաբերից դեպի ակունք հոսող էլեկտրական հոսանք [17,18,39,40]: Այդ հոսանքի՝ էլեկտրոններով պայմանավորված լինելու դեպքում տրանզիստորն անվանվում է n-հոսքուղով nՄՕԿ սարք [39], եթե հոսանքը պայմանավորված է խոռոչներով՝ p-հոսքուղով pՄՕԿ [39]:

pՄՕԿ տրանզիստորի էլեկտրական բնութագրերն ունեն nՄՕԿ-ի համապատասխան բնութագրերի հակառակ բևեռականությունը [41]:

Փականի տակ հոսքուղու առաջացման համար փականի և ակունքի միջև անհրաժեշտ է կիրառել տվյալ սարքի համար որոշակի նվազագույն շեմային լարում (ՇԼ) [42]: ՄՕԿ տրանզիստորի շեմային լարումը որոշվում է հետևյալ կերպ՝

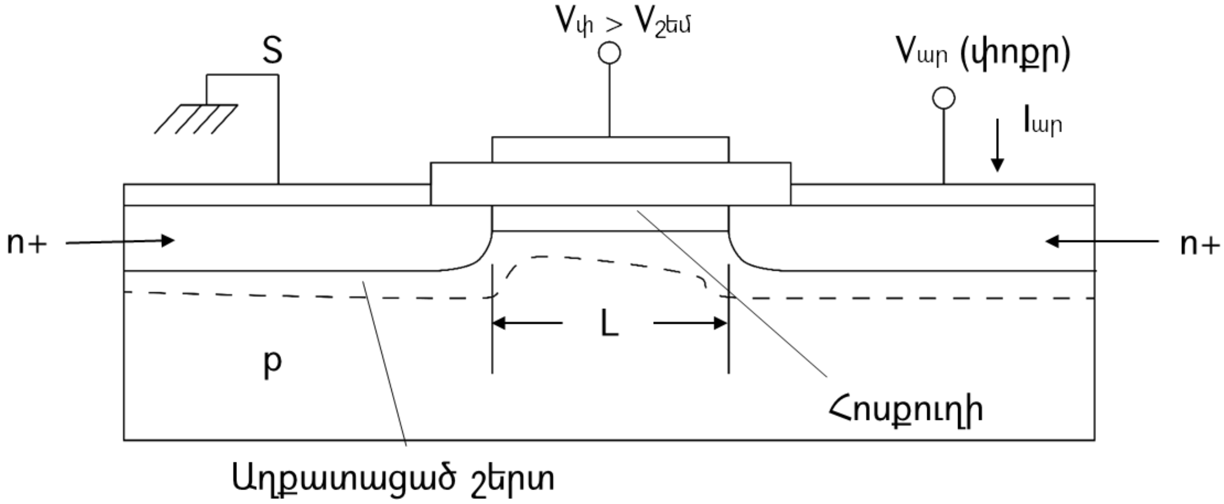
$$V_{շեմ} = V_{շեմ0} + \gamma\left(\sqrt{\left|V_{շեմ0} + 2\phi_F\right|} - \sqrt{\left|2\phi_F\right|}\right) , \qquad (1.1)$$

որտեղ $V_{շեմ0}$-ը տրանզիստորի շեմային լարումն է արտաբեր-ակունք պոտենցիալների տարբերության բացակայության պայմաններում, γ-ն՝ իրանի երևույթի գործակիցը, իսկ ϕ_F-ը՝ այդ տրանզիստորի ելքի աշխատանքը:

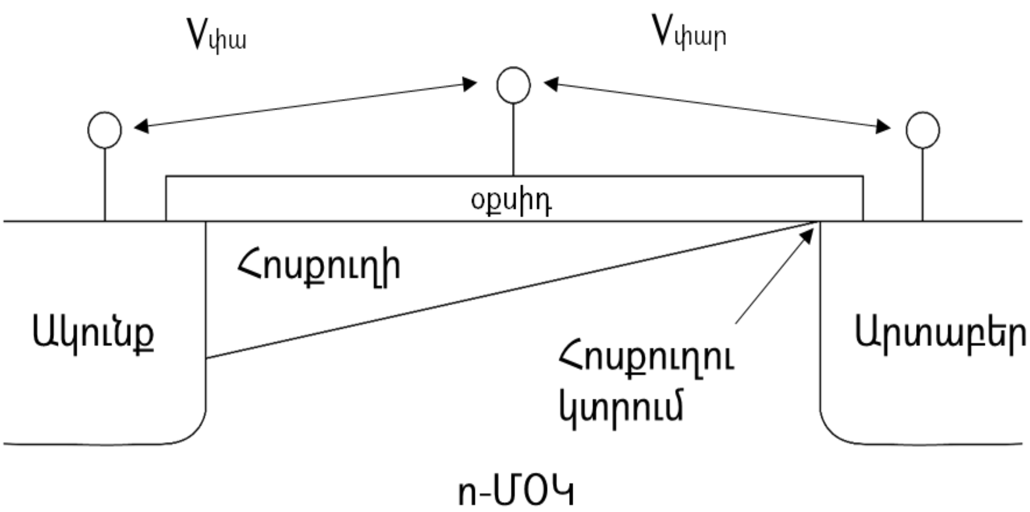
ՄՕԿ տրանզիստորի՝ մասշտաբավորման հետևանքով առաջացած վարքային փոփոխությունները [43] հանգեցրել են դրա երկու տարբեր մոդելներով՝ երկար-հոսքուղով և կարճ-հոսքուղով ՄՕԿ տրանզիստորների ստեղծմանը [43,44]: Այս դաշտային տրանզիստորն ունի աշխատանքի երեք ռեժիմ՝ անջատման, գծային (նկ. 1.3) և հագեցման (նկ. 1.4) [18,19]: Անջատման ռեժիմը համապատասխանում է հոսքուղու առաջացման համար չբավարարող փական-ակունք դրական լարմանը [20]: Այս ռեժիմում արտաբերից հոսող հոսանքն առաջին մոտավորությամբ բացակայում է [42]: Գծային ռեժիմում փականի և ակունքի միջև պոտենցիալների տարբերությունն աննշան մեծ է ՇԼ-ից [42], և փականի տակ ձևավորվում է հոսքուղի [39]: Արտաբեր-ակունք պոտենցիալի փոքր արժեքի դեպքում տրանզիստորը ցուցաբերում է լարմամբ ղեկավարվող դիմադրության [18] (ԼՂԴ) վարքագիծ: Աշխատանքի հագեցման ռեժիմում արտաբեր-ակունք լարումը բավարար մեծ է՝ արտաբերին կից մասում հոսքուղու կտրում առաջացնելու համար [40]: Գծային ռեժիմում ՄՕԿ ԼՂԴ-ի արժեքը ղեկավարվում է փական-ակունք լարմամբ [39]: Հոսքուղու երկայնքի վրա որևէ y կետով անցնող հոսանքն է [42]՝

$$I_{ար}(y) = qW \int_{x=0}^{\infty} \mu_t n(x,y) \frac{dV}{dy} dx , \qquad (1.2)$$

որտեղ $I_{\text{ար}}$-ը հոսքուղու երկայնքով անցնող հոսանքն է, μ_t-ն՝ էլեկտրոնի շարժունակությունը V-ը՝ y կետում տրանզիստորի լարումը, W-ը՝ տրանզիստորի լայնությունը, q-ը՝ հոսքուղու լիցքը:



Նկ. 1.3. Գծային ռեժիմում գտնվող ՄՕԿ տրանզիստորը



Նկ. 1.4. Հագեցման ռեժիմում գտնվող ՄՕԿ տրանզիստորը

(1.2)-ը ըստ հոսքուղու հաստության ինտեգրելով՝ ստացվում է [42]՝

$$I_{\text{ար}}(y) = -\mu_t W \frac{dV}{dy} Q_t(y), \qquad (1.3)$$

որտեղ $Q_t(y)$-ը հոսքուղու միավոր մակերեսում էլեկտրոնի լիցքն է:

Հոսքուղու միավոր մակերեսում լիցքը որոշող արտահայտությունը տեղադրելուց և ըստ հոսքուղու երկարության ինտեգրելուց հետո ստացվում է [42]՝

$$I_{ար} = \frac{\mu_t \varepsilon_{օք} W}{t_{օք} L}\left[\left(V_{փա} - V_{շեմ}\right)V_{աա} - \frac{V_{աա}^2}{2}\right], \tag{1.4}$$

որտեղ $\varepsilon_{օք}$-ն օքսիդի դիէլեկտրիկական թափանցելիությունն է, $t_{օք}$-ն՝ օքսիդի հաստությունը, $V_{աա}$-ը՝ տրանզիստորի արտաքեր-ակունք լարումը, $V_{փա}$-ը՝ տրանզիստորի փական-ակունք լարումը, իսկ L-ը՝ տրանզիստորի երկարությունը:

(1.4)-ն ունի մի քանի սահմանափակում.

- Արվում է ենթադրություն, որ հոսքուղու լիցքակիրների շարժումը կրում է բացառապես ուղղորդված բնույթ, սակայն շեմային լարման մոտ դրանք շարժվում են նաև լիցքերի՝ հոսքուղում դիֆուզման պատճառով [40]:
- Կարճ հոսքուղով տրանզիստորներում [45] ենթադրվում է, որ հոսքուղու բոլոր կետերում լիցքակիրներն ունեն միևնույն շարժունակությունը [46], սակայն հաշվի չի առնվում դրանց հագեցման արագությունը [47]:
- Փական-ակունք լարումից աղքատացված շերտի լիցքի փոփոխականության կախումն անտեսվում է [42], սակայն դա ճիշտ արդյունքի կհանգեցնի միայն թույլ լեգիրացված հոսքուղիների դեպքում [42]:

ՄՕԿ տրանզիստորը գտնվում է գծային ռեժիմում այնքան ժամանակ, քանի դեռ փական-ակունք լարումը (ՓԱԼ) ՇԼ-ից մեծ է, իսկ արտաքերին կից հոսքուղին դեռ չի կտրվել: Հոսքուղու կտրման երևույթը տեղի է ունենում, երբ [42]՝

$$V_{փա} - V_{աա} = V_{շեմ} : \tag{1.5}$$

(1.5)-ը սահմանում է գծային և հագեցման ռեժիմների սահմանը:

Հայտնի է [42,43], որ անջատման ռեժիմում՝ $|V_{փա}| < |V_{շեմ}|$ տրանզիստորով հոսող հոսանքն առաջին մոտավորությամբ զրո է: Սակայն, երբ ՓԱԼ-ը դառնում է ՇԼ-ից փոքր բայց դրան մոտ մեծություն, հոսքուղով հոսում է որոշակի մեծություն ունեցող հոսանք [39]: Այդ ենթաշեմային հոսանքը (ԵՀ) մեծ կարևորություն է ներկայացնում [29] ժամանակակից ցածր սնուցման լարմամբ և ցածր էներգասպառմամբ սխեմաներում, այդ պատճառով, չնայած գծային և հագեցման ռեժիմներում հոսանքը պայմանավորված է գլխավորապես լիցքակիրների ուղղորդված շարժմամբ, լիցքակիրների դիֆուզիայի հետևանքով առաջացած ԵՀ-ը ևս անհրաժեշտ է հաշվի առնել [42]: Ըստ էության, ՄՕԿ տրանզիստորը վարքով նմանվում է երկբևռ տրանզիստորին [48], որում ակունքը հոսքուղի է ներմուծում լիցքակիրներ, որոնք դիֆուզվելով դրա երկարությամբ՝ հավաքվում են արտաբերի կողմից: ԵՀ-ն ունի նույն ուղղությունը, ինչ հագեցման հոսանքը [49]: ԵՀ-ի՝ միայն լիցքերի դիֆուզիայով պայմանավորված լինելու դեպքում ստացվում է [42]՝

$$I_{ար} \approx K\left(1+m\right)\left(\frac{kT}{q}\right)^{2}\exp\left(\frac{q\left(V_{փա}-V_{շեմ}\right)}{mkT}\right), \tag{1.6}$$

$$S \equiv \left(\frac{d\left(\log_{10} I_{ար}\right)}{dV_{փա}}\right)^{-1}, \tag{1.7}$$

որտեղ k-ը Բոլցմանի հաստատունն է, T-ը՝ միջավայրի ջերմաստիճանը, m-ն՝ արդյունարար զանգվածը, իսկ K-ը՝ դիելեկտրիկական հաստատուն:

S-ը հայտնի է ենթաշեմային բացվածք [42,45,49,50] անվամբ և ցույց է տալիս փական-ակունք այն լարումը, որն անհրաժեշտ է տրանզիստորի հոսանքի մեծությունը 10 անգամ փոխելու համար [42]: Սենյակային ջերմաստիճանում ՄՕԿ տրանզիստորի աշխատանքը բնութագրվում է մոտ 100 *մՎ* ենթաշեմային բացվածքով [42]: Դա նշանակում է, որ ՓԱԼ-ի 100 մՎ փոփոխությանը համապատասխանում է ԵՀ-՝ 10 անգամ փոփոխությունը: Ասվածի կիրառական նշանակությունն այն է, որ տրանզիստորի շեմային լարման՝ 300 *մՎ*-ից փոքր արժեքների դեպքում մասշտաբավորումն ուղեկցվում է ՓԱԼ = 0 համապատասխան չափազանց մեծ ԵՀ-երով [42]: Տրանզիստորի իրանի երևույթի [51] միջոցով ՇԼ-ի ղեկավարումը նպաստում է ԵՀ-ի ազդեցության և արտադրության արդյունքում շեմային լարման անճշտությունների նվազեցմանը [42,51]:

Կիսահաղորդչային սարքերի մասշտաբավորման մեկ այլ հետևանք է երկար-հոսքուղով ՄՕԿ-ից բնութագրերով տարբերվող կարճ-հոսքուղով ՄՕԿ տրանզիստորը [45]: Դրա յուրահատկություններն են [45,48,49,50].

- ՇԼ-ը դառնում է հոսքուղու երկարությունից կախված: Դա հայտնի է կարճ-հոսքուղային երևույթ անվամբ:
- Հոսքուղում էլեկտրական դաշտի լարվածությունը շատ ավելի մեծ է, ինչը հանգեցնում է նրան, որ լիցքակիրների արագությունը համեմատաբար շուտ է հասնում իր առավելագույն արժեքին՝ հագեցման արագությանը:
- Հոսքուղու մոդուլյացիայի պատճառով հոսքուղու արդյունարար երկարությունը ֆունկցիա է դառնում արտաբեր-ակունք լարումից:

Նշված առանձնահատկությունները մեծ կարևորություն են ներկայացնում բարձր արագագործություն պահանջող սխեմաների նախագծման ժամանակ [42]:

Կարճ-հոսքուղային երևույթի [42,52,53] պատճառով տրանզիստորի ՇԼ-ի բացարձակ արժեքը հոսքուղու երկարության նվազմանը զուգընթաց նվազում է [42]: Դա պայմանավորված է նրանով, որ փականի տակ գտնվող աղքատացած շերտի որոշ լիցքեր հարում են նաև ակունքին և արտաբերին [42]: Կարճ հոսքուղով ՄՕԿ-երի ՇԼ-ն, երկար-հոսքուղով ՄՕԿ-ի հետ համեմատած, նվազում է (1.8)-ով [42] որոշվող չափով՝

$$\Delta V_{շեմ} \approx \frac{\varepsilon_{op}\left(L'-L\right)W_{աշ}}{2WLt_{op}} \approx -\frac{\varepsilon_{op}\left(W_{ակ}+W_{աո}\right)W_{աշ}}{2WLt_{op}}, \qquad (1.8)$$

որտեղ $W_{աշ}$ -ն աղքատացած շերտի առավելագույն լայնությունն է, իսկ $\left(W_{ակ}+W_{աո}\right)$-ն՝ ակունքի և արտաբերի աղքատացած շերտերի լայնությունների գումարը:

ՄՕԿ-երում, մասնավորապես կարճ-հոսքուղով տրանզիստորներում, կարևոր դեր է խաղում հոսքուղու երկարության մոդուլյացիան [54,55]՝ արտաբեր-ակունք լարման շարունակվող մեծացման հետևանքով հոսքուղու կտրման կետի՝ դեպի հոսքուղու խորքը տեղափոխվելը: Արդյունքում՝ հոսքուղու հոսանքն աճում է $L/\left(L-\Delta L\right)$ գործակցով [42]:

Հոսքուղու մոդուլյացիան հաշվի առնելիս գծային ռեժիմում տրանզիստորով հոսող հոսանքի համար ստացվում է [42]՝

$$I_{\text{ար}} = K\left[\left(V_{\text{փա}} - V_{\text{շեմ}}\right)V_{\text{աա}} - \frac{V_{\text{աա}}^{2}}{2}\right]\left[1+\lambda V_{\text{աա}}\right], \tag{1.9}$$

$$V_{\text{փա}} \geq V_{\text{շեմ}}, \left(V_{\text{փա}} - V_{\text{շեմ}}\right) \geq V_{\text{աա}}, \tag{1.10}$$

որտեղ λ-ն հոսքուղու մոդուլյացիայի գործակիցն է:

Հագեցման ռեժիմում նույն հոսանքի համար կստացվի [42]՝

$$I_{\text{ար}} = \frac{K}{2}\left(V_{\text{փա}} - V_{\text{շեմ}}\right)^{2}\left(1+\lambda V_{\text{աա}}\right); \tag{1.11}$$

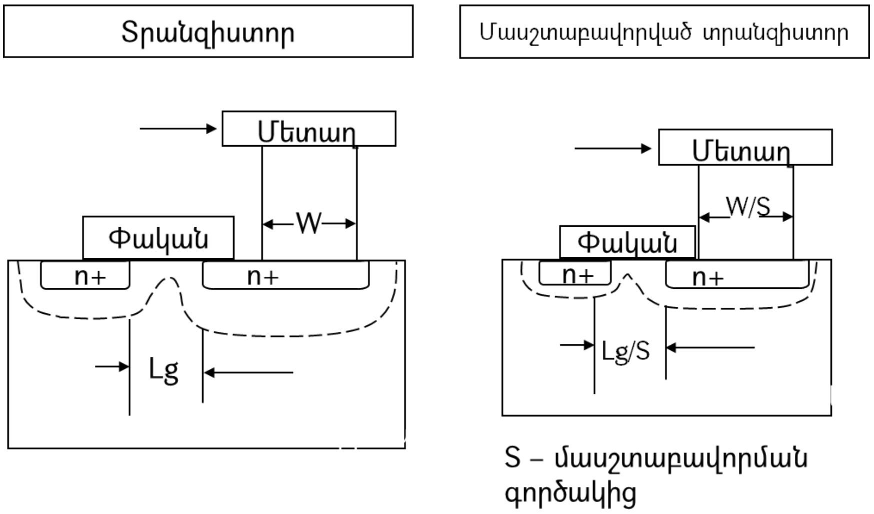
$$V_{\text{փա}} \geq V_{\text{շեմ}}, \left(V_{\text{փա}} - V_{\text{շեմ}}\right) \leq V_{\text{աա}} : \tag{1.12}$$

Թվային ԻՍ-երում ՄՕԿ տրանզիստորի արդյունավետությունը բնորոշվում է երկու հիմնական գործոններով [6,17]

- Անջատման և հագեցման ռեժիմներում հոսանքների հարաբերությամբ Iհագ/Iանջ, որտեղ Iհագ-ը տրանզիստորով հոսող հոսանքն է, երբ փական-ակունք պոտենցիալների տարբերությունը հավասար է սնուցման լարմանը, իսկ Iանջ-ը՝ V(փական-ակունք)=0 դեպքում հոսանքը:
- Տրանզիստորի՝ բաց վիճակից փակի փոխանջատման համար անհրաժեշտ ժամանակը: Այս հապաղումը պայմանավորվում է երկու գործոններով՝ հոսքուղու երկարությամբ և հոսքուղում լիցքակիրների շարժման արագությամբ: Ակնհայտ է, որ փոխանջատման փոքր ժամանակը սխեման դարձնում է արագագործ:

Ժամանակակից թվային ԻՍ-երը հիմնված են կոմպլեմենտար ՄՕԿ (ԿՄՕԿ) սխեմաների վրա [56,57], որոնց հիմքում ընկած են n և p տիպի հոսքուղիներով ՄՕԿ տրանզիստորները: ԿՄՕԿ-ի շնորհիվ հնարավոր դարձավ ապահովել ԻՍ-ի ցածր էներգասպառում [58,59]՝ միաժամանակ տրանզիստորների քանակը հասցնելով հարյուրավոր միլիոնների [60,61,62]: ԻՍ-երի զարգացման բոլոր փուլերն ուղեկցվել են ՄՕԿ տրանզիստորների չափերի մասշտաբավորմամբ (նկ. 1.5) [63]՝ հոսքուղու երկարությունը 5 *մկմ*-ից դարձնելով 22 *նմ* [64]: Երկրաչափական չափերի փոքրացմանը զուգահեռ, ՄՕԿ տրանզիստորների երկար-հոսքուղային վարքագիծը պահպանելու նպատակով [65], դրանց նախագծման մեթոդները պետք է փոփոխվեն համապատասխանորեն:

Շարունակական նվազման պատճառով հոսքուղու երկարությունը դառնում է ակունքի և արտաքերի աղքատացված շերտերի լայնությանը համեմատելի մեծություն [42], ինչի հետևանքով այդ երկու շերտերը միանում են իրար: Դա հանգեցնում է նրան, որ փականի տակ էլեկտրական դաշտը և, հետևաբար, արտաքերի հոսանքը դառնում են խիստ կախված ակունքի և արտաքերի լարումներից [42]:



Նկ. 1.5. ՄՕԿ տրանզիստորի մասշտաբավորումը

Այս խնդրից խուսափելու համար պահանջվում է հոսքուղու լրացուցիչ լեգիրացում [39,40,42]: Միևնույն ժամանակ, հոսքուղու հավելյալ լեգիրացումը հանգեցնում է տրանզիստորի շեմային լարման աճին [42]: Շեմային լարումն անհրաժեշտ արժեքին վերադարձնելու համար օքսիդի շերտը բարակեցնում են [66,67]:

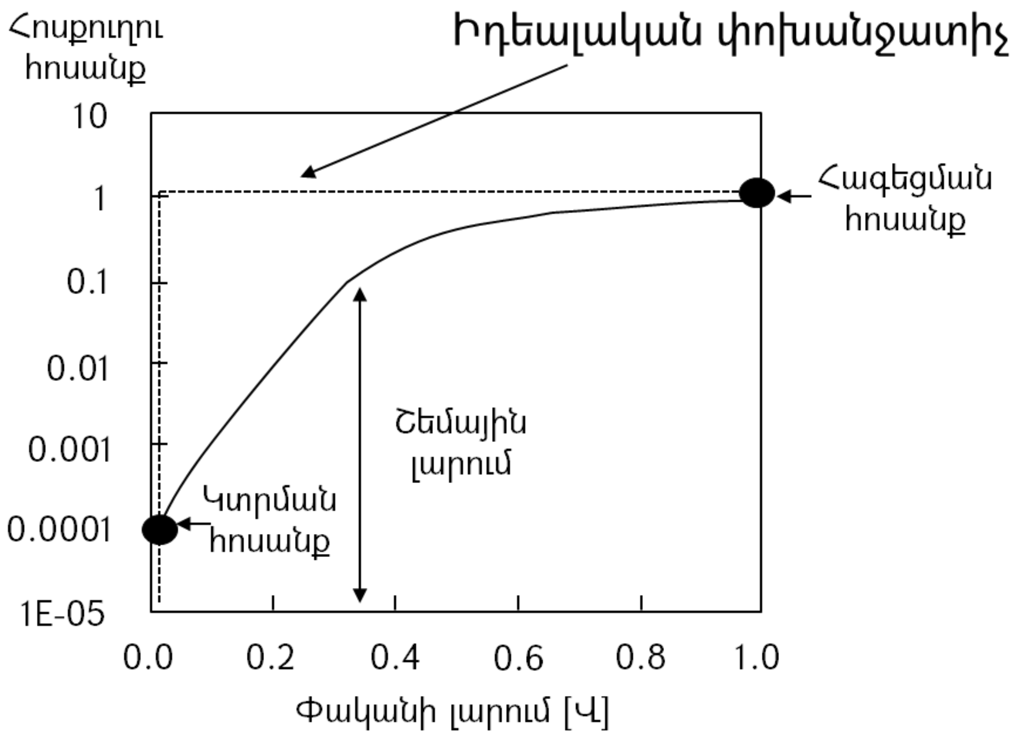
Կիսահաղորդչային սարքի պարամետրերի փոխկապակցվածության պատճառով, տրանզիստորի արդյունավետ աշխատանքն ապահովելու համար, մշակվել են մասշտաբավորման կանոններ [42,63,68-70]: Մասշտաբավորման՝ սահմանված կանոններով իրականացումը չի ապահովում լիարժեք երկար-հոսքուղային վարքագիծ. առաջանում են անխուսափելի շեղումներ՝ կարճ-հոսքուղային երևույթներ [42]: Դրանց ի հայտ գալու պատճառները հիմնականում փոքր չափերով տրանզիստորում պոտենցիալի երկչափ բաշխվածությունն է և հոսքուղու շրջակայքում բարձր էլեկտրական դաշտերը [39,42]: Հոսքուղում պոտենցիալի բաշխվածությունը պայմանավորված է փականով ղեկավարվող ուղղահայաց և արտաբերով ղեկավարվող հորիզոնական դաշտերով [42], որոնց համատեղ ազդեցությունը կարող է հանգեցնել անցանկալի վարքագծի:

Էլեկտրական դաշտի ուժեղացմանը զուգընթաց՝ հոսքուղու լիցքակիրների շարժունակությունը դառնում է դաշտից կախված, և հագեցման ռեժիմում հաղորդականությունը սահմանափակվում է [56]: Էլեկտրական դաշտի լարվածության հետագա մեծացման դեպքում արտաբերին կից լիցքակիրների քանակը բազմապատկվում է [17]՝ հանգեցնելով հարթակի հոսանքների: Կարճ-հոսքուղային երևույթներն ունեն հետևյալ ազդեցությունները [53]՝

- հոսքուղու երկարությունից կախված ՇԼ,
- լիցքակիրների՝ հագեցման արագություն ձեռք բերելու կարճ ժամանակ,
- հոսքուղու երկարությանը հակադարձ համեմատականությունը կորցրած արտաբերի հոսանք,
- տրանզիստորի՝ ժամանակի ընթացքում դեգրադացվող բնութագրեր:

Տրանզիստորի ճշգրիտ աշխատանքն ապահովելու համար անհրաժեշտ է վերացնել կամ ճնշել կարճ-հոսքուղային երևույթները:

Իդեալական դեպքում ՄՕԿ-ի մասշտաբավորման կանոնն է սարքերի բոլոր չափերի և լարումների՝ միևնույն չափով նվազեցումը [63] այնպես, որ ներքին էլեկտրական դաշտը մնա անփոփոխ: Մասշտաբավորման այդ տեսակը հայտնի է հաստատուն դաշտով մասշտաբավորում (ՀԴՄ) անվամբ [71]: S գործակցով փոքրացվում են հոսքուղու երկարությունը և լայնությունը, օքսիդի հաստությունը, pn-անցման խորությունը և բոլոր լարումները [71]: Հոսքուղու լեգիրացման աստիճանը մեծացվում է S անգամ [71]: Լարումների փոքրացումը հանգեցնում է աղքատացած շերտի լայնության նվազմանը [71]: Սակայն կան գործոններ, որոնք, լինելով չմասշտաբավորվող, խոչընդոտում են տրանզիստորների նման համաչափ մասշտաբավորումը [63]: Այդ գործոններից են pn-անցման սեփական լարումը և թույլ շրջման շերտի պոտենցիալը [42]: Նշված սահմանափակումների հիմնական պատճառը կիսահաղորդչի էներգետիկ արգելքի և ջերմային kT լարման հաստատունությունն է: Մյուս կողմից՝ օքսիդի հաստությունը ենթանանոմետրական չափերի հասցնելիս հնարավոր է դառնում լիցքակիրների թունելային անցումը [72,73]: pn-անցման ծակման հնարավորության պատճառով [74] հոսքուղու լեգիրացումը չի կարող ինչ-որ արժեքից մեծ դառնալ: Անջատման հոսանքների պատճառով շեմային լարումը չի կարող որոշակի արժեքից ավելի փոքր լինել [42]: Նշված բոլոր սահմանափակումները հաշվի առնելով՝ առաջարկվել են մասշտաբավորման այլ կանոններ [63,75,76]: Դրանցից են՝ հաստատուն լարմամբ մասշտաբավորումը, քվազի-հաստատուն լարմամբ մասշտաբավորումը և ընդհանրացված մասշտաբավորումը [75,76]: Դա թույլ է տալիս տրանզիստորի առանձին պարամետրերը մասշտաբավորել միմյանցից անկախ՝ ձգտելով պահպանել սարքի բնականոն վարքագիծը: Տրանզիստորի իրական և իդեալական փոխանջատման բնութագրերի (նկ.1.6) տարբերությունը կրճատելու, դրա աշխատանքային բնութագրերը բարելավելու համար՝ մասշտաբավորման գործընթացի նպատակն է դառնում տրանզիստորի հագեցման մեծ հոսանք, անջատման համեմատաբար փոքր հոսանք և փոխանջատման մեծ արագություն ապահովելը[63]:



Նկ. 1.6. ՄՕԿ տրանզիստորի իրական և իդեալական բնութագծերը

Հոսքուղու նվազագույն երկարությունը, որի դեպքում կիսահաղորդչային սարքը պահպանում է երկար-հոսքուղային վարքագիծը, պետք է բավարարի հետևյալ անհավասարությանը [42]՝

$$L \geq C_1 \left[r_j d \left(W_{ակ} + W_{աղ} \right)^2 \right]^{1/3}, \quad (1.13)$$

$$W_{աղ} = \sqrt{\frac{2\varepsilon_p}{qN_U} \left(V_ա + \psi_{հպ} + V_{հլ} \right)}, \quad (1.14)$$

որտեղ C_1-ն հաստատուն է, ε_S-ն՝ դիէլեկտրիկական թափանցելիությունը, $V_{հլ}$-ը՝ հարթակի լարումը, $\psi_{հպ}$-ը՝ հավասարակշռության պոտենցիալը, իսկ N_U-ն՝ ակցեպտորների խառնուրդների կոնցետրացիան:

Վերոհիշյալ սահմանափակումների պատճառով 22 *նմ* ՄՕԿ տրանզիստորի հետագա մասշտաբավորմումը դառնում է աննպատակահարմար՝ նշված բացասական ազդեցությունների անկառավարելի դառնալու պատճառով [63,64]: Տրանզիստորի ցանկալի բնութագրերը ստանալու և չափերի հետագա փոքրացումը հնարավոր դարձնելու համար մշակել են տրանզիստորների նոր տեսակներ [77,78], որոնք թույլ են տվել մասշտաբավորմանը զուգահեռ նվազեցնել և վերահսկել առաջացող ոչ օգտակար երևույթները:

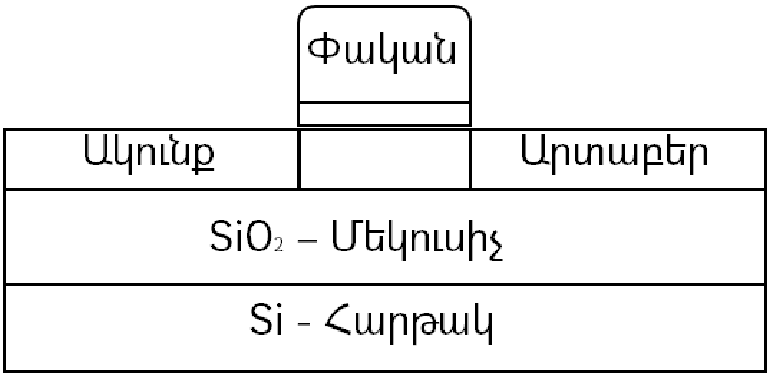
ՄՕԿ դաշտային տրանզիստորներին որպես այլընտրանք մշակված սարքերից են են սիլիցիումը մեկուսիչի վրա (ՍՄՎ) ՄՕԿ [77] և բազմափական տրանզիստորները [78]:

ՍՄՎ ՄՕԿ-ը (նկ. 1.7) տարբերվում է դասական ՄՕԿ-ից նրանով, որ կառուցվում է հարթակից սիլիցիումի երկօքսիդի հաստ շերտով առանձնացված բարակ սիլիցիումային շերտի վրա [77]: Երկօքսիդի միջոցով սարքերը էլեկտրականապես մեկուսացվում են հարթակից և միմյանցից: Դասական ԿՄՕԿ տեխնոլոգիայի նկատմամբ ՍՄՎ ԿՄՕԿ-ն ունի հետևյալ առավելությունները [77]՝

- նվազեցված պարազիտային ունակություններ,
- բարելավված ենթաշեմային բնութագրեր,
- իրանի երևույթի բացակայություն,

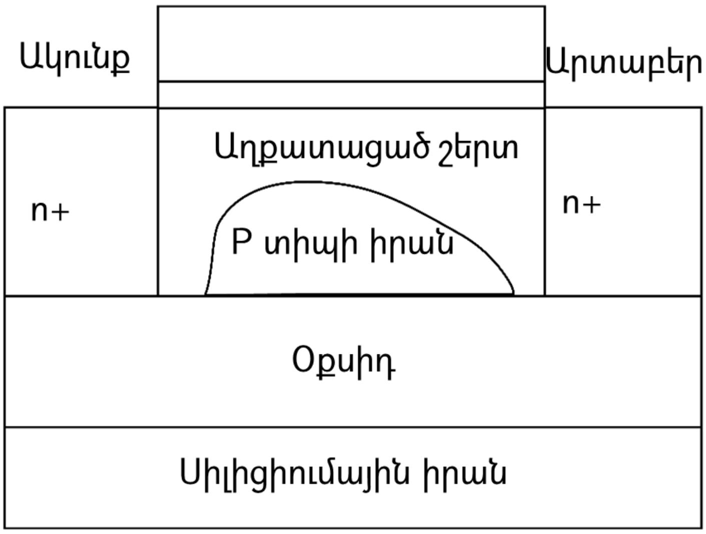
- բարելավված ռադիացիոն կայունություն:

Սիլիցիումի երկօքսիդն ունի շատ փոքր հաղորդականություն, ինչի շնորհիվ տրանզիստորները դրանով պատելը հանգեցնում է գրեթե զրոյական պարազիտային ունակությունների [77]: Վերջիններիս նվազեցումն իր հերթին մոտ 25 %-ով բարելավում է ազդանշանի տարածման հապաղումը [77]: Օքսիդի շերտով է պայմանավորված նաև ՍՄՎ տրանզիստորի ռադիացիոն կայունությունը [79]: Դասական ՄՕԿ տրանզիստորի համեմատ ունեցած առավելությունների շնորհիվ՝ ՍՄՎ տրանզիստորը նպատակահարմար է կիրառել ցածր էներգասպառում պահանջող սխեմաներում [77]:



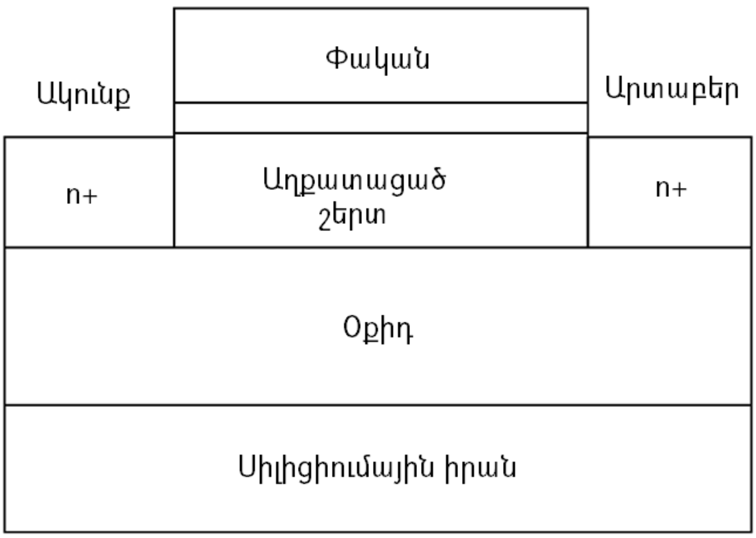
Նկ. 1.7. ՍՄՎ տրանզիստորի կառուցվածքը

Սիլիցիումի շերտի հաստությունից կախված՝ ՍՄՎ դաշտային տրանզիստորները լինում են մասնակի աղքատացած (ՄԱ) տիրույթով (նկ. 1.8) [80], որի դեպքում սիլիցիումային շերտի հաստությունն ավելի մեծ է, քան փականի աղքատացած շերտի առավելագույն հաստությունը, և լրիվ աղքատացած (ԼԱ) տիրույթով (նկ. 1.9) [80], որի դեպքում սիլիցիումի շերտի հաստությունը բավարար չափով փոքր է՝ աղքատացած շերտի ստեղծման համար:



Նկ. 1.8. ՄԱ ՍՄՎ տրանզիստորի կառուցվածքը

ՄԱ շերտով ՍՄՎ տրանզիստորների արտադրության համար սովորաբար օգտագործվում է 100 *նմ* հաստությամբ սիլիցիումի շերտ [77], իսկ ԼԱ ՍՄՎ-ի դեպքում՝ 50 *նմ* հաստությամբ շերտ [77]: Այդ պատճառով ԼԱ-ի դեպքում փականի տակ գտնվող տիրույթն ամբողջությամբ աղքատացվում է լիցքակիրներից [80], իսկ մասնակի աղքատացման դեպքում գոյություն ունի p տիպի իրան [80]:



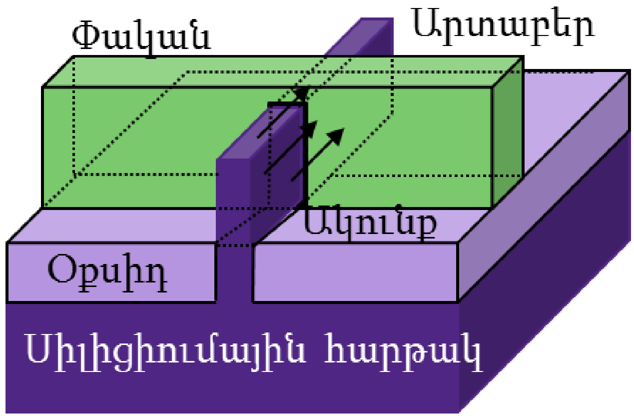
Նկ. 1.9. ԼԱ ՍՄՎ տրանզիստորի կառուցվածքը

Ընդհանուր առմամբ ՍՄՎ տրանզիստորի նշված երկու տեսակներն ունեն մի շարք առավելություններ, ինչի շնորհիվ երկուսն էլ կիրառվում են ԻՍ-երում: ՄԱ ՍՄՎ տրանզիստորների հիմնական առավելություններն են [77]՝

- օքսիդի շերտի հաստ լինելու շնորհիվ՝ համեմատաբար պարզ արտադրական գործընթացը,
- իրանի պոտենցիալավորման շնորհիվ՝ ղեկավարելի ենթաշեմային հաղորդականությունը: Սա հնարավոր չէ ԼԱ ՍՄՎ-երում, քանի որ դրանցում p տիպի իրանն ուղղակի բացակայում է:

Մյուս կողմից՝ ԼԱ ՍՄՎ-երը ունեն գրեթե իդեալական ենթաշեմային բացվածք [77], ինչի շնորհիվ իրանի պոտենցիալավորման հնարավորության բացակայությունը էական դեր չի խաղում: Ի տարբերություն ՄԱ ՍՄՎ-երի՝ այս սարքերում իրանի բացակայությունը թույլ է տալիս ԻՍ-երում ապահովել ինտեգրման ավելի բարձր աստիճան և, հետևաբար՝ ավելի մեծ ֆունկցիոնալություն [80]: ՄԱ ՍՄՎ-ն ունի իրեն հատուկ բացասական երևույթ [77], որը հայտնի է լողացող իրանի երևույթ անվամբ [81]: Այն պայմանավորված է հարվածային իոնացման [82] ժամանակ գեներացված լիցքակիրներով, որոնք տեղաբաշխվում են սարքի իրանում և ազդում են շեմային լարման վրա: Լողացող-իրանի երևույթի ազդեցության չափը [81] կախված է նրանից, թե որքան հաճախ է սարքը փոխանջատվել հարվածային իոնացման պայմաններում: Վատագույն դեպքում երկար ժամանակ միացված սարքն անջատվում է և կրկին միացվում, մինչև իրանի լիցքը կհասցնի հավասարակշիռ վիճակի հասնել [77]: Նշված երևույթների պատճառով էապես բարդանում է սխեմաների նախագծման գործընթացը: Իրան-ակունք հաղորդող pn-անցման պատճառով լողացող իրանի երևույթը մեծացնում է նաև տրանզիստորի անջատման հոսանքը [52], ինչը հանգեցնում է էներգասպառման աճին [30]: ԼԱ ՍՄՎ-երում օգտագործվող օքսիդի շերտի բարակությունը մեծ դժվարություններ է առաջացնում այդ սարքերի արտադրության ժամանակ:

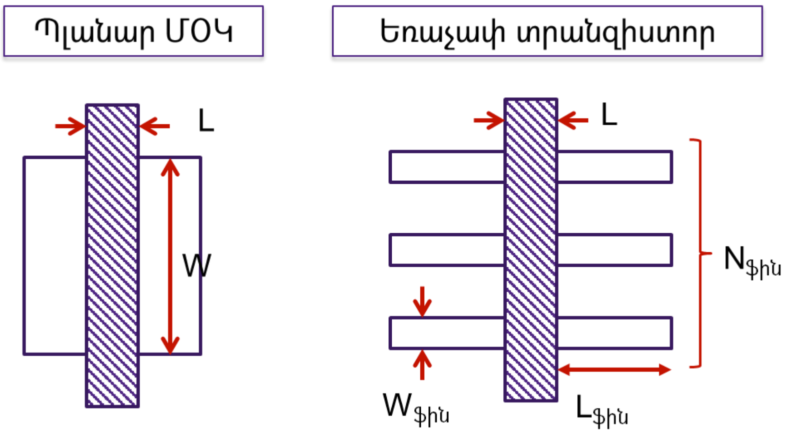
ՄՕԿ տեխնոլոգիայի մյուս այլընտրանքն է եռաչափ բազմափական (ԲՓ) տրանզիստորային տեխնոլոգիան [78,83,84]: Եռաչափ ԲՓ-ի տրանզիստորի (նկ. 1.10) կառուցվածքում փականները շրջապատում են սիլիցիումային բարակ ֆին կոչվող շերտով պատված հոսքուղին: Եռաչափ և ՄՕԿ տրանզիստորների կառուցվածքային տարբերությունը [85] այն է, որ ՄՕԿ պլանար տրանզիստորների հոսքուղին գտնվում է փականի տակ և ղեկավարվում է փականից հարթակ ուղղված էլեկտրական դաշտի միջոցով, իսկ եռաչափ տրանզիստորում (ԵՏ) այն պատված է ֆինով և շրջափակված է փականով [78]:



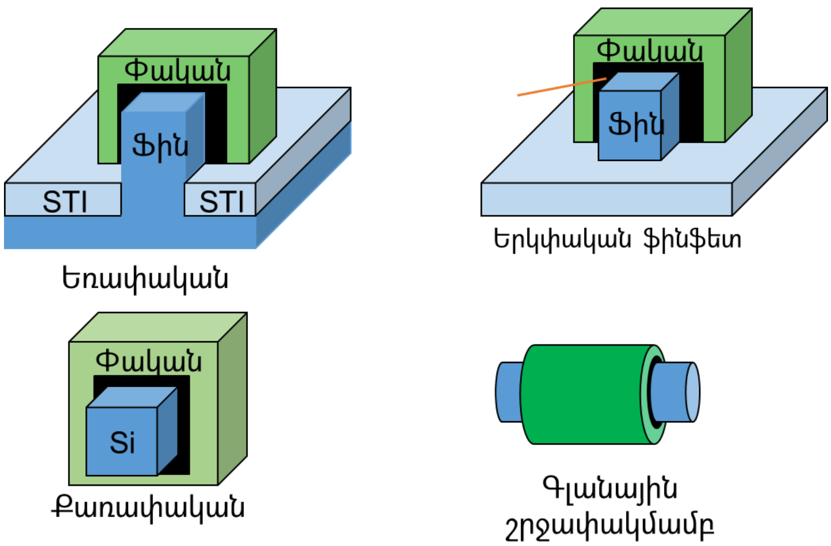
Նկ. 1.10. Եռաչափ տրանզիստորի կառուցվածքը

Ի տարբերություն ՄՕԿ-ի՝ ԵՏ-ում, հոսքուղու երկարությունից բացի, տրանզիստորի բնութագրերի վրա ազդում են ֆին շերտի լայնությունն ու հաստությունը [84]: Չնայած կառուցվածքային որոշակի փոփոխություններին՝ ԵՏ-երի տրանզիստորների տոպոլոգիան (նկ. 1.11) [86] էապես չի տարբերվում ՄՕԿ-ի տոպոլոգիայից, ինչը հեշտացնում է նախագծման մի գործընթացից մյուսին անցումը:

Ըստ ֆին շերտի՝ փականով շրջափակման ձևի՝ ԵՏ-երը լինում են երկփական, եռափական, քառափական և գլանային շրջափակմամբ (նկ. 1.12) [78]: Քառափական և գլանային շրջափակմամբ ԵՏ-երը, իրենց արտադրման բարդությունների պատճառով, լայն տարածում չունեն [78]:

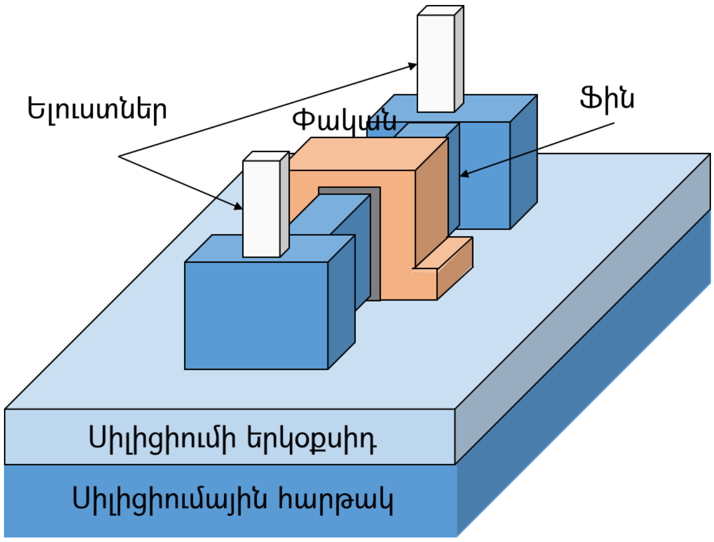


Նկ. 1.11. ՄՕԿ և ԵՏ տոպոլոգիաները

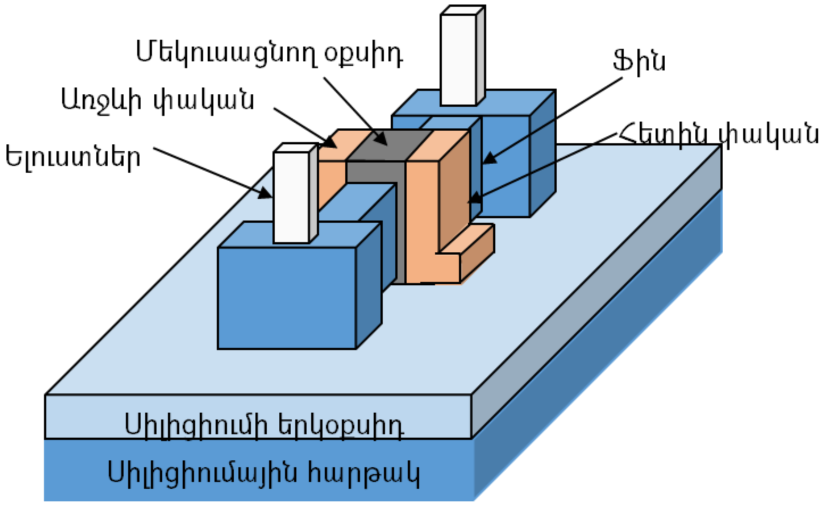


Նկ. 1.12. Բազմափական տրանզիստորների տեսակները

Իրականացման տեսանկյունից հնարավոր է ԲՓ տրանզիստորի փականները ղեկավարել մեկ ընդհանուր միացման միջոցով [87] կամ միմյանցից առանձին՝ փականների միջև ունենալով սիլիցիումի երկօքսիդի մեկուսացնող շերտ [87]: Առաջին տեսակի տրանզիստորը հայտնի է կարճ փականային միացմամբ ԵՏ (ԿՓԵՏ) (նկ. 1.13), իսկ երկրորդը՝ անկախ փականով ԵՏ (ԱՓԵՏ) անվամբ (նկ. 1.14): Արտադրական հավելյալ փուլի պատճառով ԱՓԵՏ-երը դժվար իրագործելի են, սակայն մի շարք կիրառություններում դրանք ընձեռում են նախագծման լրացուցիչ ճկունություն և տոպոլոգիական նոր լուծումների հնարավորություններ:



Նկ. 1.13. ԿՓԵՏ եռաչափ տրանզիստորը



Նկ. 1.14. ԱՓԵՏ եռաչափ տրանզիստորը

Անկախ տեսակից՝ ԵՏ-երի լայնությունը դիսկրետ մեծություն է և որոշվում է հենց ֆիների լայնությամբ ու քանակով (նկ. 1.15) [78]: Երկփական, եռափական և քառափական տրանզիստորների արդյունավետ լայնությունները որոշվում են համապատասխանաբար (1.15)-(1.17)-ով [78].

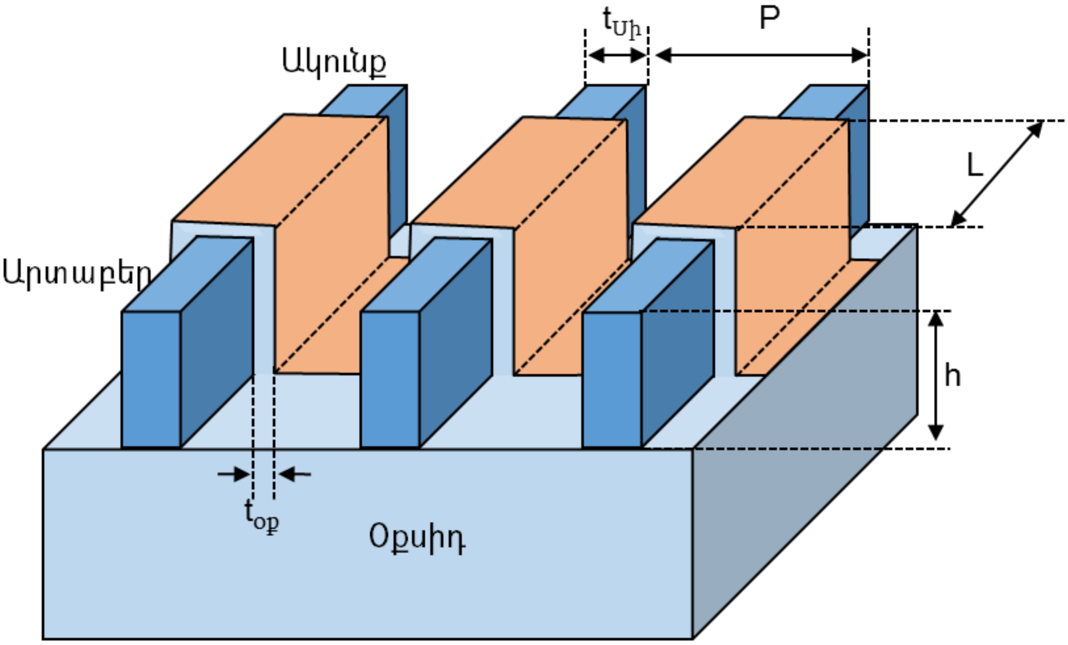
$$W_{արդ} = 2 * H_{ֆին}, \quad (1.15)$$

$$W_{արդ} = 2 * H_{ֆին} + W_{ֆին}, \quad (1.16)$$

$$W_{արդ} = 2 * H_{ֆին} + 2 * W_{ֆին}, \quad (1.17)$$

որտեղ $W_{արդ}$-ը տրանզիստորի արդյունարար լայնությունն է, $H_{ֆին}$-ը՝ ֆինի բարձրությունը, իսկ $W_{ֆին}$-ը՝ ֆինի լայնությունը:

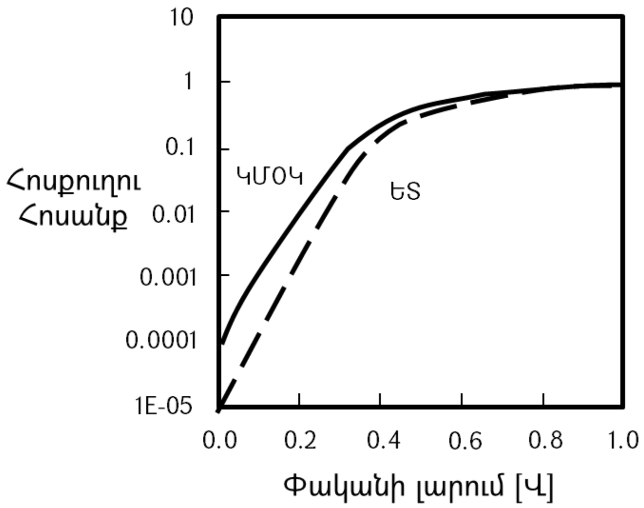
Չափերի դիսկրետացումը բարդացնում է լայնության նկատմամբ զգայուն սխեմաների նախագծումը, քանի որ ԵՏ-երի չափերի փոփոխությունները պետք է լինեն ֆին շերտի լայնության պատիկները [87]



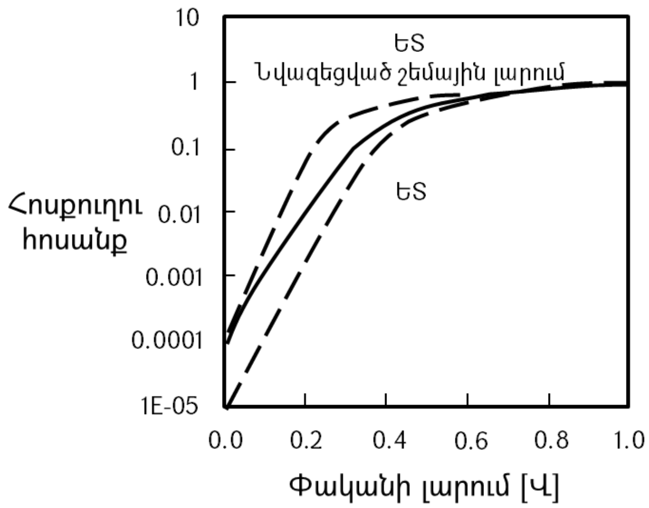
Նկ. 1.15. Բազմաֆին տրանզիստորի տեսքը

ԵՏ-երը ՄՕԿ-երի համեմատ ունեն մի շարք առավելություններ [85], որոնց շնորհիվ 22 *նմ*-ից ցածր տեխնոլոգիաներում դրանք կիրառելիության տեսանկյունից ավելի գրավիչ են: Դրանք են [83]՝

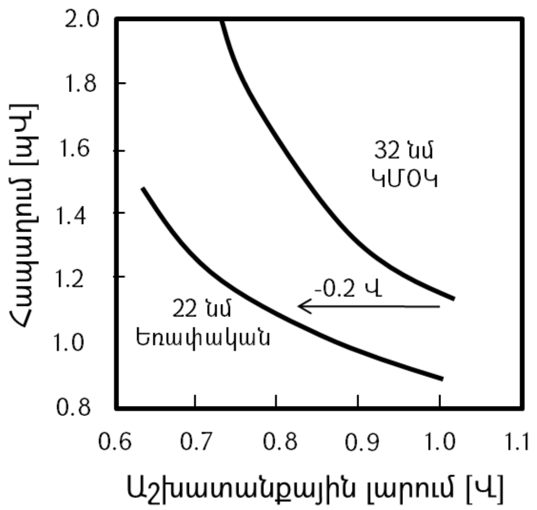
- համեմատաբար փոքր ենթաշեմային հոսանքները (նկ. 1.16), ինչը հետևանք է իրանի միջով կորստային հոսանքների ուղիների վերացման,
- կորստային հոսանքների ավելացման հաշվին հնարավոր է նվազեցնել շեմային լարումը (նկ. 1.17), սնուցման աղբյուրի լարումը և մեծացնել արագագործությունը,
- ԵՏ-երով տրամաբանական փականների նախագծման ժամանակ կարելի է ստանալ հապաղման և էներգասպառման ԿՄՕԿ-ի համեմատ բարելավված արժեքներ (նկ. 1.18):



Նկ. 1.16. ԿՄՕԿ և եռաչափ տեխնոլոգիաների ենթաշեմային ցրման հոսանքները



Նկ. 1.17. Ցածր շեմային լարումների դեպքում կորստային հոսանքների նվազումը



Նկ. 1.18. ԵՓ և ԿՄՕԿ տեխնոլոգիաների հապաղման՝ սնուցման լարումից ունեցած կախվածությունը

Այսպիսով, հաշվի առնելով ԵՏ-երի՝ ՄՕԿ սարքերի համեմատ ունեցած էական առավելությունները և դրանցով նախագծված ԻՍ-երի ինտեգրման աստիճանի և արագագործության մեծացման ու էներգասպառման նվազեցման իրական հեռանկարը, առաջանում է եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման միջոցների մշակման ծայրահեղ անհրաժեշտություն:

1.2. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման միջոցների մշակման արդի վիճակը և հիմնահարցերը

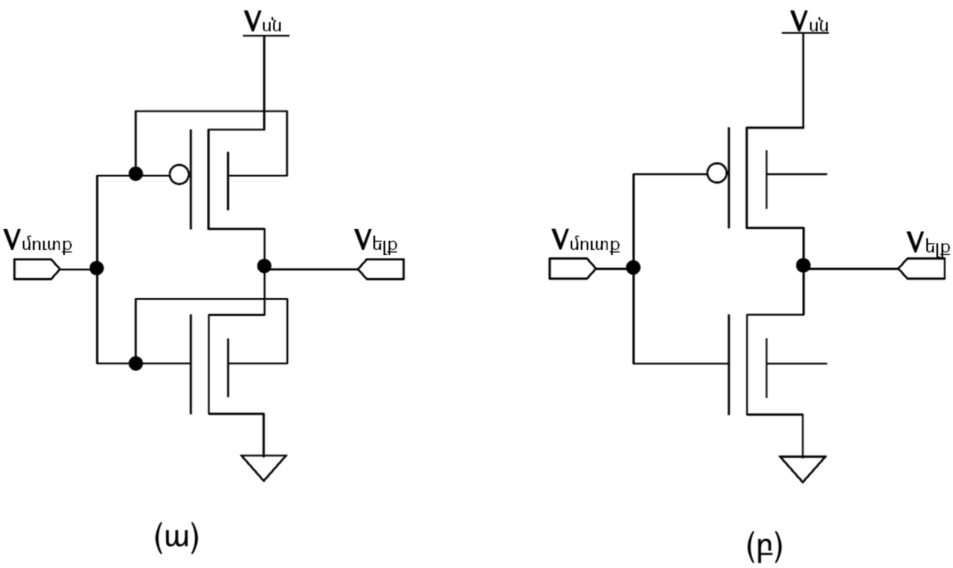
ԵՏ-երի առավելությունների շնորհիվ՝ դրանք արագորեն սկսել են օգտագործվել ԻՍ-երի նախագծման համար: Այդ տրանզիստորների էներգասպառումն ու արագագործությունը մի շարք սխեմաների օրինակներով ուսումնասիրվել են բազմաթիվ աշխատանքներում [88,89,90,91]:

Տրամաբանական փականները (ՏՓ) ցանկացած թվային սխեմայի հիմնական բաղկացուցիչներն են: Սխեմաներում ԵՏ-երը կարող են ունենալ հետևյալ կոնֆիգուրացիաները [87].

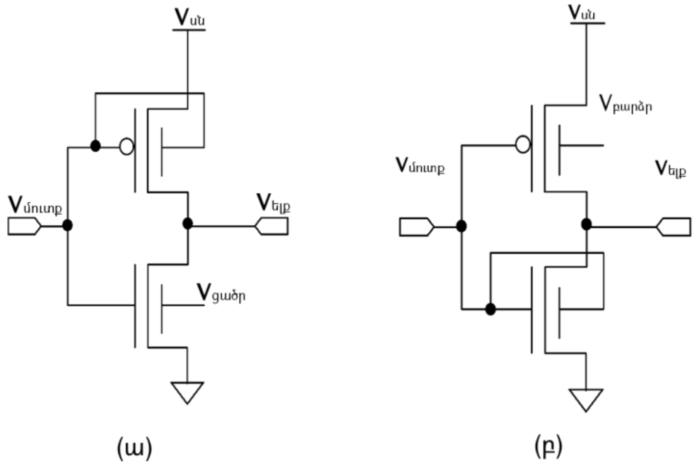
- ԿՓԵՏ միացում, որում տրանզիստորի երկու փականները ղեկավարվում են մեկ ընդհանուր միացմամբ: Այս դեպքում բարելավվում են սխեմայի տանող ուժը և հոսքուղու ղեկավարելիությունը: ԿՓԵՏ միացմամբ տրանզիստորներով նպատակահարմար է նախագծել բարձր արագագործություն պահանջող սխեմաներ:
- ԱՓԵՏ միացում, որի դեպքում տրանզիստորի փականները ղեկավարվում են միմյանցից անկախ՝ տարբեր պոտենցիալներով: Այս ԵՏ-երի օգտագործման դեպքում ՏՓ-ի մակերեսը կարելի է նվազեցնել՝ փոքրացնելով տրանզիստորների քանակը:
- Ցածր-էներգասպառմամբ միացում (ՑԷՄ), որի դեպքում n տիպի տրանզիստորի հետին փականի վրա կիրառվում է ցածր լարում, իսկ p տիպի սարքի հետին փականին՝ բարձր լարում: Այս միացման միջոցով հնարավոր է դառնում ղեկավարել տրանզիստորների ՇԼ-երը, և ենթաշեմային հոսանքների նվազեցման միջոցով փոքրացնել էներգասպառումը:

ԵՏ-ի նշված միացումներն օգտագործելով՝ նախագծվել են [87] շրջիչի (նկ. 1.19-1.20) և ԵՎ-ՈՉ տարրի (նկ. 1.21-1.23) սխեմաները: ՑԷՄ միացված շրջիչի սխեմայում p տիպի տրանզիստորի հետին փականին տրվել է 1 *Վ*, իսկ n տիպին՝ -0.2 *Վ* լարում [87]: Այս եղանակով տրանզիստորների շեմային լարումների ղեկավարելիության շնորհիվ շրջիչի սխեման կարող է նախագծվել ցածր էներգասպառում պահանջող կիրառությունների համար: Շրջիչի՝ անկախ փականներով միացմամբ սխեմաներից մեկում (նկ. 1.20ա) [87] p տիպի տրանզիստորի փականները միացված են իրար, n տիպի սարքի հետին փականը միացված է -0.2 *Վ* լարմանը, իսկ առջևինը՝ շրջիչի մուտքին: ԱՓԵՏ մյուս սխեմայում (նկ. 1.20բ) [87] n տրանզիստորի փականները միացված են իրար, p տիպի սարքի առջևի փականը միացված է շրջիչի մուտքին, իսկ հետինը՝ 1 Վ լարմանը: Վերջին երկու սխեմաներն ընդունված է անվանել ԱՓԵՏ/ՑԷՄ հիբրիդային միացմամբ կառուցվածքներ [87]:

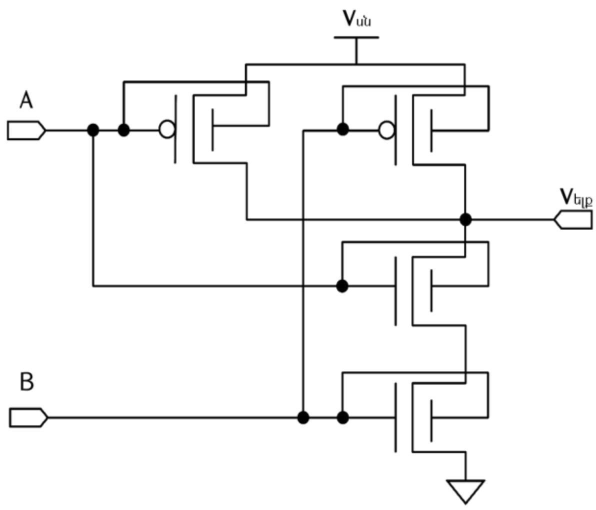
ԵՎ-ՈՉ սխեմաներն ունեն համանման կառուցվածք: ԿՓԵՏ միացման դեպքում (նկ. 1.21) [87] ՏՓ-ում տրանզիստորների երկու փականները միացված են իրար: ԵՎ-ՈՉ ՏՓ-ի համար ՑԷՄ միացման յուրաքանչյուր p տրանզիստորի հետին փականը միացված է տրամաբանական բարձր մակարդակին, իսկ n տրանզիստորների հետին փականները՝ տրամաբանական ցածր մակարդակին:



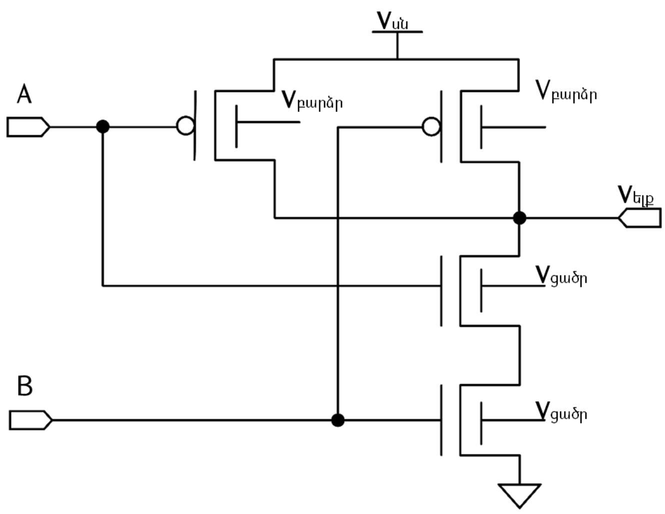
Նկ. 1.19. ԿՓԵՏ և ԱՓԵՏ միացմամբ շրջիչները



Նկ. 1.20. ԱՓԵՏ/ՑԷՄ հիբրիդային միացմամբ շրջիչները

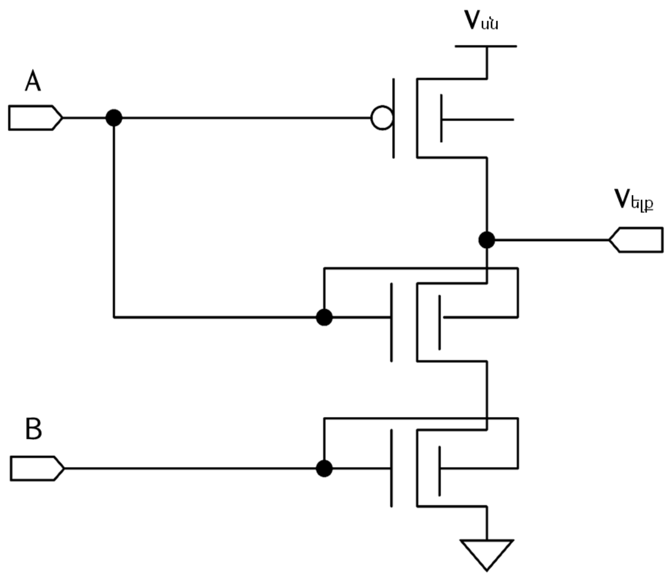


Նկ. 1.21. ԿՓԵՏ միացմամբ ԵՎ-ՈՉ տարրը



Նկ. 1.22. ՑԷՄ միացմամբ ԵՎ-ՈՉ տարրը

ԵՎ-ՈՉ կառուցվածքը փոխվում է, երբ այն իրականացվում է ԱՓԵՏ տրանզիստորներով: p տրանզիստորի հետին փականը միացվում է ԵՎ-ՈՉ-ի մի մուտքին, իսկ դիմացինը՝ մյուսին: Երկու n տրանզիստորների փականները միացվում են իրար: Տրանզիստորների՝ երկու փական ունենալու շնորհիվ՝ այս սխեման չորսի փոխարեն իրագործվում է 3 տրանզիստորով և, հետևաբար, զբաղեցնում է ավելի փոքր մակերես [87]:



Նկ. 1.23. ԱՓԵՏ/ՑԷՄ հիբրիդային միացմամբ ԵՎ-ՈՉ տարրը

ԱՓԵՏ/ՑԷՄ հիբրիդային միացման դեպքում սխեման նման է ԱՓԵՏ սխեմային, այն տարբերությամբ, որ երկու n տրանզիստորների առաջնային փականները միացվում են ՏՓ-ի մուտքերին, իսկ հետին փականները՝ տրամաբանական ցածր լարմանը:

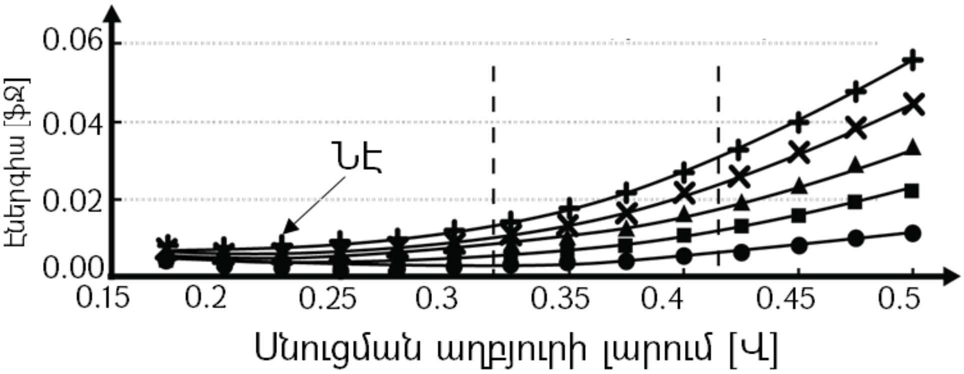
Արագագործությունն ու էներգասպառումը միշտ եղել են ԻՍ-երի աշխատանքի արդյունավետության չափանիշներ [30]: Մի շարք կիրառություններում, որոնք չեն պահանջում բարձր արագագործություն, սնուցման լարման մասշտաբավորումը թույլ է տվել ստանալ էներգասպառման անհրաժեշտ ցածր մակարդակ [88]: Հայտնի է նաև [88,89], որ սնուցման լարման արժեքը, որի դեպքում սխեմայի էներգասպառումը հասնում է հնարավոր նվազագույն արժեքի, հանդիսանում է սխեմայում օգտագործված տրանզիստորների ՇԼ-ին մոտ մեծություն է: Ուսումնասիրությունները ցույց են տվել, որ ԵՏ-երը ենթանանոմետրական տիրույթում մասշտաբավորվող, համեմատաբար ցածր փականային ցրվող հոսանքներ ունեցող և կարճ-հոսքուղային երևույթների լավ կառավարելիությամբ նախագծման միջոցներ են [78]: Մի շարք աշխատանքներում [90-

92] ուսումնասիրվել են ԵՏ-երով նախագծված թվային ստանդարտ տարրերի գրադարաններ (ԹՍՏԳ), ստատիկ հիշողություններ և այլ սխեմաներ:

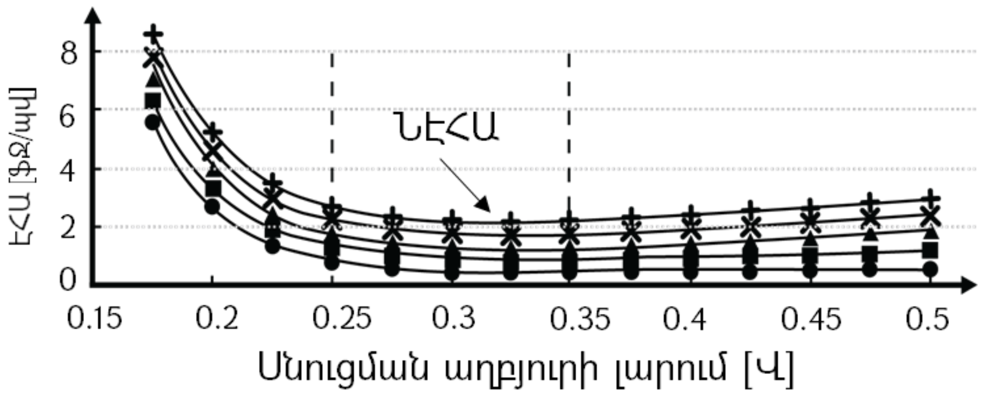
ԹՍՏԳ-երում առկա տարրերի արագագործությանը և էներգասպառմանը ներկայացվող պահանջները բավարարելու նպատակով [90] արվում է տրանզիստորների օպտիմալ չափերի ընտրություն, որոնց դեպքում տարրի ելքում ստացվում են դրական և բացասական ճակատների քվազիհավասար արժեքներ: Ի տարբերություն ԿՄՕԿ տեխնոլոգիաների, որտեղ տրանզիստորների չափերը անընդհատ մեծություններ են, ԵՏ-երի չափերի ընտրությունը բերվում է դիսկտրետ արժեքների [90]՝ տրանզիստորի ֆիների քանակի ընտրության: Չափերի դիսկրետացումը էականպես հեշտացնում է նախագծման գործընթացը, բացառությամբ այն սխեմաների, որոնք խիստ զգայուն են չափերի փոփոխության նկատմամբ: Դիտարկվում է [90] 5 *նմ* հոսքուղու երկարությամբ տրանզիստորների սնուցման լարման երկու տարբերակ՝ տրանզիստորների շեմային լարմանը մոտ արժեք և շեմային լարումից էապես մեծ արժեք: ԹՍՏԳ-երում տարրերի տանող ուժերի բազմազանությունն ու արագագործությունն ապահովելու նպատակով դրանց նախագծումն իրականցվել է ԿՓԵՏ տրանզիստորներով [90]:

Սնուցման լարման՝ ՇԼ-ին մոտ արժեքների դեպքում սխեմայի արագագործության նվազեցման հաշվին տեղի է ունենում էներգասպառման նվազեցում [90]: Դա պայմանավորված է նրանով, որ սնուցման լարման նվազեցման դեպքում դինամիկ էներգասպառումը նվազում է քառակուսային օրենքով (նկ. 1.24)[90]: Սակայն, սխեմայի հապաղման մեծացման պատճառով, կորստի էներգասպառումը՝ կորստի էներգիայի և հապաղման արտադրյալը (ԷՀԱ), աճում է (նկ. 1.25) [90]: Ստացված արդյունքները [90] ցույց են տալիս, որ ԷՀԱ-ն ավելի փոքր արժեք ունի համեմատաբար մեծ շեմային լարումների դեպքում:

Նշված պատճառով ԹՍՏԳ-ի միջոցով բարձր արագագործություն և ցածր էներգասպառում պահանջող ԻՍ-եր նախագծելու համար անհրաժեշտ է ԹՍՏԳ-ի տարրերը նախագծել երկու տարբեր սնուցման լարումների համար [90]: Օգտագործվել են [90] կիցշեմային 0.3 *Վ* և գերշեմային՝ 0.45 *Վ* սնուցման լարումներ: Հաշվի առնելով գրադարանի տարրերի բնութագրերի վրա դրանց ունեցած ազդեցությունը՝ տրանզիստորների չափերի ճիշտ ընտրությունը դառնում է խիստ կարևոր:



Նկ. 1.24. Նոմինալ շեմային լարման դեպքում շրջիչների շղթայի էներգասպառման՝ սնուցման լարումից կախվածությունը



Նկ. 1.25. Նոմինալ շեմային լարման դեպքում շրջիչների շղթայի ԷՀԱ-ի՝ սնուցման լարումից կախվածությունը

Շրջիչի ելքում քվազիհավասար դրական և բացասական ճակատներ ստանալու համար դիտարկվում է վերջինիս լավարկումը [91]: ԵՏ-ի արտաբեր-ակունք հոսանքը տրվում է հետևյալ արտահայտությամբ [91]՝

$$I_{\text{աա}} = I_0 e^{\frac{\left(V_{\text{փա}}+\lambda V_{\text{աա}}-V_{\text{շեմ}}\right)-a\left(V_{\text{փա}}+\lambda V_{\text{աա}}-V_{\text{շեմ}}\right)^2}{m v_T}} \left(1-e^{\frac{-V_{\text{աա}}}{v_T}}\right), \tag{1.18}$$

որտեղ I_0-ն հագեցման հոսանքն է:

Հավասար ճակատներ ստանալու համար անհրաժեշտ է, որ p տիպի ԵՏ-ի ֆիների քանակը լինի [91]՝

$$N_P = N_N \frac{I_{\text{աա},N}}{I_{\text{աա},P}}: \quad (1.19)$$

HSPICE նմանակումների արդյունքում 0.3 *Վ* սնուցման լարմամբ 1,2,4 և 8 տանող ուժերով շրջիչների Np/Nn-ը (1.19)-ից համապատասխանաբար հավասար են 1/1, 2/2, 4/4, 8/7 [91]: Նմանակումների միջոցով որոշվել են նաև ԹՍՏԳ-երի մյուս տարրերի տրանզիստորների օպտիմալ չափերը [91]: Մշակված գրադարանի տարրերի՝ լարման և հաճախականության դինամիկ մասշտաբավորման համար նախատեսված լինելու պատճառով կարելի է դրանք առանց չափերի փոփոխության սնուցել նաև գերշեմային 0.45 *Վ* լարմամբ [91]:

Մշակված ԹՍՏԳ-ն [92] օգտագործվել է համակցական ու հաջորդական սխեմաներ և LEON2 SPARC պրոցեսորը սինթեզելու նպատակով: Սինթեզված սխեմաների փականային մակարդակի նմանակման արդյունքում ստացված տվյալները համեմատվել են նույն սխեմաները 16 *նմ* և 45 *նմ* ԿՄՕԿ տեխնոլոգիաներով սինթեզման արդյունքների հետ [92]: Համեմատությունը ցույց է տվել, որ 0.45 *Վ* սնուցման լարմամբ ԵՏ-ով ստացված սխեման 8-14 անգամ ավելի արագագործ է քան 0.7 *Վ* սնուցման լարմամբ և 16 *նմ* ԿՄՕԿ սխեման, և 25-41 անգամ ավելի արագագործ է, քան 1.1 *Վ* սնուցման լարմամբ 45 *նմ* ԿՄՕԿ սխեման [92]: Էներգասպառման տեսանկյունից, նոմինալ պայմաններում, 5 *նմ* ԵՏ տեխնոլոգիայով սինթեզված սխեման 16 *նմ* ԿՄՕԿ սխեմայից 30-80 անգամ ավելի էներգախնայող է: 45 *նմ* ԿՄՕԿ սխեմայի հետ համեմատությունը ցույց է տվել էներգասպառման 660 – 1050 անգամ բարելավում [92]:

Եռաչափ տրանզիստորների կարևոր առավելություն է $I_{\text{հագ}}/I_{\text{անջ}}$ մեծ հարաբերությունը, որը հանգեցնում է շատ ավելի մեծ դինամիկ էներգասպառում / կորստային էներգասպառում հարաբերության, քան ապահովվում էր ԿՄՕԿ տեխնոլոգիայով [92]: Անվանական և ցածր էներգասպառման տարբերակների բնութագրերի համեմատությունը ցույց է տվել, որ 0.45 *Վ*-ից սնուցման լարումը 0.3 *Վ* նվազեցնելու դեպքում հապաղման՝ 3 անգամ ավելացման հաշվին, էներգասպառումը նվազում է 6 անգամ [92]: Հաշվի առնելով, որ 16 *նմ* ԿՄՕԿ սխեմաներում սնուցման լարումը 0.9 *Վ*-ից 0.7 *Վ* դարձնելիս էներգասպառումը 2,3 անգամ բարելավվում է հապաղման 3 անգամ վատթարացման հաշվին, ակնհայտ է դառնում, որ ԵՏ տեխնոլոգիայում ԷՀԱ-ն էապես նվազել է [92]: Կատարված հետազոտությունները [92] ցույց են տվել, որ 16 նմ ԿՄՕԿ հաջորդական սխեմաների հետ համեմատած, ԵՏ սխեմաներում սինքրոազդանշանի սպառած հզորությունը նվազել է 98 անգամ:

ԹՍՏԳ-երի տարրերում տրանզիստորների չափերի ընտրության լավարկման մի քանի մեթոդներ կան [93]: Դրանց բոլորի նպատակն է ապահովել սխեմայի արագագործության և էներգասպառման օպտիմալ արժեքներ: Քննվել են [93] գործընթացի ոչ միօրինակությունների, սնուցման լարման և ջերմաստիճանի (ՍԼՋ) տատանումների ազդեցությունները տարբեր մեթոդներով լավարկված ԹՍՏԳ տարրերի բնութագրերի վրա: Դիտարկվել են սխեմաների լավարկման հետևյալ մեթոդները [93]՝ տրամաբանական ջանքի մեթոդ (ՏՋ), տրանզիստորների օպտիմալացված լավարկմնան (SOL) մեթոդ և նվազագույն չափերով լավարկման մեթոդ (ՆՉԼ): ՏՋ լավարկումը հիմնված է պարզ հապաղման մոդելի վրա, որի դեպքում տարբեր ՏՓ-երի չափերը որոշվում են դրանց ելքին 4 նվազագույն չափերով շրջիչներ միացնելով [93]: SOL մեթոդի դեպքում երկրաչափական չափերի օպտիմալացումը կատարվում է հապաղման և մակերեսի նվազագույն արժեքները գտնելու միջոցով [93]: Փորձերի համար սահմանվել են հետևյալ անվանական արժեքները՝ 25^0 ջերմաստիճան, 0.8 *Վ* սնուցման լարում: Արդյունքները ցույց են տվել, որ ՏՋ մեթոդով լավարկված սխեմաների հապաղումները 57%-ով փոքր են ՆՉԼ մեթոդով լավարկված սխեմաների հապաղումներից, սակայն դրանց էներգասպառումը 38%-ով ավելին է վերջիններից [93]: Դրա պատճառն այն է, որ ՏՋ մեթոդի կիրառման դեպքում որոշ տրանզիստորների չափերն ավելի մեծ են, քան ՆՉԼ մեթոդի դեպքում, ինչը հանգեցնում է ավելի մեծ էներգասպառման [93]: Սակայն նշված երկու մեթոդների դեպքում ԷՀԱ-երը գրեթե հավասար արժեքներ ունեն [93]: SOL եղանակով լավարկումը տալիս է սխեմաների հապաղման և էներգասպառման ավելի մեծ արժեքներ: Հաշվի առնելով լավարկման տարբեր մեթոդների կիրառման դեպքում այդ բնութագրերի տարբեր արժեքները՝ կատարվել է [93] դրանցով լավարկված սխեմաների բնութագրերի վրա ՍԼՋ տատանումների ազդեցությունների գնահատում:

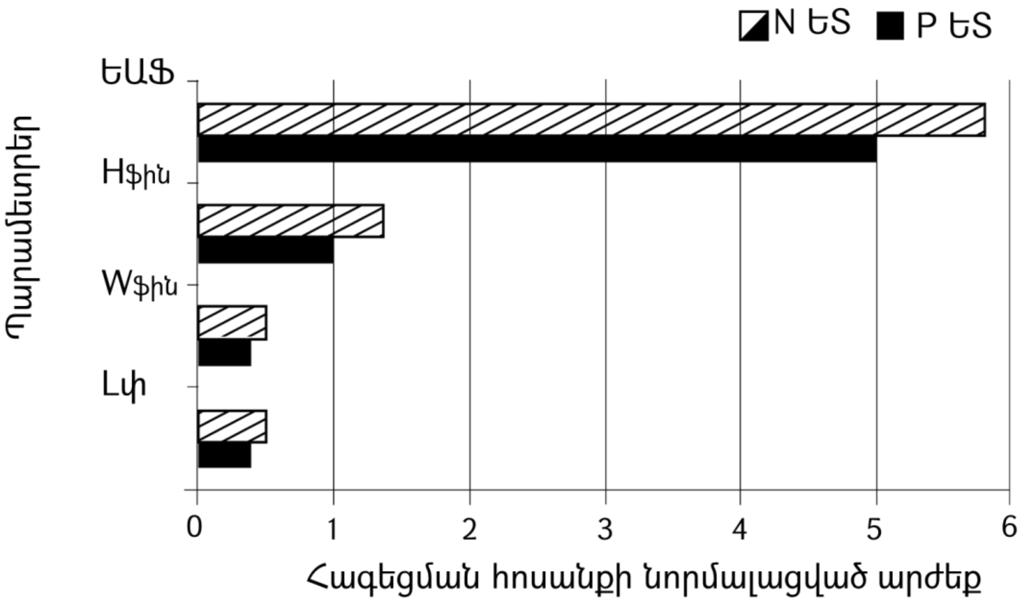
Բնութագրերի փոփոխականությունը կարելի է դասակարգել հետևյալ տեսակների՝ միջավայրի գործոններ և ֆիզիկական գործոններ [93]: Ֆիզիկական գործոններին է դասվում գործընթացի փոփոխականությունը, որը պայմանավորված է լիտոգրաֆիայի գործընթացի անկատարություններով: Մի շարք աշխատանքներում ցույց է տրվել, որ գործընթացի փոփոխականությունների հետևանքով ամենամեծ ազդեցությունների ենթարկվել է տրանզիստորներում լիցքակիրների ելքի աշխատանքը (նկ. 1.26) [93]:

Ելքի աշխատանքների ֆլուկտուացիաները (ԵԱՖ) [93] հիմնականում պայմանավորված են արտադրական գործընթացում տարբեր ելքի աշխատանքներ ունեցող մետաղների օգտագործմամբ: ԵՏ-երի մեկ այլ թերությունն այն է, որ n տիպի տրանզիստորը ավելի զգայուն է գործընթացային անկատարությունների նկատմամբ, քան p տիպը:

Ջերմաստիճանի և լարման փոփոխականությունները դասվում են միջավայրային գործոնների շարքը [93]: ԹՍՏԳ-երի մշակելիս դիտարկվում են էներգասպառման և արագագործության՝ սնուցման լարման մինչև 10% տատանումների նկատմամբ զգայնությունը [93]: Այսինքն՝ դիտարկվում է սնուցման լարման 0.72-0.88 *Վ* միջակայքը: Ջերմաստիճանի՝ նոմինալ աշխատանքային անվանական արժեքի շուրջ տատանումները դիտարկվում են -25-ից 125 միջակայքում [93]: Տրանզիստորների չափերի ընտրության նկարագրված երեք դեպքերում գործընթացի փոփոխականության ազդեցությունը էներգասպառման վրա գրեթե միանման է [93]: ԷՀԱ-ի վերլուծությունը ցույց է տվել, որ ՏՋ լավարկմամբ սխեմաները ամենազգայունն են գործընթացի փոփոխությունների նկատմամբ [93]: SOL մեթոդով լավարկված սխեմաները ցուցաբերել են առավելագույն կայունություն [93]:

ԵԱՖ-ը ամենամեծ ազդեցությունը թողնում է սխեմայի ժամանակային բնութագրերի վրա [93]:

ՏՋ մեթոդով լավարկված OAI211 և OAI22 սխեմաների հապաղումները անվանական արժեքներից շեղվել են շուրջ 150%-ով [93]: NAND2 բջջի ԷՀԱ-ն ՆՉԼ, ՏՋ և SOL մեթոդներով լավարկման դեպքում ցուցաբերել է անվանական արժեքից համապատասխանաբար 19.8%, 22.6% 19.4% շեղումներ [93]: ԵԱՖ-ի նկատմամբ ամենամեծ զգայնություն ունեն ՏՋ սխեմաները, իսկ կայունություն՝ SOL սխեմաները [93]:



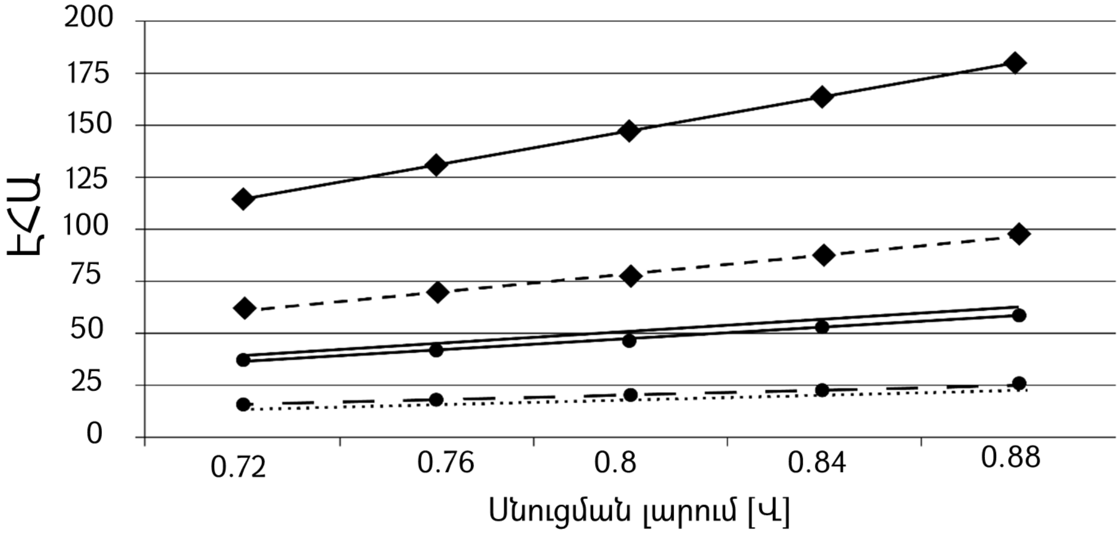
Նկ. 1.26. Հագեցման հոսանքի՝ տեխնոլոգիական շեղումներից կախվածությունը

ՍԼ տատանումների ազդեցությունը զգալի է ԷՀԱ-ի վրա: ՏՋ և ՆՉԼ դեպքերում վատագույն արդյունքները գրեթե իրար հավասար են, մինչդեռ SOL դեպքում ԷՀԱ-ը մոտ երկու անգամ մեծ թիվ է (նկ. 1.27) [93]: Երբ ՍԼ արժեքները անվանական արժեքից մեծ են, ԷՀԱ-ը մինչև 27.6% շեղվում է իր արժեքից, իսկ երբ փոքր են՝ շեղումը 22.1 % է [93]:

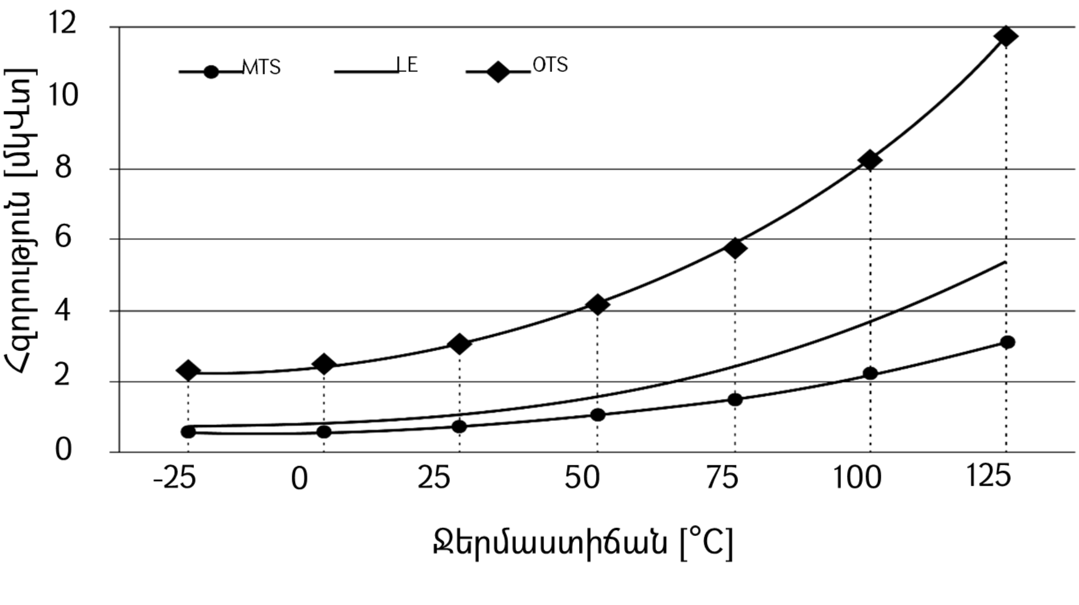
Ջերմաստիճանի տատանումները չնչին ազդեցություն են թողնում բջիջների ժամանակային բնութագրերի վրա և էական ազդեցություն՝ էներգասպառման վրա (նկ. 1.28) [93]:

ՏՋ և ՆՉԼ մոտեցումներով լավարկման դեպքում սխեմաները կայունություն են դրսևորել ջերմաստիճանի՝ անվանական արժեքից ցածր տատանումների դեպքում, սակայն 100-125 ջերմաստիճանների դեպքում էներգասպառումն անվանական արժեքը գերազանցում է 3-4 անգամ [93]: AOI22 բջջի ՆՉԼ, ՏՋ և SOL մոտեցումների դեպքում ստացվում են համապատասխանաբար 77%, 85% և 76% շեղումներ:

Հաշվի առնելով [93]-ում ստացված արդյունքները՝ սխեմայի աշխատանքային միջավայրից կախված կարելի է ընտրել տրանզիստորների չափերի որոշման օպտիմալ եղանակ:



Նկ. 1.27. ԷՀԱ-ի՝ սնուցման լարումից կախվածությունը



Նկ. 1.28. Հզորության՝ ջերմաստիճանից կախվածությունը

Ժամանակակից համակարգային ԻՍ-երի մակերեսի հիմնական մասը զբաղեցնում են ստատիկ հիշող (ՍՀ) սարքերը [94]: ՍՀ սարքերի տարրական բաղկացուցիչները՝ բջիջները (ՍՀԲ), զգայուն են տեխնոլոգիական անկատարությունների և փոփոխականությունների նկատմամբ, ինչը դրանց նախագծման գործընթացում առաջացնում է մի շարք բարդություններ [94]: Ստատիկ հիշողության մատրիցներ նախագծելիս անհրաժեշտ է ճշգրտորեն պլանավորել սխեմայի արագագործության, էներգասպառման, մակերեսի և կայունության միջև փոխզիջումային լուծումները: Սովորաբար ստատիկ ռեժիմում կորստային էներգասպառումը նվազեցնելու համար օգտագործում են բարձր շեմային լարում ունեցող տրանզիստորներ, և որոշ դեպքերում մեծացվում են կիսահաղորդչային սարքի չափերը [94]: Չափերի մեծացումն իր հերթին հանգեցնում է սխեմայի արագագործության նվազմանը [94]: Նախագծման ժամանակ որոշվում են նաև սխեմայի աշխատանքին անհրաժեշտ օպտիմալ սնուցման լարումները: Ցրվող հզորությունը նվազեցնելու համար սահմանվում է նվազագույն սնուցման լարում հասկացությունը, ըստ որի սխեման բավարարում է ֆունկցիոնալության պահանջները [93]:

ՍՀԲ-երի սխեմաները կառուցվել են ԵՏ-երի հետևյալ կառուցվածքներով [93].

- Երկու փականների կարճ միացում, որի դեպքում վերջիններս ղեկավարվում են նույն պոտենցիալով՝ ապահովելով սխեմայի արագագործությունը:
- Փականների անկախ ղեկավարմամբ միացում: Երկու փականներին տրվում է առանձնացված լարումներ: Առջևի փականն օգտագործվում է ԵՏ-ի միացման և անջատման, իսկ հետին փականը՝ սարքի շեմային լարման փոփոխության համար:

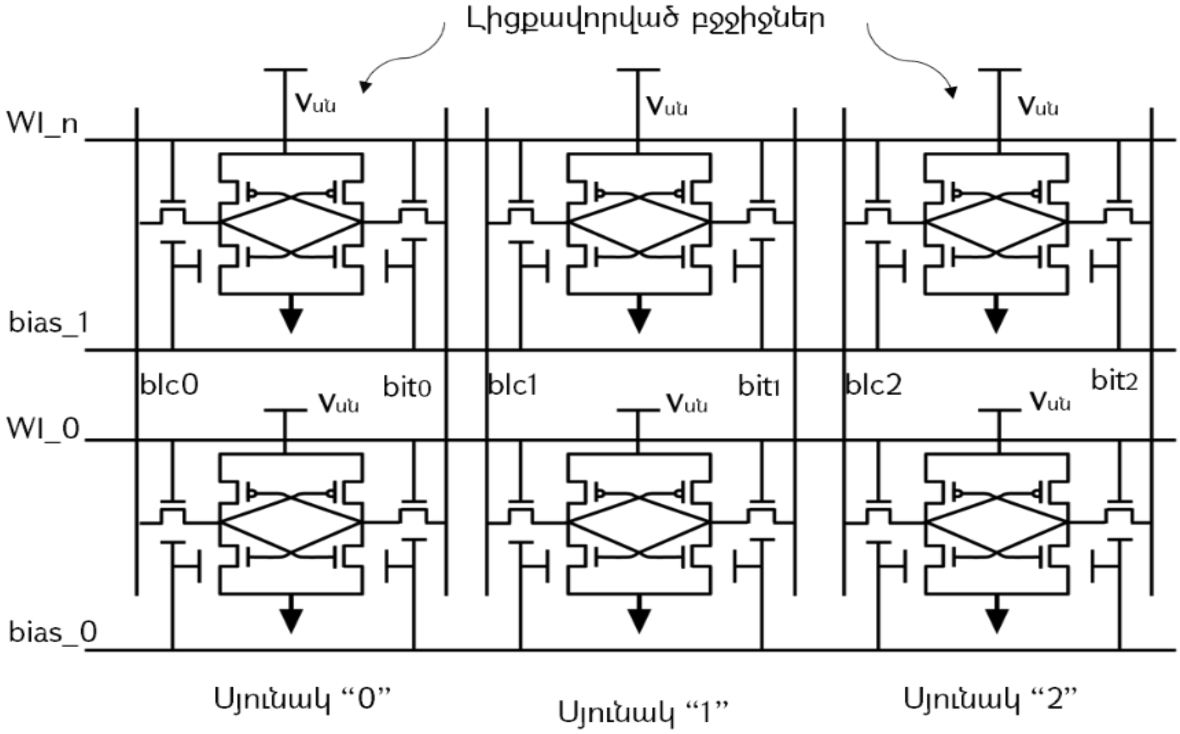
ԵՏ-երի կիրառման արդյունավետությունը ՍՀԲ-երում գնահատելու համար կատարվել է 90 նմ ԼԱ ՍՄՎ ՍՀԲ-ի և ԵՏ ՍՀԲ-ի համեմատություն [94]: Լայնության քվանտացումը սահմանափակում է ՍՀԲ-երի չափերի ընտրությունը, բայց, միևնույն ժամանակ, ֆինի բարձրության մեծացման հաշվին կարելի է ապահովել մեծ դինամիկ հոսանք և հետևաբար՝ արագագործություն [94]: ՍՀԲ-երում տրանզիստորների օպտիմալ չափերի ընտրությունը նվազեցնում է կորստի էներգիան [94]: Շեմային լարման և ֆինի բարձրության մեծացումը բարելավում է սխեմայի կայունությունը և էքսպոնենցիալ նվազեցնում է ակունք-արտաբեր կորուստը [94]: Նշվածը հաշվի առնելով՝ շատ կարևոր է ուսումնասիրել քվանտացման երևույթի ազդեցությունը արագագործության վրա [94]:

Մասնավորապես, ընթերցման գործողության հապաղումը, ԵՏ-ի լրիվ աղքատացած տիրույթի և pn-անցման ունակության արհամարհելի փոքր արժեքի շնորհիվ, զգալիորեն բարելավվում է [94]: Փոքր սնուցման լարումների դեպքում ԵՏ-ի արդյունավետությունը կազմում է 12-13% [94]: ԵՏ ՍՀԲ-երի գրանցման գործողության հապաղումներն ավելի մեծ են, քան շեմային անվանական լարմամբ ՍՄՎ տրանզիստորներով սխեմայինը [94]: Քվանտայնացման պատճառով p տիպի տրանզիստորների տանող ուժը մեծանում է՝ դժվարացնելով գրանցման գործողությունը: Այս դեպքում հապաղումը 8%-ով մեծ է նոմինալ ՇԼ-ով ՍՄՎ-երից և 12%-փոքր՝ բարձր ՇԼ-ով դեպքից [94]:

Քննարկվել է նշված երկու տեխնոլոգիաների դեպքում սխեմայի էներգասպառումը [94]: Փորձերի արդյունքում պարզվել է, որ ԵՏ սխեման 4.5 անգամ փոքր էներգասպառում է ապահովում, քան ՄԱ ՍՄՎ սխեման [94]: ԵՏ-ի լրիվ աղքատացած լինելու շնորհիվ՝ բացառվում է շեմային լարումների փոփոխականությունը, ինչն առկա է ՄԱ ՍՄՎ տրանզիստորների դեպքում [94]: Հայտնի է [77], որ շեմային անվանական լարմամբ ՄԱ ՍՄՎ տրանզիստորների կորստի էներգասպառումը սնուցման լարման աճին զուգընթաց աճում է էքսպոնենցիալ օրենքով, մինչդեռ ԵՏ-ի դեպքում կախվածությունը գծային է [78]: 0.8 *Վ* սնուցման լարման դեպքում շեմային անվանական լարմամբ ՄԱ ՍՄՎ տրանզիստորներով սխեմայի էներգասպառումը 3 անգամ մեծ է ԵՏ սխեմայի էներգասպառման արժեքից [94]:

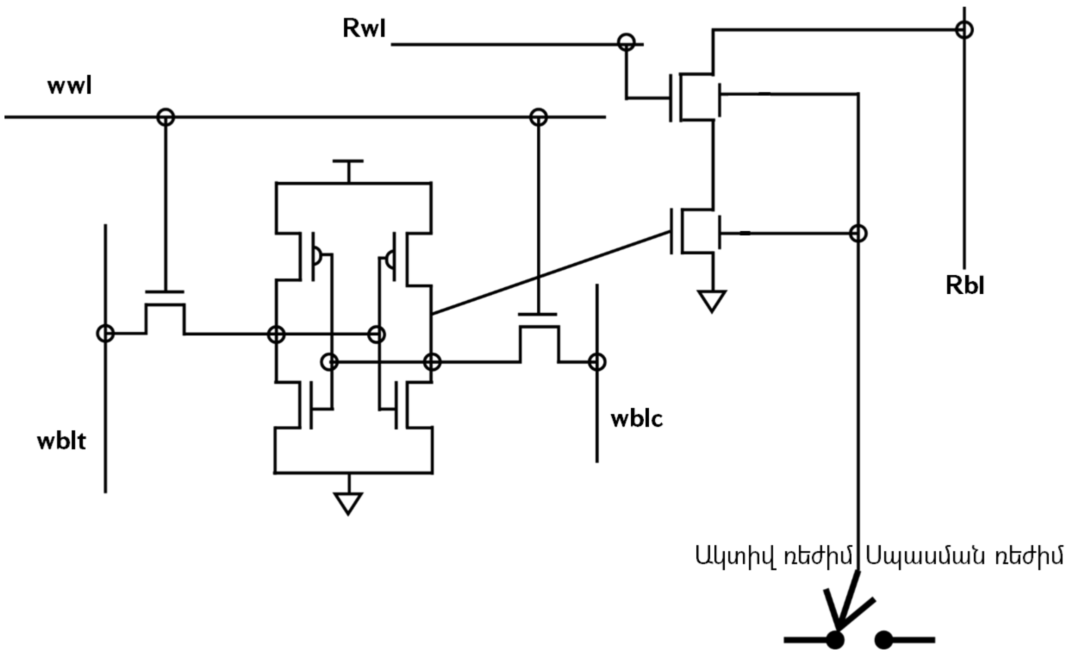
ՍՀԲ-ում գործողությունների կայունությունը պայմանավորված է սարքերի շեմային լարումների, երկարությունների և լայնությունների փոփոխականությամբ [94]:

Մշակվել է [94] բարձր արագագործությամբ, ցածր կորստային էներգիայով և բարձր կայունությամբ ՍՀ մատրից (ՍՀՄ)՝ կիրառելով ՍՀԲ-երի տողի անկախ փականներով ոչ համաչափ ԵՏ-ի հետին փականի ընդհանուր շեղմամբ մեթոդը: Սնուցման լարումից բարձր կամ 0-ից ցածր պոտենցիալներով շեղումը նպաստում է ընթերցման և գրանցման գործողությունների արագագործությանը, նվազեցնում է ստատիկ կորուստները և գործընթացի փոփոխություններից կախվածությունը [94]: Այս մեթոդի թերություններից է հետին փականի լրացուցիչ միացման պատճառով մակերեսի ավելացումը [94]: Առջևի փականի շեմային լարումը ղեկավարվում է հետին փականի շեղման պոտենցիալով: Նշված մեթոդի հիմնական առավելություններն են միջմիացումների ընդհանուր լինելու հաշվին համեմատաբար փոքր մակերեսը և շեմային լարման լավ ղեկավարելիությունը՝ համեմատած ԿՄՕԿ և ՄԱ ՍՄՎ տեխնոլոգիաների հետ: Մշակված ՍՀՄ-ում (նկ. 1.29) [94] յուրաքանչյուր տողի բջիջների տրանզիստորները միացված են տողի համար ընդհանուր շեղման պոտենցիալին: Այդ պոտենցիալը կիրառվում է բջջի հասանելիության և ցածր մակարդակն ապահովող տրանզիստորների հետին փականներին [94]: Դրա հաշվին բարելավվում են ընթերցման գործողության արագագործությունն ու աղմկակայունությունը և գրանցման գործողության տևողությունը [94]: Նմանակումների արդյունքում մշակված մեթոդը կորստային էներգիան նվազեցրել է 3 անգամ, իսկ ընթերցման արագագործությունը բարելավել է 20-40%-ով՝ համեմատած ԿՓԵՏ միացմամբ սխեմաներին [94]: Նկարագրված նախագծման մեթոդը կարելի է կիրառել նաև բազմամուտք ՍՀԲ-երի նախագծման համար (նկ. 1.30): Այս դեպքում ևս գրանցվում է ընթերցման գործողության արագագործության աճ [94]:

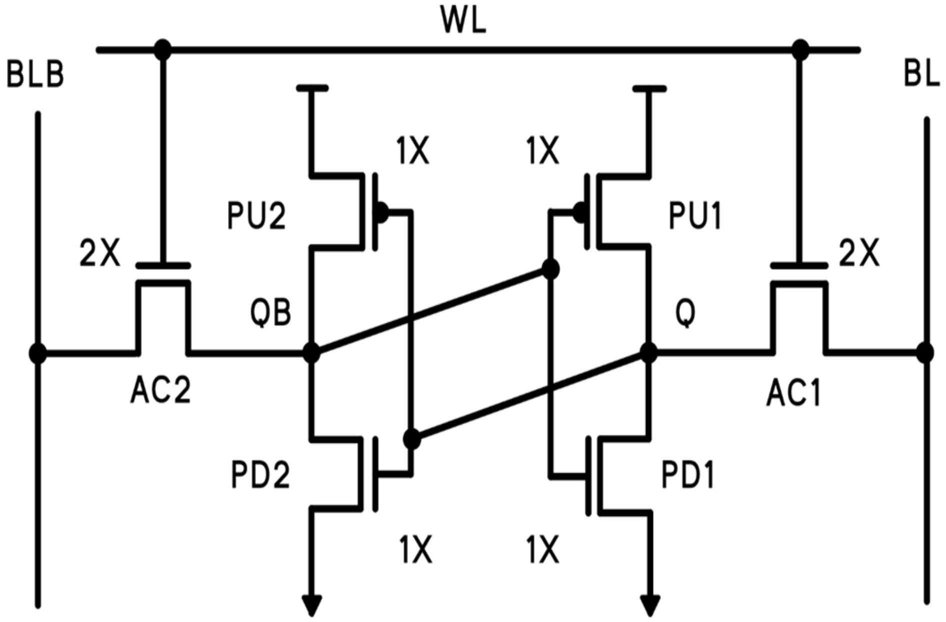


Նկ. 1.29. Ընդհանուր շեղման լարմամբ ՍՀԲ-երի մատրիցի կառուցվածքը

ԵՏ-երով սխեմաների նախագծման արդյունավետության ուսումնասիրության մեկ այլ օրինակ են նախագծված և հետազոտված [95] ստատիկ հիշողության 6 (նկ. 1.31) [95] և 8 (նկ. 1.32) [95] տրանզիստոր ունեցող բջիջները:



Նկ. 1.30. Բազմամուտք ՍՀԲ-ի կառուցվածքը

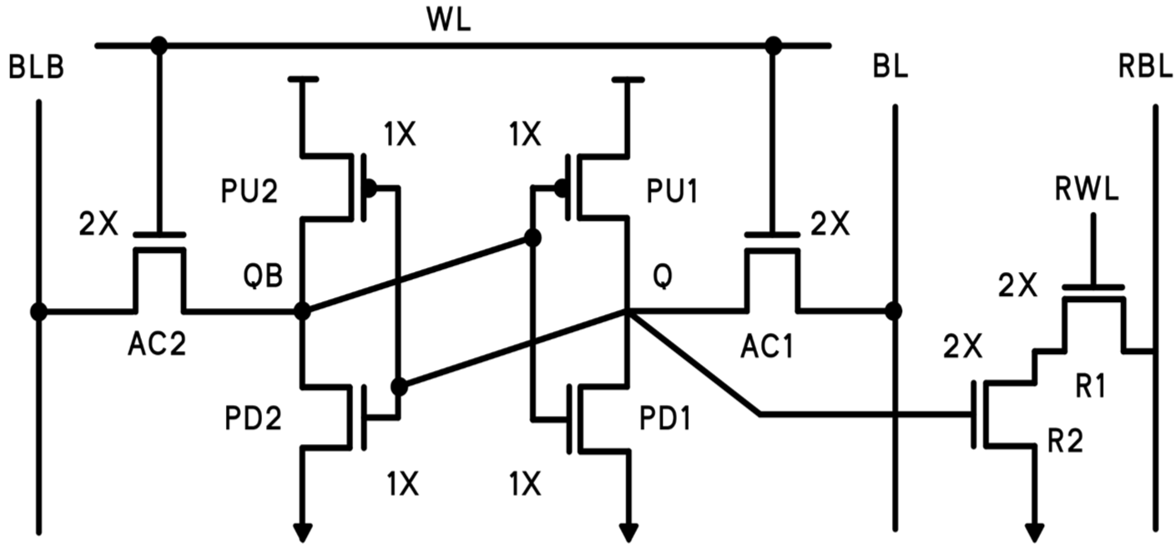


Նկ. 1.31. 6 տրանզիստոր ունեցող ստատիկ հիշողության բջջի կառուցվածքը

Նկ. 1.31-ում շրջիչներն օգտագործվում են տվյալային կարգը հիշելու համար, իսկ AC1 և AC2 տրանզիստորները՝ հիշողության մեջ գրելու և դրանից կարդալու համար

[95]: Բջջի հուսալի աշխատանքն ապահովելու նպատակով նշված գործողություններից ամեն մեկը պահանջում է համապատասխան տրանզիստորների գերակայող լինելը, ինչը ենթադրում է տրանզիստորների չափերի փոխզիջումային ընտրություն [95]: Նախագծման այդ բարդությունը շրջանցել են՝ օգտագործելով 8 տրանզիստոր պարունակող բջիջ (նկ. 1.32), որում գրանցման և ընթերցման գործողությունների ուղիներն առանձնացված են: Ստատիկ հիշողության բջիջների նախագծման հիմնական մարտահրավերը փոքր ցրման հոսանքների ապահովումն է, քանի որ դրանց աշխատանքի հիմնական ժամանակը սպասման ռեժիմում է: Բացի դա, մեծ տեղաբաշխման խտություն ստանալու համար օգտագործվում են նվազագույն չափեր ունեցող տրանզիստորներ:

Հետազոտվել են [95] վերոհիշյալ երկու տեսակի հիշողության բջիջների ընթերցման գործողության աղմկակայունությունը (ԸԳԱԿ), գրելու գործողության աղմկակայունությունը (ԳԳԱԿ), սնուցման լարման մասշտաբավորվելիությունը և էներգասպառումը:



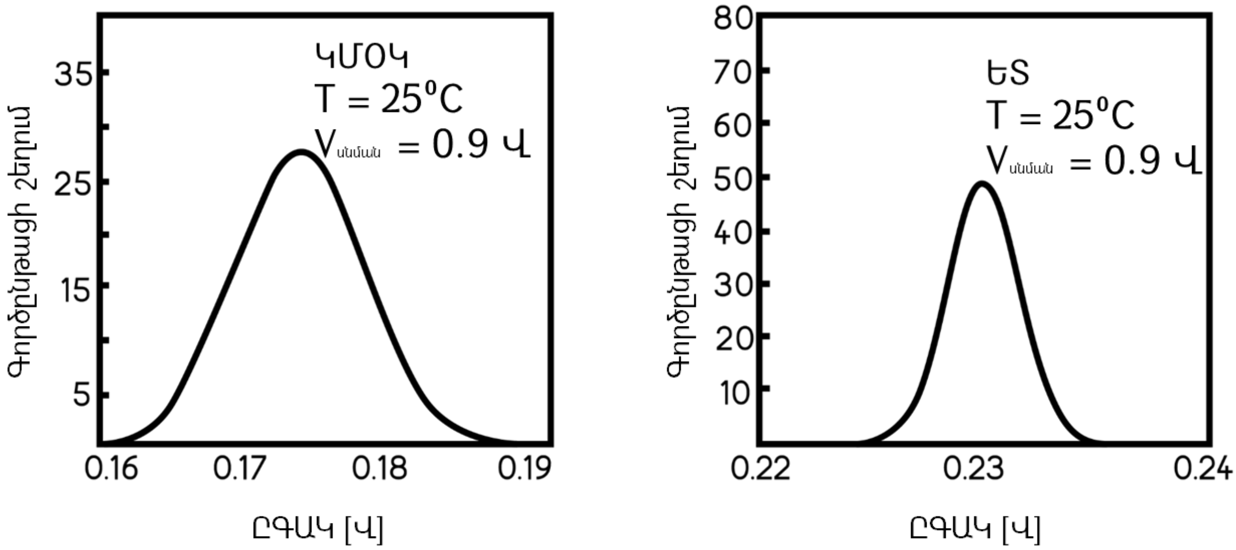
Նկ. 1.32. 8 տրանզիստոր ունեցող ստատիկ հիշողության բջջի կառուցվածքը

<u>6 տրանզիստորով (6Տ) ստատիկ հիշողության բջիջ</u>

Իրականացված Մոնտե Կառլո վերլուծության (ՄԿՎ) արդյունքները (նկ. 1.33) [95] ցույց են տվել, որ ԸԳԱԿ-ի արժեքը ԵՏ-ի դեպքում 39%-ով մեծ է՝ 223 *մՎ*, քան ԿՄՕԿ-ի դեպքում՝ 160 *մՎ* [95]: Դրա պատճառը ԵՏ-երի շեմային լարումների համեմատաբար փոքր շեղումներն են:

ՄԿՎ-ի արդյունքներով ցույց է տրվել, որ ԳԳԱԿ-ը ԵՏ-ի դեպքում 54%-ով ավելի լավ արժեք ունի, քան ԿՄՕԿ-ի դեպքում [95]:

Սնուցման լարման մասշտաբավորելիությունը գնահատելու համար արված ՄԿՎ-ի ժամանակ վերջինս փոփոխվել է 0.95 *Վ*-ից մինչև 0.2 *Վ* [95]: Ենթադրելով, որ ԸԳԱԿ-ը պետք է նվազագույնը կազմի սնուցման լարման 15%-ը, ֆինֆետի դեպքում նվազագույն



Նկ. 1.33. ՄԿՎ-ից ստացված ԸԳԱԿ-ի՝ տարբեր ջերմաստիճաններում արժեքները

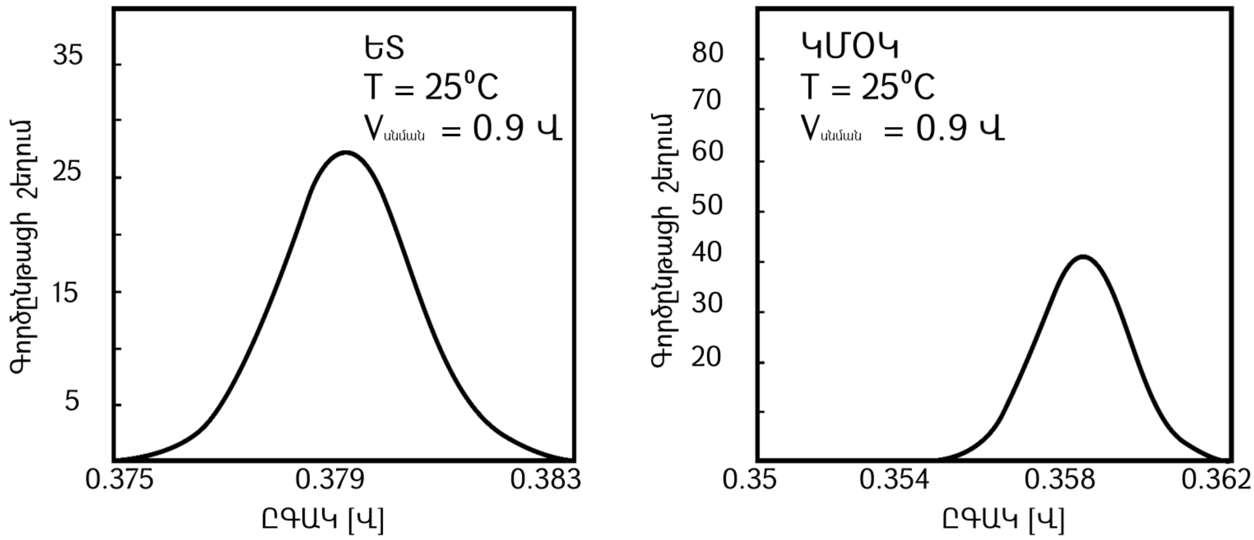
հնարավոր աշխատանքային լարումը ստացվել է 0.3 *Վ*, իսկ ԿՄՕԿ-ի պարագայում՝ 0.65 *Վ* [95]: Ակնհայտ է, որ ցածր լարումներով կիրառվելու դեպքում եռաչափ տրանզիստորներն ավելի նախընտրելի են:

Քննարկվող բջջի կորստային էներգասպառումը եռաչափ տրանզիստորների դեպքում կազմել է 72 *պՎտ*, իսկ ԿՄՕԿ-ի դեպքում՝ 530 *պՎտ*, ինչը ցածր էներգասպառում պահանջող նախագծերում առավելություն է տալիս եռաչափ տրանզիստորին [95]:

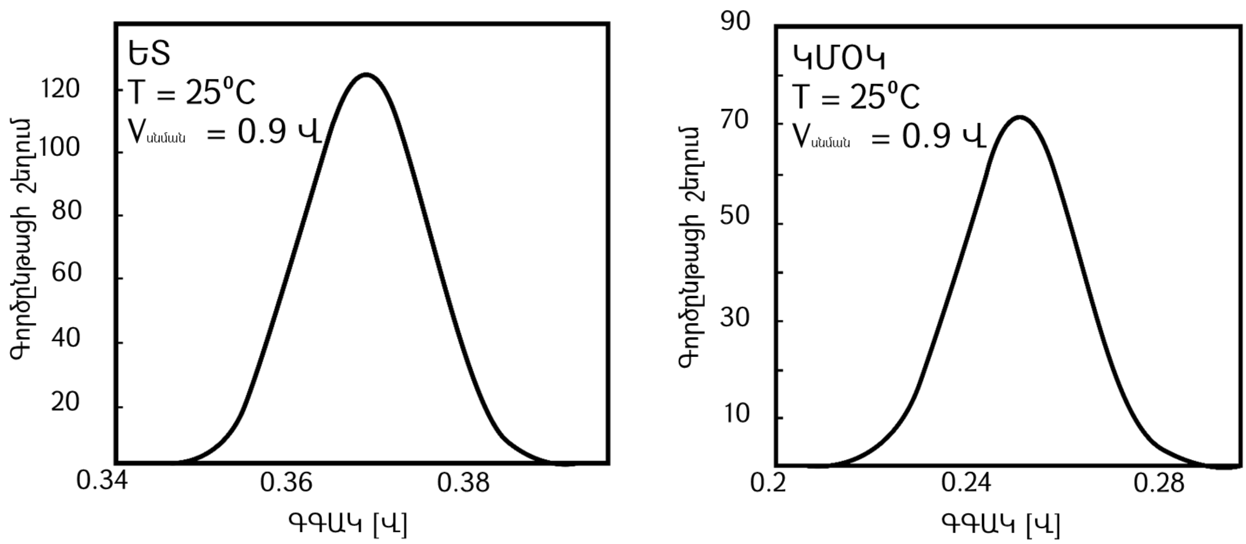
<u>8 տրանզիստորով (8Տ) ստատիկ հիշողության բջիջ</u>

8Տ կառուցվածքի համար արված ՄԿՎ-ն ցույց է տվել, որ ԵՏ տեխնոլոգիայի դեպքում ԸԳԱԿ-ը ունի 375 *մՎ* արժեքը, իսկ ԿՄՕԿ-ի դեպքում՝ 352 *մՎ* (նկ. 1.34) [95]: Այս կառուցվածքում ԸԳԱԿ-ը բարելավված է ընթերցման և գրանցման առանձնացված ուղիների շնորհիվ: Ստատիկ հիշողության այս բջջի համար ստացվել են ԳԳԱԿ-ի հետևյալ արժեքները՝ 365 *մՎ* ԵՏ դեպքում և 352 *մՎ* ԿՄՕԿ-ի դեպքում (նկ. 1.35) [95]:

Նվազագույն սնուցման լարումը պարզելու համար այս դեպքում ևս սխեմայի աշխատանքը դիտարկվել է 0.95 *Վ*-ից մինչև 0.2 *Վ*: Եռաչափ տրանզիստորներով կառուցված 8Տ բջջի նվազագույն աշխատանքային լարումը ստացվել է 0.15 *Վ*, իսկ ԿՄՕԿ-ի համար 0.3 *Վ*: Այս բջջի հզորության սպառումը հետազոտելիս ԵՏ դեպքում ստացվել է 165 *պՎտ*, իսկ ԿՄՕԿ-ի դեպքում 53 *պՎտ*:



Նկ. 1.34. 8Տ կառուցվածքով ՄԿՎ-ի արդյունքում ԵՏ և ԿՄՕԿ սխեմաների համար ԸԳԱԿ-ի ստացված արժեքները



Նկ. 1.35. 8Տ կառուցվածքով ՄԿՎ-ի արդյունքում ԵՏ և ԿՄՕԿ սխեմաների համար ԳԳԱԿ-ի ստացված արժեքները

Այսպիսով, եռաչափ տրանզիստորներով սխեմաների նախագծման առկա միջոցների վերլուծութունը ցույց տվեց, որ ցածր էներգասպառման և արագագործության տեսանկյունից դրանք արդյունավետ նախագծման միջոցներ են: Սակայն գոյություն ունեցող գրեթե բոլոր միջոցներում ՄՕԿ սարքերը պարզապես փոխարինվել են բնութագրերով դրանց նման ԿՓԵՏ միացմամբ տրանզիստորներով, ինչի պատճառով ԱՓԵՏ սարքերի արագագործությունն ու էներգասպառումը չեն ենթարկվել բավարար հետազոտման: Ուստի առաջանում է սխեմաների նախագծման նոր միջոցների ստեղծման անհրաժեշտություն:

1.3. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման միջոցներին ներկայացվող պահանջները և առաջարկվող սկզբունքները

ԻՍ-երն առաջադրվող մասշտաբավորման, արագագործության, էներգասպառման և մակերեսի պահանջները բավարարելու համար անհրաժեշտ է կառուցել եռաչափ տրանզիստորներով սխեմաների նախագծման միջոցներ, որոնց ներկայացվում են որոշակի պահանջներ:

§1.2.-ից հետևում է, որ ԿՄՕԿ սխեմաների՝ ժամանակակից ԻՍ-երին առաջադրվող պայմաններն այլևս չբավարարելու պատճառով մշակվել են այլընտրանքային տեխնոլոգիական լուծումներ, որոնցից են եռաչափ բազմափական տրանզիստորային տեխնոլոգիաները:

ԻՍ-երի ներկայացվող սահմանափակումները հաշվի առնելով՝ կարելի է ընդհանրացնել, որ եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման միջոցները պետք է բավարարեն հետևյալ պահանջները.

- էներգասպառման նվազեցում: Էներգասպառումը կարելի է նվազեցնել՝ փոքրացնելով դրա ստատիկ և դինամիկ բաղադրիչներից առնվազն մեկը: Եռաչափ տրանզիստորների ենթաշեմային փոքր հոսանքների և լավ ղեկավարվող կարճ-հոսքուղային երևույթների շնորհիվ՝ հնարավոր է դառնում էներգասպառման ստատիկ բաղադրիչի էական նվազեցումը: Այդ սարքերի տարատեսակ միացումների միջոցով հնարավոր է նաև ղեկավարել էներգասպառման դինամիկ բաղադրիչը:

- Արագագործության հնարավորինս բարելավում: Տարատեսակ միացման եղանակների առկայության շնորհիվ՝ հնարավոր է ղեկավարել եռաչափ տրանզիստորների արագագործությունը: Քանի որ վերջինիս բարելավումը հանգեցնում է էներգասպառման վատթարացման, շատ կարևոր է ձգտել բարելավել էներգասպառում-հապաղում արտադրյալը:
- Զբաղեցրած մակերեսի նվազեցում: ԻՍ-երի ինտեգրման աստիճանը բարձրացնելու նպատակով եռաչափ տրանզիստորներով նախագծումը պետք է հետապնդի նաև նվազագույն մակերես ապահովելու նպատակ: Ինտեգրման բարձրացման շարունակականությունը կապահովվի միայն, եթե հնարավոր լինի կիրառվող նախագծման միջոցների շարունակական մասշտաբավորումը:

Ատենախոսությունում առաջարկվում են վերոհիշյալ պահանջները բավարարող, եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման նոր սկզբունքներ, որոնք կապահովեն գործնական պահանջները բավարարող ցածր էներգասպառում, արագագործություն և փոքր մակերես:

Մշակված է եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման 6 նոր մեթոդ: Դրանցից 4-ը վերաբերում են ՀԲ-երի նախագծմանը՝ օգտագործելով ԵՏ-երի ԱՓԵՏ միացման առանձնահատկությունները: Մյուս երկու մեթոդները դոմինո տրամաբանության և ստատիկ տրիգերների նախագծման սկզբունքորեն նոր մոտեցումներ են:

Եզրակացություններ

1. Ժամանակակից ԻՍ-երի բարձր արագագործության, ցածր էներգասպառման և ինտեգրման բարձր աստիճանի պահանջները թելադրում են եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման միջոցների մշակման անհրաժեշտությունը:
2. Եռաչափ տրանզիստորներով սխեմաների նախագծման առկա միջոցների վերլուծությունը ցույց է տալիս, որ վերջիններս լիովին չեն ապահովում ԻՍ-երին

առաջադրվող արագագործության և էներգասպառման պահանջների բավարարումը, ինչը նշանակում է, որ առաջացել է նոր սկզբունքներով գործող նախագծման միջոցների մշակման անհրաժեշտություն:

3. Ձևակերպվել են եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների մշակման միջոցներին առաջադրվող պահանջները, որոնց բավարարումը կապահովի ԻՍ-երին առաջադրված պահանջները:

ԳԼՈՒԽ 2. ԵՌԱՉԱՓ ՏՐԱՆԶԻՍՏՈՐՆԵՐՈՎ ՄԻԿՐՈԷԼԵԿՏՐՈՆԱՅԻՆ ՍԽԵՄԱՆԵՐԻ ՆԱԽԱԳԾՄԱՆ ԱՌԱՋԱՐԿՎՈՂ ՄԻՋՈՑՆԵՐԸ

Աշխատանքի նպատակը եռաչափ տրանզիստորներով սխեմաների նախագծման այնպիսի մեթոդների և միջոցների համալիրի մշակումն է, որոնց կիրառումը ԻՍ-երում թույլ կտա նվազեցնել վերջիններիս էներգասպառումը, բարելավել արագագործությունը և ապահովել համակարգի հնարավորինս փոքր մակերես: Առաջարկվող մեթոդները պետք է լինեն կիսահաղորդչային սարքերի երկրաչափական չափերից անկախ:

Սխեմաների նախագծման առկա միջոցների թերությունները հաշվի առնելով՝ մշակվել են ԻՍ-երի մակրոբնութագրերը բարելավող մեթոդներ, որոնցով սխեմաների նախագծումն իրականացվում է նոր սկզբունքներով աշխատող միջոցներով:

Առաջարկվել է ԵՏ-երով սխեմաների նախագծման 6 մեթոդ.

- Չորսը ԱՓԵՏ միացման առավելություններն օգտագործող հիշողության բջիջների նախագծման մեթոդներ են:
- Մեկ նախագծման մեթոդ նվիրված է արագագործ և էներգախնայող դոմինո տրամաբանության իրագործմանը:
- Առաջարկվող վերջին մեթոդն ապահովում է էներգախնայող և արագագործ ստատիկ տրիգերների նախագծումը:

Վերոհիշյալ մեթոդների հիման վրա մշակվել են թվային սխեմաներ, որոնք ապահովում են ԻՍ-երի մակրոպարամետրերին առաջադրված պահանջների բավարարումը:

Առաջարկված մեթոդների գործնական կիրառմամբ իրականացվել են տարբեր չափերով ստատիկ հիշողության մատրիցների, դոմինո տրամաբանությամբ փականների և ստատիկ տրիգերների նախագծեր:

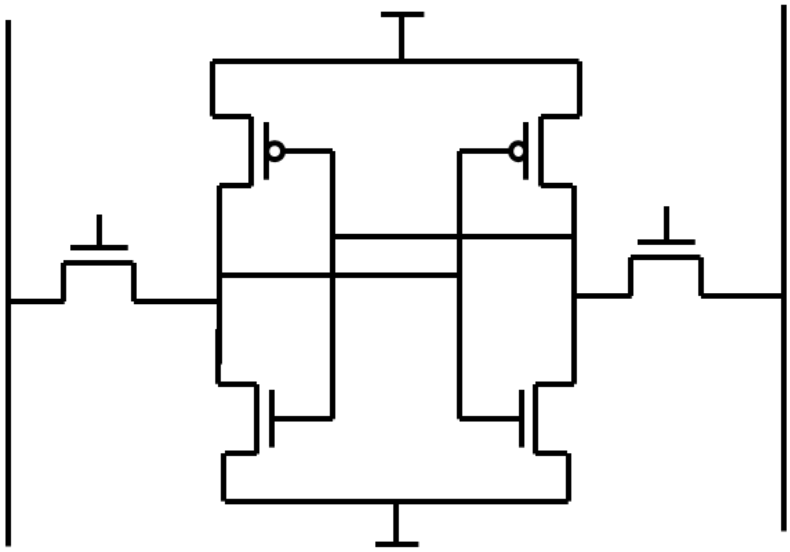
2.1. Եռաչափ տրանզիստորներով նախագծված հիշողության բջիջների ստատիկ աղմկակայունության պաշարի որոշումը

Ժամանակակից ԻՍ-երի ընդհանուր մակերեսի ավելի քան 40%-ը զբաղեցնում է հիշողության բջիջների մատրիցը [96,97], ինչի պատճառով վերջինիս բնութագրերն էական ազդեցություն ունեն համակարգի արագագործության [98], էներգասպառման [99] ու մակերեսի վրա [100]:

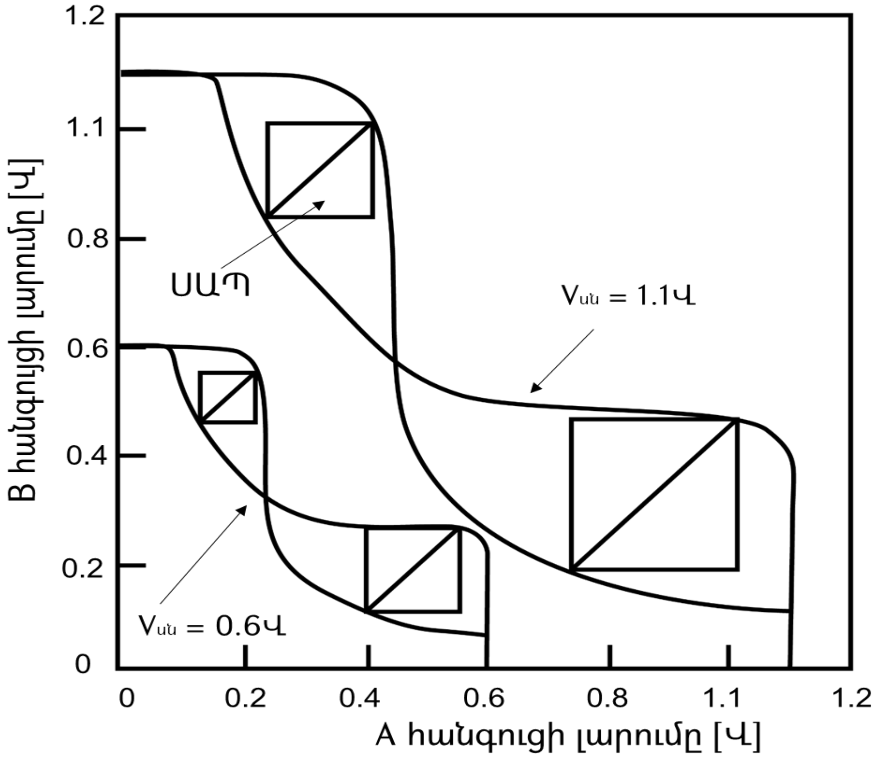
Հիշողության բջիջների կարևորագույն բնութագրերն են ստատիկ վիճակում ցածր էներգասպառումը [101] և գրանցման/ընթերցման գործողությունների արագությունն ու ճշգրտությունը [102, 103]: Գործողությունների ժամանակ տվյալների կայունությունը որոշվում է ստատիկ աղմկակայունության պաշարի (ՍԱՊ) միջոցով: Այս մեծության գնահատումը և բարելավումը ծայրահեղ կարևորություն են ներկայացնում ՀԲ-երի բնականոն աշխատանքի համար: 6Տ կառուցվածքի (նկ. 2.1) [104] օրինակով ՍԱՊ-ի մեծությունը գնահատվում է՝ միացնելով բառի և կարգային գծերը սնուցման լարմանը և իրականացնելով հետևյալ գործողությունները.

- A հանգույցի լարումը 0 Վ-ից մինչև սնուցման $V_{\text{սն}}$ լարում փոխելիս չափվում է B հանգույցի լարումը: Կառուցվում է դրանց՝ միմյանցից կախվածության գրաֆիկը (նկ. 2.2):
- B հանգույցի լարումը 0 *Վ*-ից մինչև սնուցման $V_{\text{սն}}$ լարում փոխելիս չափվում է A հանգույցի լարումը: Կառուցվում է դրանց՝ միմյանցից կախվածության գրաֆիկը (նկ. 2.2):
- A և B գծերի միջև կառուցվում են այդտեղ տեղավորվող ամենամեծ քառակուսիները: Դրանցից փոքրի անկյունագծի երկարությունը սահմանվում է որպես ՍԱՊ:

Սնուցման լարման 0,6 *Վ* և 1,1 *Վ* արժեքներին համապատասխանող A և B գծերից (նկ. 2.2) երևում է, որ փոքր սնուցման լարմանը համապատասխանում է համեմատաբար փոքր ՍԱՊ: ՀԲ-ի կառուցվածքում օգտագործվող տրանզիստորների շեմային լարումները էական ազդեցություն ունեն ՍԱՊ-ի արժեքի վրա [105]: Մասնավորապես, շրջիչների n տրանզիստորների շեմային լարման փոքր արժեքները հանգեցնում են ՍԱՊ-ի նվազման, ինչը նվազեցնում է ՀԲ-ի կայունությունը: Կայունության ապահովման նպատակով անհրաժեշտ է ապահովել հասանելիության և հողանցման տրանզիստորների լայնությունների հնարավորինս մեծ հարաբերություն, որը հայտնի է β հարաբերություն անվամբ: Հաշվի առնելով ԻՍ-երի սնուցման լարման շարունակական նվազեցումը և տրանզիստորների շեմային լարումների փոքրացումը՝ էլ ավելի է կարևորվում նախագծման այնպիսի միջոցների մշակումը, որոնք թույլ կտան ապահովել անհրաժեշտ մեծությամբ ՍԱՊ՝ կիրառելով ցածրավոլտ սնուցման լարում:



Նկ. 2.1. 6Տ կառուցվածքով ՀԲ-ը



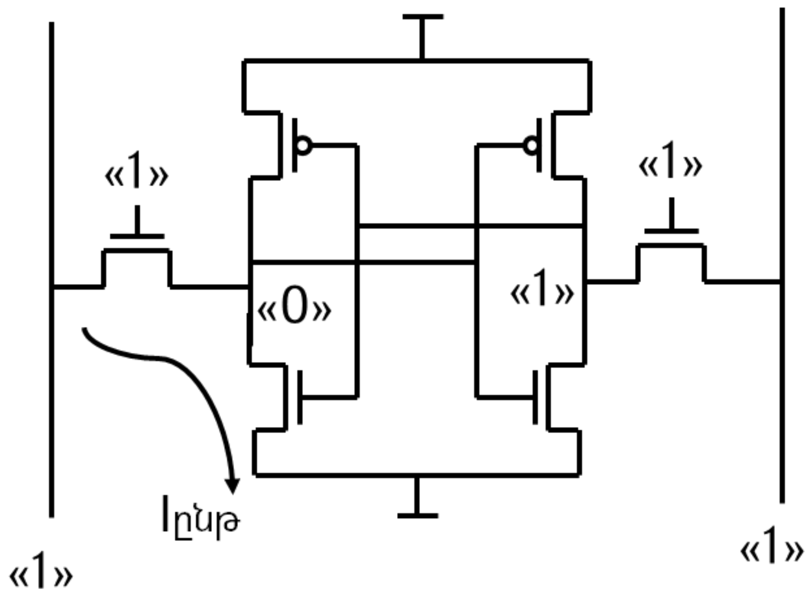
Նկ. 2.2. Շրջիչների բնութագրերից ՀԲ-ի ՍԱՊ-ի որոշումը

ՀԲ-ի կայունությունը գրանցման և ընթերցման գործողությունների բնականոն իրականացման հիմնական պայմաններից է: Ստորև քննարկվում են տրամաբանական արժեքների գրանցման և ընթերցման բնականոն և խափանվող գործընթացները:

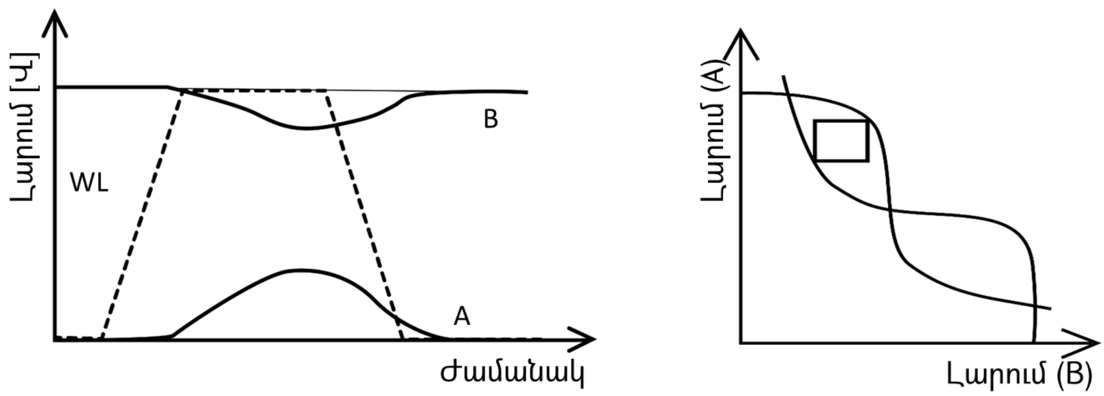
ՀԲ-ում տրամաբանական «0» արժեք գրանցելիս [106]՝ կարգային գծերը (ԿԳ), կախված գրանցվող արժեքից և ակտիվացված բառի գծից (ԲԳ), ընդունում են տրամաբանական «1» և տրամաբանական «0» արժեքներ, իսկ հիշողության A և B հանգույցների (ՀՀ) արժեքները փոխատեղվում են: Դրանով ավարտվում է գրանցման գործողությունը: «0»-ի ընթերցման ժամանակ ԿԳ-երը նախապես լիցքավորվում են «բարձր» լարում ապահովող լիցքով, և ապա ԲԳ-ն ակտիվացվում է: Բջջում պահպանված արժեքը հայտնվում է ԿԳ-երի վրա՝ դրանց միջև առաջացնելով պոտենցիալների տարբերություն: ՀԲ-ին միացված ուժեղարարների միջոցով այդ տարբերությունն ընթերցվում է: Հաջորդիվ, բջջի մեջ «1»-ի գրանցման ժամանակ, հնարավոր են հետևյալ տարբերակները.

- Գրանցման գործողության (ԳԳ) խափանում, երբ հիշողության երկու հանգույցները պահպանում են իրենց արժեքները, որը հաջորդ ընթերցման գործողության (ԸԳ) ժամանակ հանգեցնում է սխալ արժեքի ընթերցման:
- ԳԳ-ի հաջողում, ինչին հետևում է ճիշտ արժեքի ԸԳ: Վերջինիս ժամանակ հնարավոր է՝ տեղի ունենա հիշողության հանգույցների արժեքների փոխատեղում, ինչը կհանգեցնի ՀԲ-ում պահվող արժեքի ոչնչացման և ԸԳ-ի խափանման:

Այսպիսով, ԸԳ-ն իրականացվում է նախապես լիցքավորված ԿԳ-երի՝ հասանելիության տրանզիստորով և ՀՀ-ով անցնող հոսանքի միջոցով լիցքաթափմամբ (նկ. 2.3): Հասանելիության տրանզիստորի և n տրանզիստորի՝ լարման բաժանիչի նմանվող միացման պատճառով ընթերցման ժամանակ տեղի է ունենում «0» լարմամբ ՀՀ-ի պոտենցիալի աճ և «1» լարմամբ ՀՀ-ի նույն չափով նվազում (նկ. 2.4): Փոփոխության չափի՝ շրջիչների տրամաբանական արժեքների շեմից մեծ արժեք ընդունելու դեպքում տեղի կունենա վերջիններիս ելքային արժեքների փոխատեղում՝ ԸԳ-ի խափանում (նկ. 2.5) [107]: Գրանցման գործողություն սկսելուց առաջ A հանգույցը լիցքավորվում է մինչև «1», իսկ B-ն՝ «0» լարումներ (նկ. 2.6): Երբ գրանցվող «ցածր» լարումը տրվում է համապատասխան ԿԳ-ին, տեղի է ունենում $I_{գր}$ հոսանքի միջոցով A հանգույցի լիցքաթափում, ինչը հանգեցնում է ՀՀ-երի արժեքների փոխատեղման՝ նոր արժեքի գրանցմանը (նկ. 2.7ա):

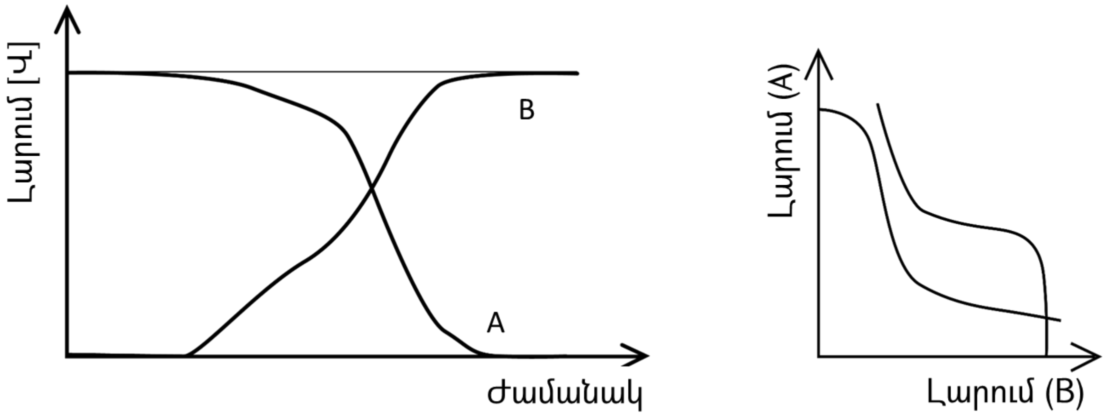


Նկ. 2.3. 6Տ ՀԲ-ի ընթերցման գործողության ազդանշանները



ա) բ)

Նկ. 2.4. Բնականոն ընթերցման գործողությունը

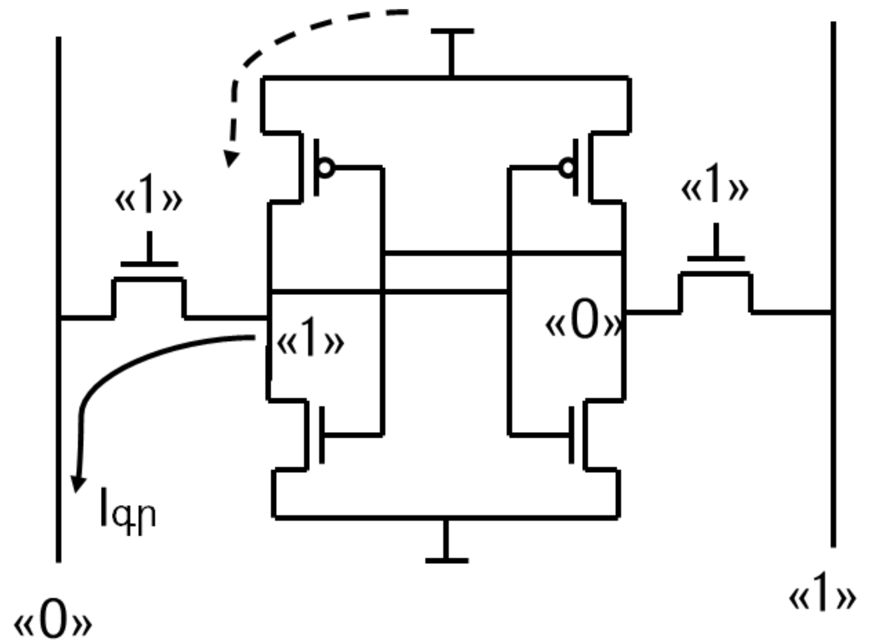


ա) բ)

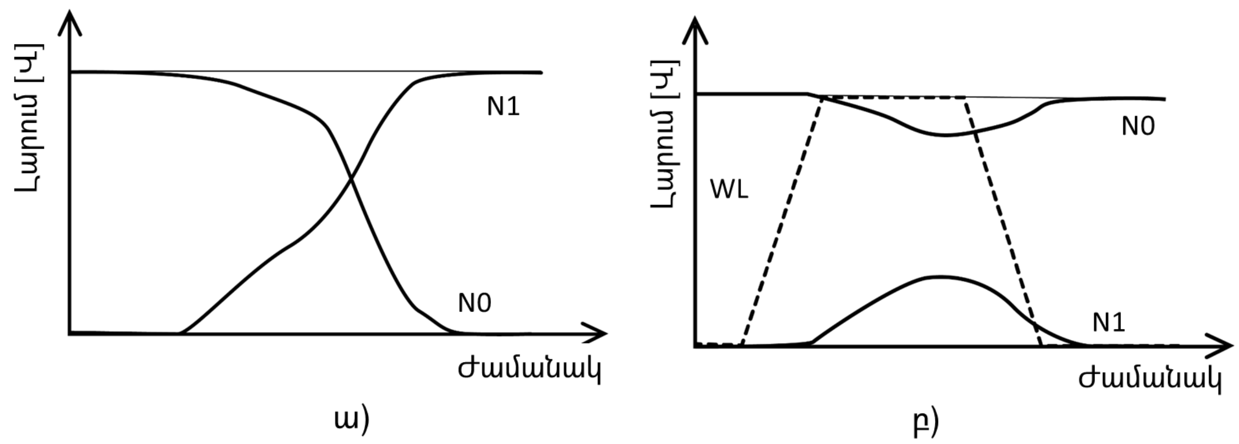
Նկ. 2.5. Խափանված ընթերցման գործողությունը

ԳԳ-ի ժամանակ p տրանզիստորից դեպի A հանգույց հոսում է որոշակի հոսանք, որը սկսում է լրացնել կորցրած լիցքի պաշարը: Եթե այդ հոսանքն ավելի մեծ է, քան $I_{գր}$ – ը, ապա նշված ՀՀ-ն չի լիցքաթափվում, և նոր արժեքի գրանցումը խափանվում է (նկ. 2.7բ):

Վերևում շարադրվածից ակնհայտ է, որ ԳԳ-ի և ԸԳ-ի խափանումները բացառելու համար անհրաժեշտ է կատարել համապատասխան տրանզիստորների չափերի օպտիմալ ընտրություն: Սակայն խնդիրըն այն է, որ այդ երկու գործողությունների կայունացման պայմանները իրար հակասում են, ինչը լրացուցիչ մարտահրավեր է ՀԲ-երի նախագծման գործում:



Նկ. 2.6. 6S ՀԲ-ի գրանցման գործողության ազդանշանները



Նկ. 2.7. Բնականոն և խափանված գրանցման գործողությունները

2.2. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման միջոցների մշակումը

2.2.1. Եռաչափ տրանզիստորներով ստատիկ հիշող բջիջների նախագծման միջոցների մշակումը

ՀԲ-երի բնութագրերի՝ ԻՍ-երում ունեցած կարևորությունը հաշվի առնելով՝ կատարվել է տարբեր տեխնոլոգիական գործընթացների դեպքում դրանց բնութագրերի ուսումնասիրություն, և առաջարկվում է ցածր էներգասպառմամբ և մեծ ՍԱՊ-ով ՍՀԲ-երի նախագծման 4 մեթոդ, որոնք հիմնված են ԵՏ-երի անկախ փականային միացման առավելությունների և նախագծման ընթացքում ճկունության վրա: Մեթոդների հիման վրա մշակվել են 6S կառուցվածքով տարբեր ՀԲ-եր:

Ուսումնասիրվել են [108] ՀԲ-երի 5S, 6S և 8S կառուցվածքները: Կատարվել են դրանց հիմնական պարամետրերի՝ էներգասպառման, կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի և կայունության համեմատություններ (աղ. 2.1):

Աղյուսակ 2.1

32/28 *նմ* և 16 *նմ* տեխնոլոգիաներով մոդելավորման արդյունքները

Բջիջ	Լարում (*Վ*)		Հոսանք (*մկԱ*)		Կորստի հոսանք (*մկԱ*)	
	16 *նմ*	32/28 *նմ*	16 *նմ*	32/28 *նմ*	16 *նմ*	32/28 *նմ*
6S	0,75	1,05	7,2	15	0,57	1,1
5S	0,75	1,05	10,1	17,5	1,007	2,05
8S	0,75	1,05	6,8	13	0,4	0,95

ԵՏ-երով ՀԲ-երի իրականացումն ապահովել է վերջիններիս մակրոբնութագրերի կրկնակի բարելավում՝ 32/28 նմ տեխնոլոգիական գործընթացով իրականացված ԿՄՕԿ սխեմաների արդյունքների համեմատ:

Մոդելավորումը կատարվել է աշխատանքային երեք ռեժիմներում՝ TT - 25°C, FF - 40 $^\circ$C, SS - 95 $^\circ$C: Ընտրվել է սնման լարման այն նվազագույն արժեքը, որի դեպքում հիշողության բջիջը դեռևս պահպանում է իր ֆունկցիոնալությունը (գրանցում/ընթերցում): Աղ. 2.2-ից [108] կարելի է եզրակացնել, որ սխեմաների՝ ԵՏ-երով իրականացումը ընձեռում է սնուցման լարման կրկնակի նվազեցման հնարավորություն, ինչն էապես կրճատում է այդ տրանզիստորներով նախագծված ԻՍ-երի ընդհանուր էներգասպառումը:

6Տ և 8Տ կառուցվածքով ՀԲ-երի համեմատ 5Տ ՀԲ-ն ունի նվազագույն մակերես՝ 9% 32/28 *նմ* -ով, իսկ 16 *նմ* -ով՝ 60%, սակայն զիջում է 8Տ ՀԲ-ին՝ ծախսած հզորությամբ՝ 23% 32/28 *նմ* -ով, իսկ 16նմ -ով՝ 12%: Ժամանակային պարամետրերը լավացվել են 47..55% 16 *նմ* տեխնոլոգիայով նախագծված ՀԲ-ի ճարտարապետության դեպքում:

Աղյուսակ 2.2

Հիշողության բջիջների նվազագույն սնման լարման արժեքները երեք աշխատանքային պայմաններում՝ TT, FF, SS

Բջիջ	Աշխատանքային պայմաններ	Նվազագույն սնման լարման արժեքներ VDD (*Վ*)	Նվազագույն սնման լարման արժեքներ VDD (*Վ*)
		32/28 *նմ*	16 *նմ*
6Տ	TT	0,77	0,394
	FF	0,8	0,41
	SS	0,74	0,377
5Տ	TT	0,73	0,35
	FF	0,75	0,4
	SS	0,7	0,3
8Տ	TT	0,8	0,51
	FF	0,82	0,47
	SS	0,78	0,39

ՀԲ-երի ժամանակային բնութագրերի համար ստացված արդյունքները (աղ. 2.3) հավաստում են ԵՏ-երի՝ արագագործ կիրառություններում արդյունավետ նախագծման միջոց լինելը:

§ 1.2. -ում քննարկվել են երկփական տրանզստորների՝ իրարից բնութագրերով տարբերվող միացումներ, որոնց շահագործման հիմնական նպատակն է համապատասխան կիրառությանն առաջադրված պահանջների բավարարումը: ԵՏ-երի շեմային լարման մոդուլացիայի հնարավորությունը թույլ է տալիս բարձրացնել ՀԲ-ի ՍԱՊ-ը՝ միաժամանակ մեծացնելով հիշողության մատրիցում ինտրեգրման աստիճանը և նվազեցնելով ստատիկ և դինամիկ էներգասպառումը [108]:

Աղյուսակ 2.3

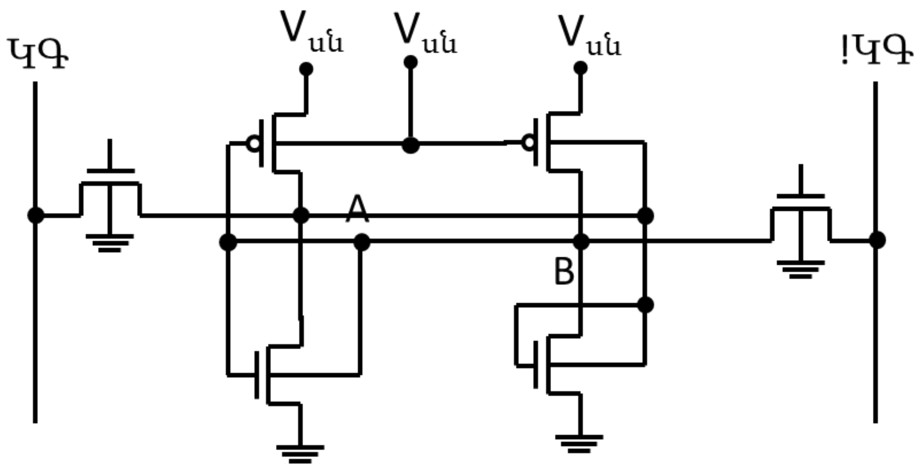
Հիշողության բջիջների ժամանակային պարամետրը և մակերեսը երեք աշխատանքային պայմաններում՝ TT, FF, SS

Բջիջ	Աշխատան քային պայմաններ	Ժամանակային պարամետր tr/tf (*պվ*)	Մակերես (մկմ^2)	Ժամանակային պարամետր tr/tf (*պվ*)	Մակերես (մկմ^2 ^{2})
		32/28 *նմ*		16 *նմ*	
6S	TT FF SS	62/65 59/61 65/70	2,12	34/36 27/31 36/39	1,01
5S	TT FF SS	73/75 69/72 76/80	1,93	38/41 37/41 38/39	0,91
8S	TT FF SS	65/67 60/65 70/72	3,33	32/34 28/29 31/34	1,63

Առաջարկվում են [108-112] ՀԲ-երի նախագծման 4 մեթոդ և դրանց հիման վրա մշակված բջիջներ: Դրանցում բոլոր ԱՓԵՏ սարքերն ունեն նվազագույն՝ մեկ ֆին լայնություն:

Միջոց 1.

Առաջարկվող առաջին ՍՀԲ-ի դեպքում (նկ. 2.8) հողանցման տրանզիստորները միացված են ԿՓԵՏ միացմամբ [108], իսկ հասանելիության և սնուցման տրանզիստորները ԱՓԵՏ միացմամբ և միափական ռեժիմում գործող սարքեր են: Վերջին դեպքում n տիպի տրանզիստորների մի փականը հողանցված է, իսկ p տիպինը՝ միացված է սնուցման լարմանը: Այդպիսի միացումը ՀԲ-երի տրանզիստորների համար ապահովում է բարձր շեմային լարում և հետևաբար՝ բարելավում է ԸԳ-ի ՍԱՊ-ը՝ ապահովելով գործողության ժամանակ տվյալի կայունությունը: ԱՓԵՏ միացման ժամանակ β հարաբերությունը հնարավոր է դառնում մեծացնել, ինչն ուղղակիորեն ազդում է ՍԱՊ-ի արժեքի վրա:



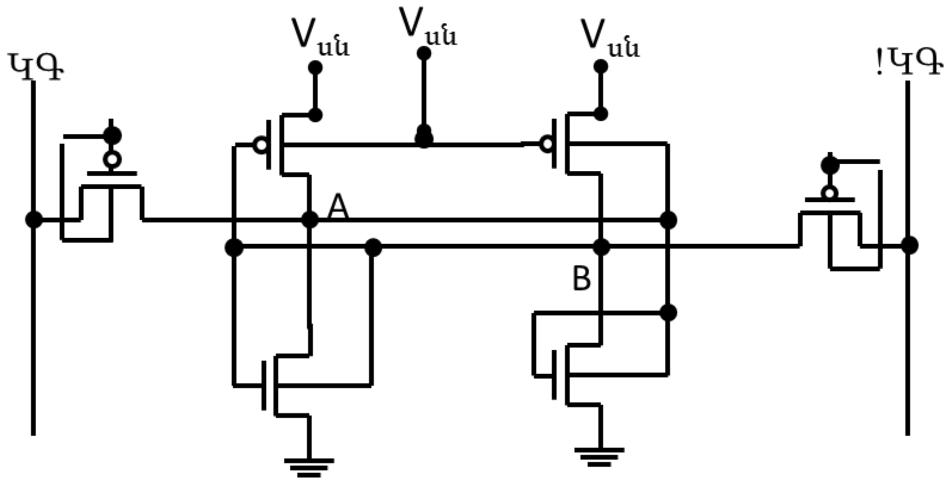
Նկ. 2.8. Բարձր ՇԼ-ով հասանելության տրանզիստորներով ՀԲ-ը

<u>Միջոց 2.</u>

Առաջարկվող երկրորդ ԵՏ ՀԲ-ում (նկ. 2.9) հողանցման և հասանելիության տրանզիստորներն ունեն ԿՓԵՏ միացում [109], իսկ սնուցման տրանզիստորները միափական ռեժիմում գործող ԱՓԵՏ միացմամբ են, որոնց մի փականը միացված է սնուցման լարմանը: Այստեղ ԿԳ-ի հասանելիության տրանզիստորները p տիպի «թույլ» տրանզիստորներ են: ԸԳ-ի ժամանակ, P հասանելիության տրանզիստորի միջոցով «1» արժեքն ունեցող հանգույցն ուժեղացնելու շնորհիվ, ապահովվում են պահվող տվյալի կայունությունը և ՍԱՊ-ի մեծ արժեքը: Համանման ձևով՝ ԳԳ-ի ժամանակ ՍԱՊ-ը ավելի մեծ է, քան նույն չափի ԿՓԵՏ միացման դեպքում:

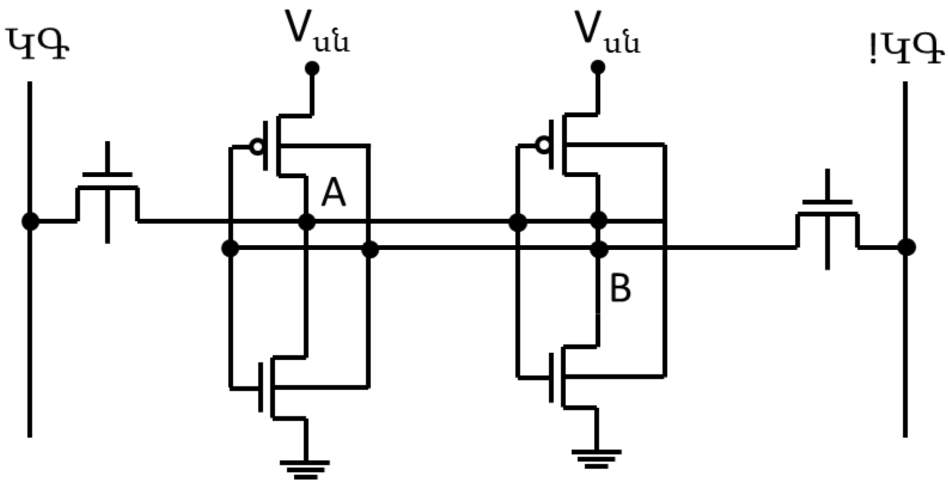
<u>Միջոց 3.</u>

ՀԲ-ի՝ առաջարկվող երրորդ կառուցվածքում (նկ. 2.10) շրջիչները կազմող տրանզիստորները ԿՓԵՏ միացմամբ են [110], իսկ հասանելիության տրանզիստորները՝ ԱՓԵՏ: Այս դեպքում, օգտագործելով ԱՓԵՏ սարքերի փականների շեղման միջոցով շեմային լարման մոդուլյացիայի հնարավորությունը, ԳԳ-ի և ԸԳ-ի ժամանակ իրականացվում է տվյալ գործողության բնութագրերի բարելավմանը նպաստող տրանզիստորների ուժերի դինամիկ համապատասխանեցում: Այս կառուցվածքում ԸԳ-ը և ԳԳ-ն իրականացվում են իրարից անկախ մեխանիզմների միջոցով:



Նկ. 2.9. P տիպի հասանելիության տրանզիստորներով ՀԲ-ը

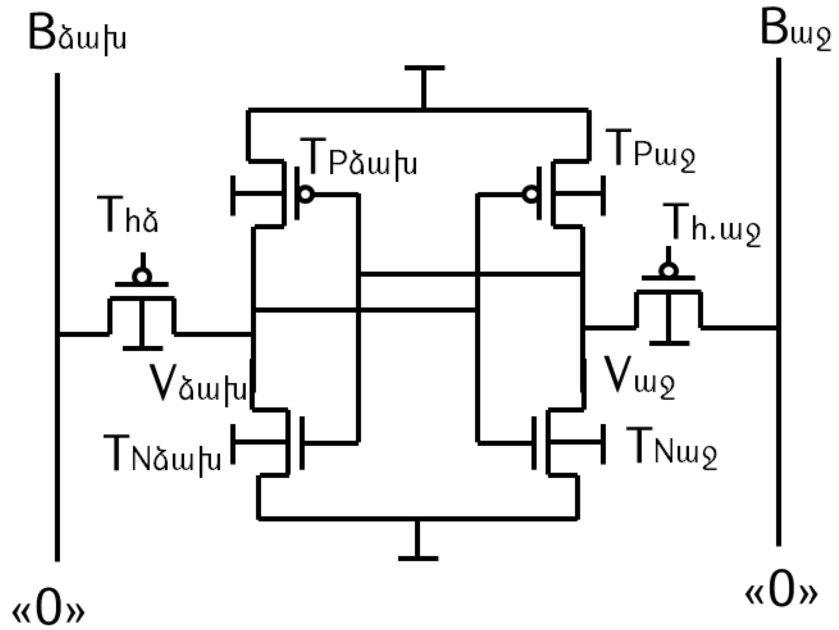
Հասանելիության տրանզիստորներից յուրաքանչյուրի փականներից մեկը ղեկավարվում է ընթերցման-գրանցման ազդանշանով, իսկ մյուսը՝ առանձնացված գրանցման ազդանշանով: Նշված երկու ազդանշաններին ՀԲ-ի ստատիկ ռեժիմում տրվում է «0» լարում: Հասանելիության տրանզիստորներն ընթերցման գործողության ժամանակ հանդիսանում են բարձր շեմային լարմամբ և փոքր հոսանք անցկացնող սարքեր: ԸԳ-ի ժամանակ հնարավոր աղմուկներն այս կառուցվածքում ևս ճնշվում են՝ ապահովելով գործողության ժամանակ տվյալի բարձր կայունություն: ԳԳ-ի ժամանակ հասանելիության տրանզիստորները ղեկավարող ազդանշաններն ընդունում են «1» արժեքը, և այդ սարքերը դառնում են փոքր շեմային լարմամբ և մեծ հոսանք անցկացնող տրանզիստորներ:



Նկ. 2.10. Շեմային լարման դինամիկ փոփոխությամբ ՀԲ-ը

Միջոց 4.

ՍՀԲ-երի նախագծման առաջարկված չորրորդ մեթոդը [111] հիմնված է ներկառուցված հետադարձ կապի և փականների շեմային լարումների դինամիկ համապատասխանեցման վրա: Մեթոդն իրագործվել է p տիպի հասանելիության տրանզիստորներով 6S կառուցվածքի օրինակով (նկ. 2.11):



Նկ. 2.11. 6S ՀԲ-ի՝ ԵՏ-երով իրականացումը

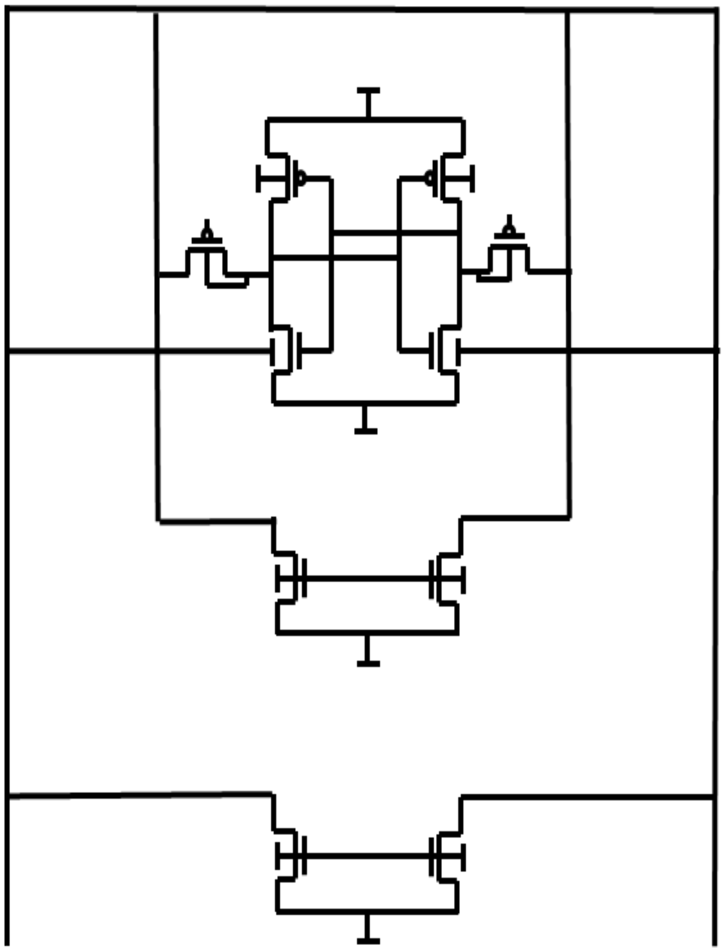
Նկ. 2.11-ում ենթադրվում է, որ $V_{\text{ձախ}}$ = «1», իսկ $V_{\text{աջ}}$ = «0»: Ընթերցման գործողությունն իրականացվում է հասանելիության տրանզիստորի միջոցով լիցքավորելով $B_{\text{ձախ}}$ ԿԳ-ը: $T_{\text{hձ}}$ և $T_{\text{Pձախ}}$ տրանզիստորների լարման բաժանիչ միացման շնորհիվ՝ տեղի է ունենում $V_{\text{ձախ}}$-ի արժեքի ΔV չափով նվազում: Այս մեծության մեծ արժեքների դեպքում շրջիչի ելքը կարող է փոխանջատվել, ուստի անհրաժեշտ է ապահովել ΔV-ի բավարար փոքր արժեք: ΔV-ը փոքրանում է $T_{\text{Pձ}}$-ից $T_{\text{hձ}}$ հոսող հոսանքի միջոցով, այսինքն՝ հանգույցի լարման փոփոխության նվազեցման համար անհրաժեշտ է մեծացնել $T_{\text{hձ}}$-ի շեմային լարումը և փոքրացնել $T_{\text{Pձախ}}$-ինը: ԱՓԵՏ միացմամբ տրանզիստորներում դա կարելի է իրականացնել՝ նշված տրանզիստորների հետին փականներին ապահովելով համապատասխանաբար «1» և «0» լարման մակարդակներ: ΔV-ի նվազեցմանը նպաստում է նաև $T_{Nձախ}$ տրանզիստորի կորստի հոսանքների նվազեցումը, որի համար անհրաժեշտ է ապահովել վերջինիս շեմային լարման բավարար մեծ արժեք: Դա կարելի է իրականացնել $T_{Nձախ}$-ի հետին փականին փակող «0» լարում կիրառելով: $V_{ձախ}$-ի նվազման ժամանակ $V_{աջ}$-ի կտրուկ աճը կանխելու համար անհրաժեշտ է ապահովել լիցքաթափող $T_{Nաջ}$ և $T_{h.աջ}$ տրանզիստորների գերակայություն՝ լիցքավորող $T_{Pաջ}$ –ի համեմատ, դրանց հետին փականներին կիրառելով համապատասխանաբար «1», «0» և «1» լարումները:

Գրանցման գործողության ժամանակ պատկերն այլ է: «1» արժեքը ՀԲ-ի մեջ գրանցելիս $B_{ձախ}$ = «0», իսկ $B_{աջ}$ = «1»: $T_{h.աջ}$-ի միջոցով տեղի է ունենում $V_{աջ}$-ի արժեքի աճ: Եթե լարման այդ աճը դառնում է բավարար մեծ ձախ շրջիչը փոխանջատելու համար, ապա ԳԳ-ը համարվում է հաջողված: $V_{աջ}$-ի հանգույցի լիցքավորումն անհրաժեշտ արագությամբ ապահովելու համար անհրաժեշտ է, որ լիցքավորումն ապահովող $T_{h.աջ}$-ը ուժեղ լինի լիցքաթափող $T_{Nաջ}$ -ից: Դա իրականացվում է վերջիններիս հետին փականներին «0» լարում կիրառելով: Մյուս կողմից՝ ձախ շրջիչի $V_{ձախ}$ լարման նվազման համար անհրաժեշտ է, որ $T_{Nձախ}$-ը լինի $T_{Pձախ}$-ից և $T_{hձ}$-ից ուժեղ: Դրա համար վերջիններիս հետին փականներին անհրաժեշտ է կիրառել «1» լարում:

Սպասման ռեժիմում ԿԳ-երը լիցքաթափված են [112], իսկ ԲԳ-ը պետք է լինի տրամաբանական «բարձր» լարմամբ: Վերոհիշյալ քննարկումների համաձայն՝ հիշվող տրամաբանական մակարդակների դեգրադացում կարող են առաջացնել $T_{Pաջ}$, $T_{hձ}$ և $T_{Nձախ}$ տրանզիստորների կորստի հոսանքները, որոնց նվազման համար տրանզիստորների հետին փականներին անհրաժեշտ է կիրառել համապատասխանաբար «1», «1» և «0» լարումներ: Մնացած տրանզիստորների հետին փականների միացումները այս ռեժիմում դեր չեն խաղում:

6Տ կառուցվածքի վերոհիշյալ վերլուծությունների արդյունքում պարզ է դառնում, որ որոշ տրանզիստորների հետին փականների օպտիմալ լարումները ՀԲ-ի աշխատանքի տարբեր ռեժիմներում ընդունում են տարբեր արժեքներ: Այդ պատճառով կիրառվող լարումների արժեքները պետք է փոփոխել դինամիկ կերպով: Կատարված մոդելավորումները ցույց են տվել, որ շրջիչների P տրանզիստորների շեմային լարումների համեմատ՝ ՍԱՊ-ի զգայնությունը էապես փոքր է P տրանզիստորների շեմային լարումներից ունեցած կախվածությունից, ինչի շնորհիվ հնարավոր է դառնում ամրակայել առաջինների հետին փականների լարումները՝ ԸԳ-ի և ստատիկ ռեժիմի համար նպաստավոր արժեքներով:

Առաջարկվող մեթոդի համաձայն՝ ՀԲ-ի հասանելիության տրանզիստորների հետին փականների լարումները միացվում են՝ հարակից ՀՀ-ին, իսկ շրջիչների N տրանզիստորներինը փոփոխվում են կախված աշխատանքային ռեժիմից: ԸԳ-ի ժամանակ N տրանզիստորների ՀՓ-երը միացվում են «0» լարմանը, ինչով ապահովվում է դրանց շեմային լարումների բարձր արժեքը: Վերոհիշյալ վերլուծության վրա հիմնվելով՝ ԳԳ-ի ընթացքում այդ տրանզիստորների ՀՓ-երին ապահովվում է միևնույն՝ «1» լարումը, որը նպաստում է ձախ շրջիչի ելքի հեշտ փոխանջատմանը: Ընթերցման և գրանցման գործողությունների համար անհրաժեշտ ՀՓ լարումներն ու ՀԲ-ի անհրաժեշտ ՍԱՊ-ն ապահովելու համար առաջարկված մեթոդի հիման վրա մշակվել է դինամիկ շեղմամբ ՀՓ-երով ՍՀԲ-երի նախագծման միջոց (նկ. 2.12):



Նկ. 2.12. Ներկառուցված հետադարձ կապով, դինամիկ շեմային լարմամբ 6Տ կառուցվածքը

Նկ. 2.11-ում պատկերված առանց ներկառուցված հետադարձ կապի սխեմայի համեմատ ՍԱՊ-ը առաջարկվող միջոցի դեպքում բարելավվում է մոտ 2 անգամ:

2.2.2. Եռաչափ տրանզիստորներով դոմինո տրամաբանական փականների նախագծման միջոցների մշակումը

Իրենց բարձր արագագործության և փոքր մակերեսի շնորհիվ՝ դոմինո տրամաբանության սխեմաները (ԴՏՍ) լայնորեն կիրառվում են բարձր արագագործություն պահանջող սարքերում: Սակայն արտադրողականության այդ բարձր արժեքն ապահովվում է փոքր աղմկակայունության պաշարի հաշվին:

Կիսահաղորդչային սարքերի մասշտաբավորմանը զուգընթաց՝ ԻՍ-ում աղմուկների ազդեցությունները դառնում են զգալի, ինչի պատճառով ԴՏՍ սխեմաների նախագծումը էապես բարդանում է: Ստանդարտ սխեմաներում դինամիկ պահպանման հանգույցի լիցքը աղմուկներից և ենթաշեմային հաղորդականությունից պահպանելու համար կիրառում են հետադարձ կապով պաշտպանիչ տրանզիստոր (ՊՏ): Դրա էներգասպառումն ու մակերեսը նվազեցնելու համար ընտրվում են նվազագույն չափերով տրանզիստորներ: Սակայն փորձերը ցույց են տվել, որ ենթամիկրոնային չափերով ԻՍ-երում փոքր ՊՏ-ն անհրաժեշտ կերպով չի կատարում իր գործառույթը , ուստի առաջանում է աղմկակայունության բարձրացման նոր մեթոդների կիրառման կարիք:

Խնդրի լուծման հայտնի ձևերից են փոփոխական շեմային լարմամբ ՄՕԿ պաշտպանիչ տրանզիստորով և երկու ՄՕԿ պաշտպանիչ տրանզիստորներով դոմինո տրամաբանությունները, որոնք, սակայն, չեն ապահովում էներգասպառման անհրաժեշտ մակարդակը:

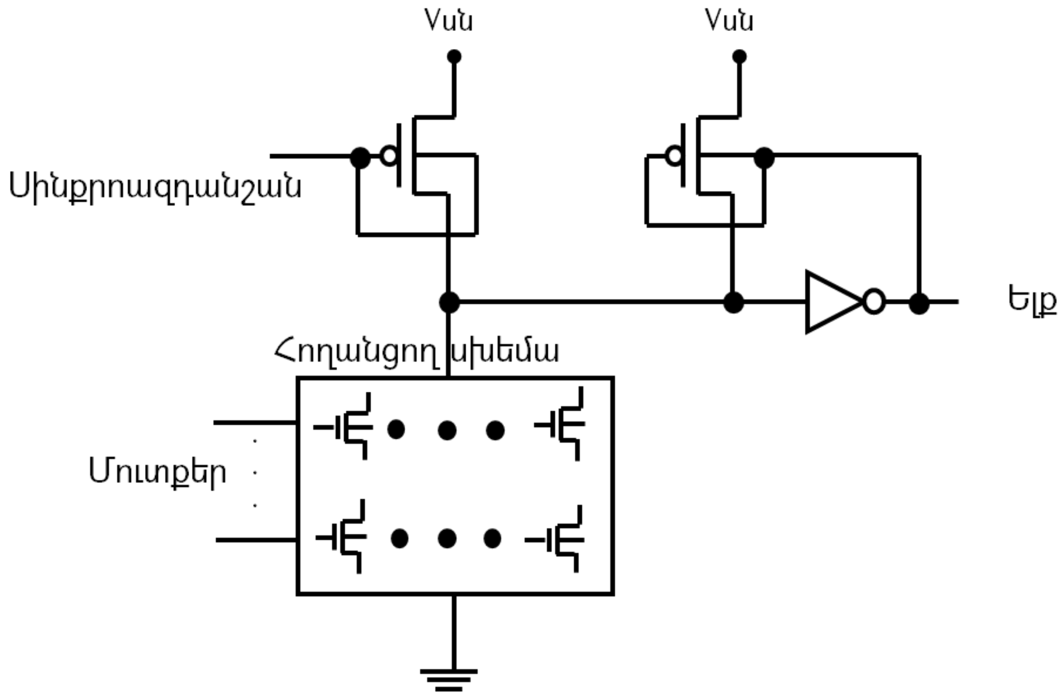
Առաջարկվում է [113] ԱՓԵՏ տրանզիստորների շեմային լարման ղեկավարելիության վրա հիմնված, շեմային լարման փոփոխությամբ դոմինո տրամաբանության դինամիկ հանգույցում պահվող լիցքի պաշտպանման մեթոդ, որն ապահովում է սխեմայի ցածր էներգասպառում և աղմկակայունության անհրաժեշտ մակարդակ: Առաջարկվող մեթոդում շեմային լարման փոփոխությունը տեղի է ունենում ԵՏ-ի ՀՓ-ի անկախ ղեկավարման շնորհիվ: Ուսումնասիրությունները ցույց են տվել, որ այս մեթոդի միջոցով միևնույն աղմկակայունության համար հնարավոր է ապահովել նվազագույն էներգասպառում և հապաղում:

ԻՍ-երում գերարագագործ բոլոր ուղիներն իրականացվում են դոմինո տրամաբանության միջոցով: ԿՓԵՏ միացմամբ ԵՏ-երով դոմինո փականի նախագծումը(նկ. 2.13): Վերջինիս աշխատանքի սկզբունքը հետևյալն է. երբ սինքրոազդանշանը պասսիվ է, սխեման գտնվում է լիցքավորման փուլում, որի ժամանակ լիցքավորման տրանզիստորի միջոցով դինամիկ հանգույցը լիցքավորվում է սնուցման լարում ապահովող լիցքով: Այդ դեպքում ելքային հանգույցն ընդունում է «0» արժեք՝ բացելով պաշտպանիչ տրանզիստորը (ՊՏ): Վերջինս, մինչև սինքրոազդանշանի ակտիվ դառնալը, պահպանում է դինամիկ հանգույցի լիցքը աղմուկներից և ենթաշեմային հաղորդականության հոսանքներից: Աշխատանքի հաջորդ փուլը գնահատման փուլն է, որի ժամանակ, մուտքերից կախված, տեղի է ունենում դինամիկ հանգույցի լրիվ լիցքաթափում: Լիցքաթափում տեղի չունենալու դեպքում ՊՏ-ը կրկին ապահովում է լիցքի կայունությունը:

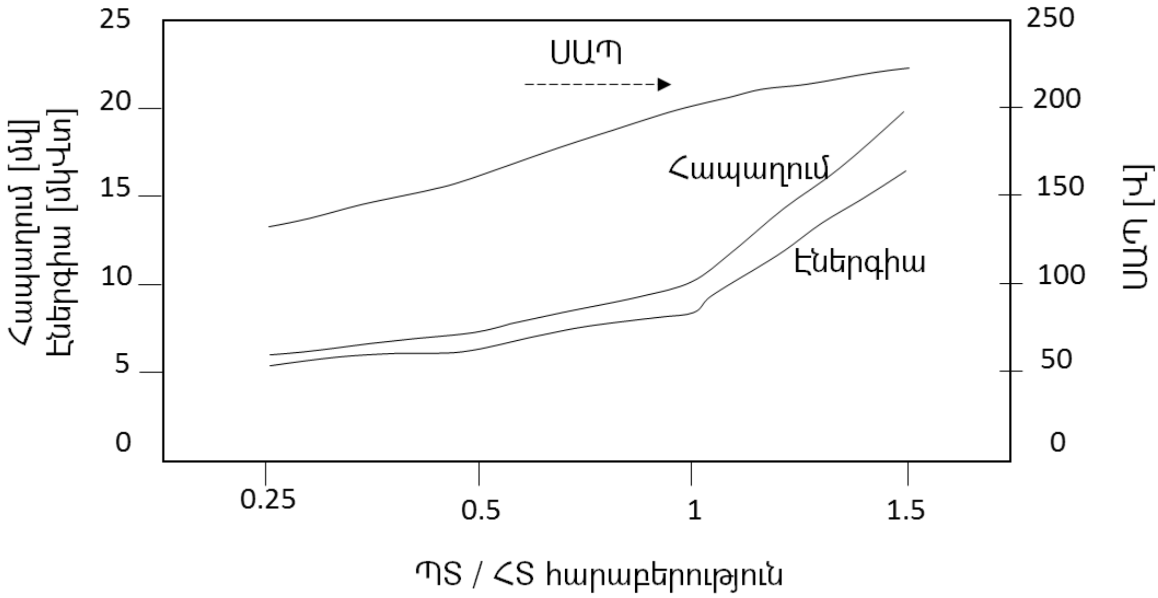
ԿՓԵՏ միացմամբ դոմինո տրամաբանության աղմկակայունության գնահատումն իրականացվել է 16-մուտքանի ԿԱՄ փականի օրինակով: Տրամաբանական «ցածր» մակարդակի աղմկակայունության պաշարը (ՑԱՊ) գնահատվել է այն դեպքի համար, երբ բոլոր մուտքերն ունեն «0» լարում, բացի մեկից, որը փոխանջատվում է: Նաև ենթադրվում է, որ աղմուկները միաժամանակ ազդում են բոլոր մուտքերի վրա:

Նմանակումները ցույց են տվել, որ ՊՏ-ի և nՄՕԿ տրանզիստրոներից մեկի չափերի հարաբերության 0.25-ից 1.25-ի մեծացումը հանգեցնում է ՑԱՊ-ի 70% աճման, միևնույն ժամանակ, 3.3 և 3 անգամ վատացնելով համապատասխանաբար գնահատման փուլի հապաղումն ու էներգասպառումը (նկ. 2.14): Վերոհիշյալից հետևում է, որ դոմինո տրամաբանության աղմկակայունության պաշարի, հապաղման և էներգասպառման անհրաժեշտ արժեքների ապահովման համար նոր մոտեցումների անհրաժեշտություն կա:

Փոփոխական շեմային լարմամբ ՊՏ-ով դինամիկ հանգույցի լիցքի պահպանման մեթոդի հիման վրա մշակված միջոցը պատկերված է նկ. 2.15-ում:



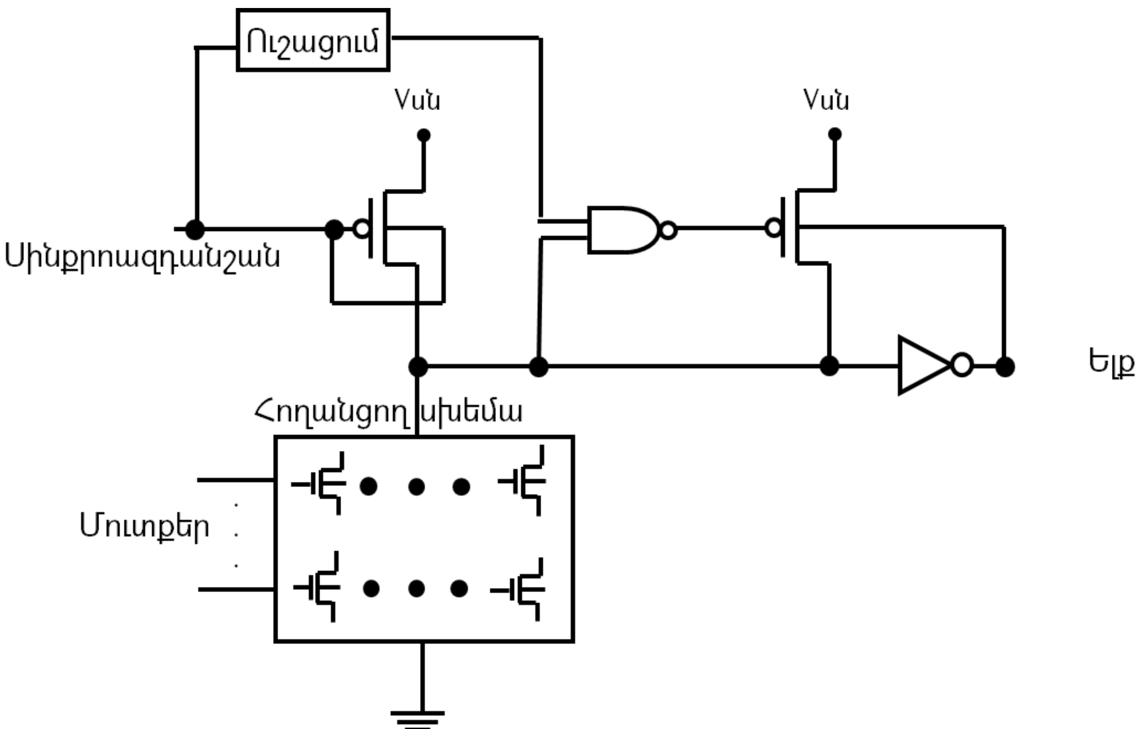
Նկ. 2.13. Դոմինո տրամաբանության՝ ԿՓԵՏ ԵՏ-երով իրականացումը



Նկ. 2.14. ՊՏ և հողանցող տրանզիստորների չափերի հարաբերությունից կախված՝ գնահատման փուլի հապաղման, փականի սպառած էներգիայի և ՍԱՊ-ի կախվածությունը

Դոմինո փականի աշխատանքի սկզբունքը հետևյալն է.

- Լիցքավորման փուլում սինքրոազդանշանը պասիվ է, իսկ հողանցող տրամաբանությունը (ՀՏ)՝ անջատված վիճակում: Դինամիկ ու ելքային հանգույցների լարումներն են համապատասխանաբար «1» և «0»: ՊՏ-ի փականներից մեկը՝ ղեկավարող ազդանշանը, փոխանջատվում է ապաակտիվացնելով այդ փականը: Մյուս փականին կիրառված «0» լարման միջոցով ՊՏ-ը գործում է բարձր շեմային լարմամբ միափական ռեժիմում:
- Գնահատման փուլը սկսվում է, երբ սինքրոազդանշանն ակտիվանում է՝ անջատելով լիցքավորման տրանզիստորը: ՀՏ-ի մուտքերով պայմանավորված՝ դինամիկ հանգույցը լրիվ լիցքաթափվում է: ՊՏ-ը սնուցման աղբյուրից եկող հոսանքի միջոցով փորձում է պահպանել հանգույցի լիցքը, սակայն դրա բարձր շեմային լարման պատճառով այդ հոսանքներն ունեն փոքր արժեքներ, ինչի շնորհիվ ապահովվում են հանգույցի արագ լիցքաթափումը և փոքր դինամիկ էներգասպառումը: Երբ ելքային հանգույցը ձեռք է բերում «1» լարում, ՊՏ-ն ամբողջությամբ անջատվում է:



Նկ. 2.15. Պաշտպանիչ տրանզիստորի առաջարկվող մեթոդի իրականացումը

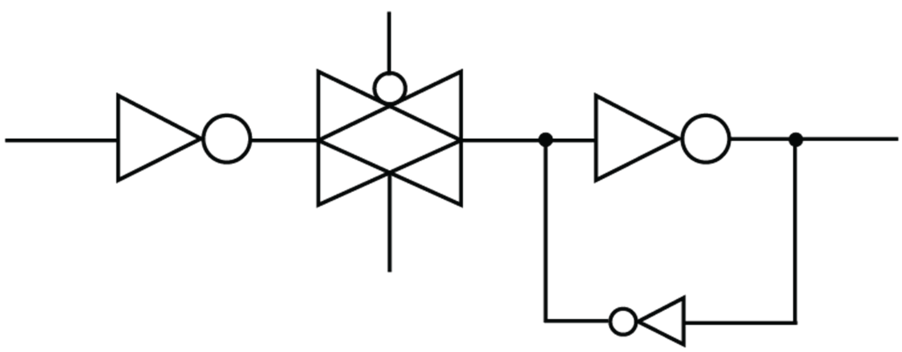
Այն դեպքում, երբ ՀՏ-ի մուտքային կոմբինացիայի համաձայն՝ դինամիկ հանգույցի լիցքը պետք է պահպանվի, ԵՎ-ՈՉ տարրի ելքային ազդանշանն ընդունում է «0» արժեքը՝ ՊՏ-ը տեղափոխելով ցածր շեմային լարմամբ ռեժիմ՝ ապահովելով լիցքի լիարժեք պահպանում:

Շեմային լարման փոփոխականության շնորհիվ հնարավոր է դառնում ապահովել բարձր աղմկակայունություն՝ միաժամանակ պահպանելով արագագործության ու էներգախնայողության անհրաժեշտ մակարդակ:

2.2.3. Եռաչափ տրանզիստորներով ստատիկ տրիգերների նախագծման միջոցի մշակումը

Ստատիկ տրիգերները սինքրոն սխեմաներում [114] լայնորեն կիրառվող սխեմաներից են: Դրանց հիմնական բաղկացուցիչ մասն են կազմում իրար միացված շրջիչները: Տվյալային կարգը այս տրիգերների մեջ գրանցվում է, մուտքային ավելի ուժեղ տրանզիստորների միջոցով, ուժային մեթոդով (ՈՒՄ), որն օգտագործում է սինքրոազդանշանով ղեկավարվող փոխանցման փական: ՈՒՄ-ի կիրառման դեպքում տեղի է ունենում սպառված էներգիայի և զբաղեցրած մակերեսի զգալի նվազում: Այդ պատճառով դրանք ակտիվորեն կիրառվում են ժամանակակից ԻՍ-երում:

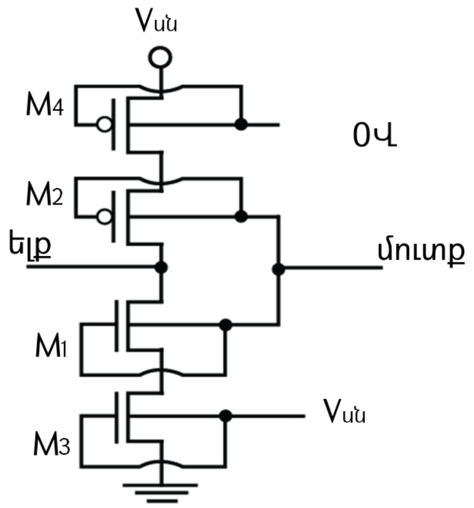
ԵՏ-երով նախագծված ՈՒՄ տրիգերում (նկ. 2.16) բոլոր տրանզիստորներն ունեն ԿՓԵՏ միացում: Որպեսզի հնարավոր լինի սարքում պահվող տվյալային կարգը փոփոխել, Նկ. 2.16-ում պատկերված հետադարձ կապի շրջիչը պետք է լինի ավելի թույլ, քան տրիգերի մուտքային կասկադը:



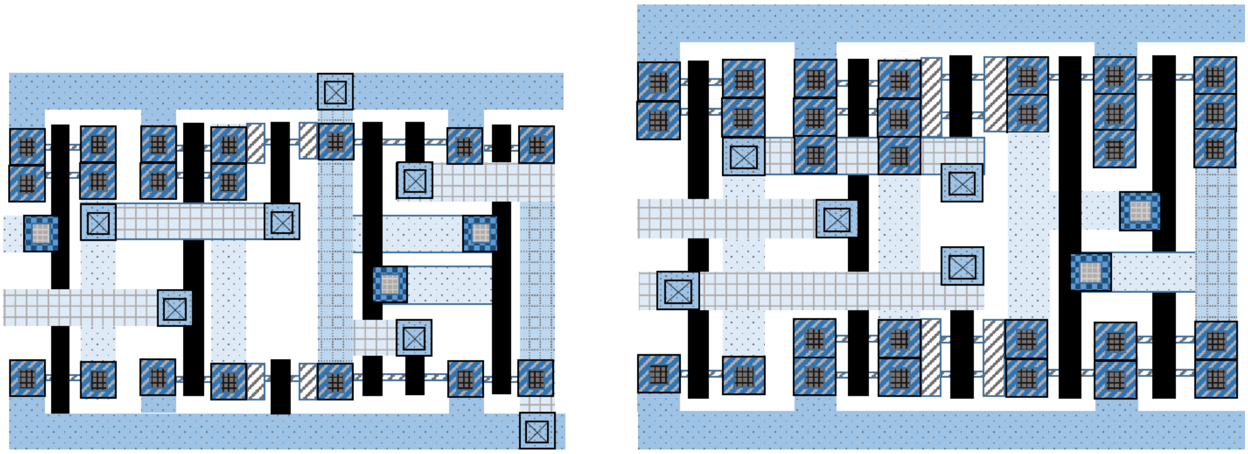
Նկ. 2.16. Դասական ստատիկ տրիգերը

Հաստատուն շեմային լարմամբ ԿՓԵՏ սարքերով հետադարձ կապի շրջիչի իրականացման տարբերակներից մեկում (նկ. 2.17) կիրառվում են նվազագույն չափերով և փոքր շեմային լարմամբ տրանզիստորներ: Այդ շրջիչը թուլացվում է հողանցման և սնուցման տրանզիստորային ցանցերի դիմադրությունների մեծացման միջոցով: Մուտքային շրջիչն ու փոխանցման փականը նախագծվում են այնպես, որ ավելի ուժեղ լինեն հետադարձ կապի շրջիչից (ՀԿՇ):

ՀԿՇ-ի մեկ այլ իրականցման դեպքում դրա տրանզիստորները դարձյալ ունեն նվազագույն չափ և փոքր շեմային լարում: Մուտքային կասկադի չափերի՝ ՀԿՇ-ից կրկնակի մեծ լինելու դեպքում կապահովվի տրիգերի անհրաժեշտ ֆունկցիոնալությունը: Ի տարբերություն նախորդ տարբերակի՝ այս տարբերակն զբաղեցնում է ավելի մեծ մակերես և սպառում համեմատաբար մեծ էներգիա: ՀԿՇ-ի նշված երկու իրականացումների ֆիզիկական նախագծերը պատկերված են նկ. 2.18-ում:

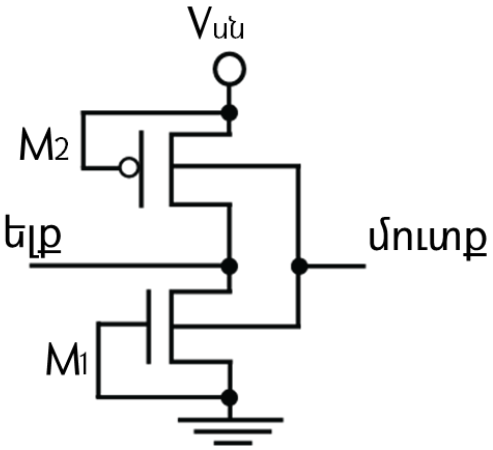


Նկ. 2.17. Նվազագույն չափերով ԵՏ-երով հետադարձ կապի շրջիչը



Նկ. 2.18. ՀԿՇ-ի երկու իրականացումների ֆիզիկական նախագծերը

Առաջարկվում է [115] բազմաշեմային լարումներով ԵՏ-երով տրիգերների նախագծման նոր միջոց, որը հիմնված է ԱՓԵՏ տրանզիստորների առավելությունների վրա: Մուտքային շրջիչն ու փոխանցման փականն ունեն նվազագույն չափեր: ՀԿՇ-ի (նկ. 2.19) նվազագույն չափ ունեցող տրանզիստորները թուլացվում են դրանց միափական աշխատանքի միջոցով:



Նկ. 2.19. Հետադարձ կապի շրջիչի մեկ այլ իրականացումը

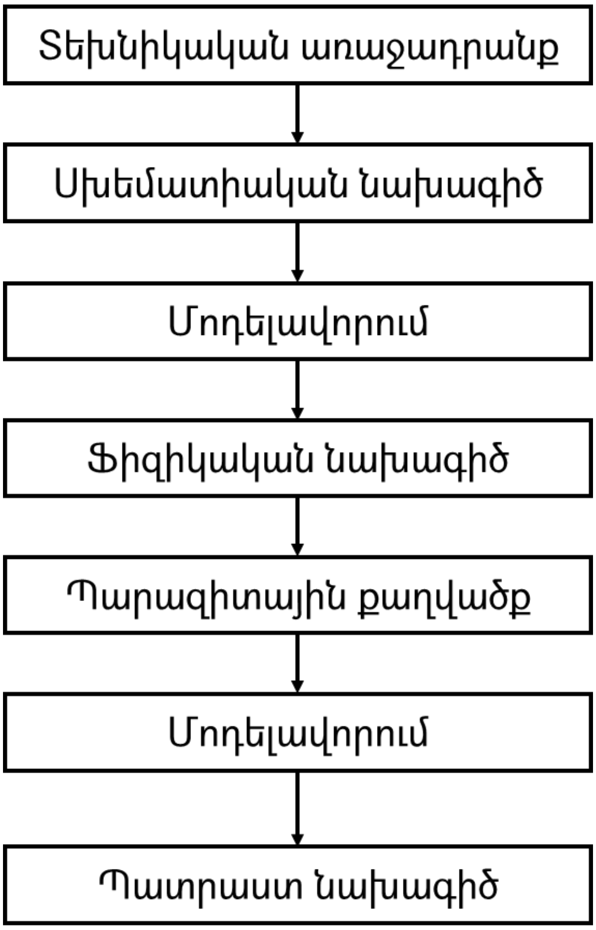
Առանց դրանց չափերի մեծացման անհրաժեշտության՝ մուտքային շրջիչն ու փոխանցման փականն առաջացնում են ավելի մեծ հոսանք, քան ՀԿՇ-ը: Այսպիսով իրականացման դեպքում տեղի է ունենում էներգասպառման և զբաղեցրած մակերեսի էական նվազում՝ միաժամանակ պահպանվելով նույն արագագործությունը, ինչ ԿՓԵՏ միացմամբ տրանզիստորներով ՀԿՇ-ի դեպքում:

2.3. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների կիրառմամբ նախագծման գործընթացը

Եռաչափ և ՄՕԿ տրանզիստորների կառուցվածքային տարբերությունների պատճառով ԵՏ-երով սխեմաների նախագծումն ունի սկզբունքային առանձնահատկություններ: Ստորև նկարագրվում է ԵՏ-երով միկրոէլեկտրոնային սխեմաների նախագծման գործընթացը:

ԵՏ-երով ԹՍՏԳ բջիջների և ՀԲ-երի նախագծումն իրականացվում է նախագծման պատվիրովի երթուղու (ՆՊԵ) [116] միջոցով(նկ. 2.20):

ԵՏ-երով սխեմատեխնիկական նախագծման փուլը ԿՄՕԿ տրանզիստորներով դեպքից տարբերվում է ԵՏ-երի պարամետրերով: Եռաչափ տրանզիստորի նմանակման մոդելում [117] նախապես ամրագրված են պարամետրեր (աղ. 2.4), որոնցից շատերը գոյություն ունեն միայն այս սարքերում: Այդ և որոշ այլ պարամետրեր (աղ. 2.5) նախատեսված են օգտագործողի կողմից փոփոխվելու համար (նկ. 2.21):



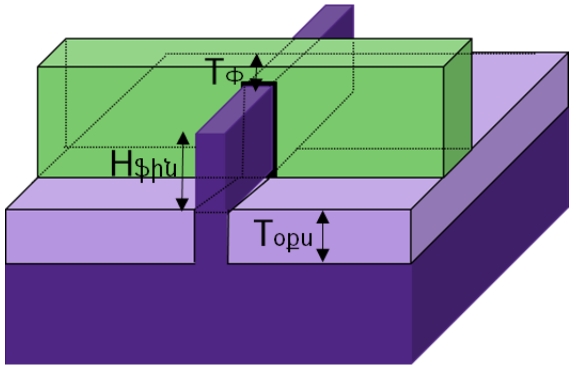
Նկ. 2.20. Պատվիրովի նախագծման երթուղին

Աղյուսակ 2.4

ԵՏ-երի նմանակման մոդելի պարամետրերի ցուցակը

Պարամետրի անունը	**Պարամետրի նշանակությունը**
$L_{Հ}$	Հոսքուղու երկարությունը
$H_{ֆին}$	Ֆին շերտի բարձրությունը
$T_{ֆին}$	Ֆին շերտի հաստությունը
$T_{օքսիդ}$	Օքսիդի հաստությունը

$T_{\text{Փ}}$	Փականի հաստությունը



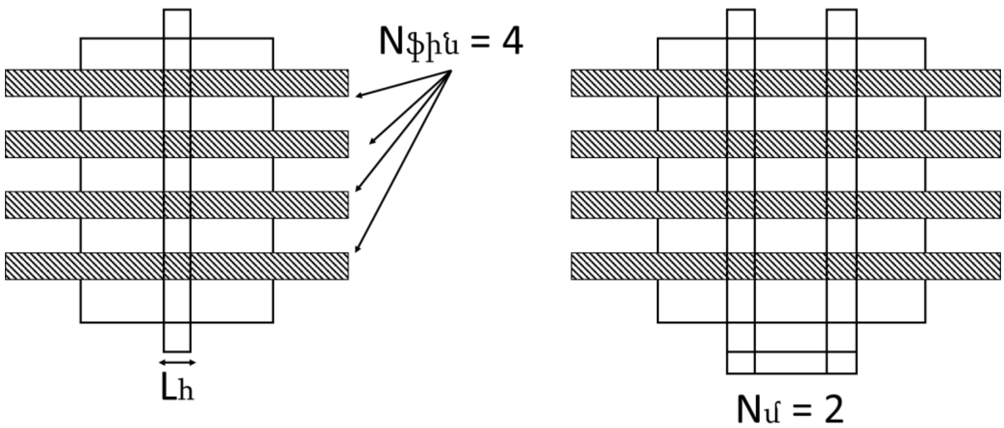
Նկ. 2.21. Եռաչափ տրանզիստորի պարամետրերը

Աղյուսակ 2.5

ԵՏ-երի՝ նախագծման ընթացքում օգտագործվող պարամետրերը

Պարամետրի անունը	**Պարամետրի նշանակությունը**
$N_{\text{ֆին}}$	Ֆիների քանակը
m	Տրանզիստորի բազմապատկիչը
$N_{\text{մ}}$	Տրանզիստորի մատների քանակը
$L_{\text{Հ}}$	Հոսքուղու երկարությունը

Սխեմատեխնիկական նախագիծն ավարտելուց հետո տրանզիստորային մակարդակի մոդելավորման ծրագրային գործիքով իրականացվում է նախագծի մոդելավորում և ստուգվում է դրա՝ տեխնիկական առաջադրանքին համապատասխանությունը: Անհրաժեշտ պայմանները բավարարող սխեմայի հիման վրա նախագծվում է դրա ֆիզիկական տեսքը [118] (նկ. 2.22):



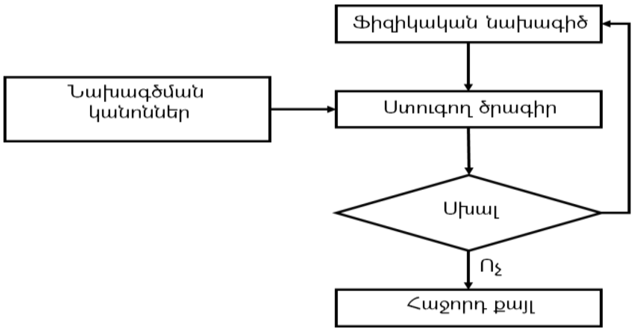
Նկ. 2.22. ԵՏ-երի՝ ֆիզիկական նախագծման ընթացքում օգտագործվող պարամետրերը

Չնայած ԵՏ-երի և ՄՕԿ տրանզիստորների կառուցվածքային էական տարբերություններին՝ դրանց ֆիզիկական երկչափ տեսքերը էական տարբերություններ չունեն: Ֆիզիկական նախագիծը կազմված է տարբեր կիսահաղորդչային և մետաղական շերտերից, որոնք ներկայացնում են արտադրության մեջ օգտագործվող համապատասխան շերտերը, դրանց դիրքերը և չափերը: Տվյալ արտադրական գործընթացով օժանդակվող շերտերի ցուցակը տրվում է ԻՍ-ն արտադրող գործարանի կողմից գործընթացի նախագծման հավաքածուում (ԳՆՀ):

Ֆիզիկական նախագիծը (ՖՆ) անցնում է ստուգման երկու փուլ՝ նախագծման կանոնների ստուգում և սխեմատեխնիկական նախագծի հետ կառուցվածքային համեմատություն:

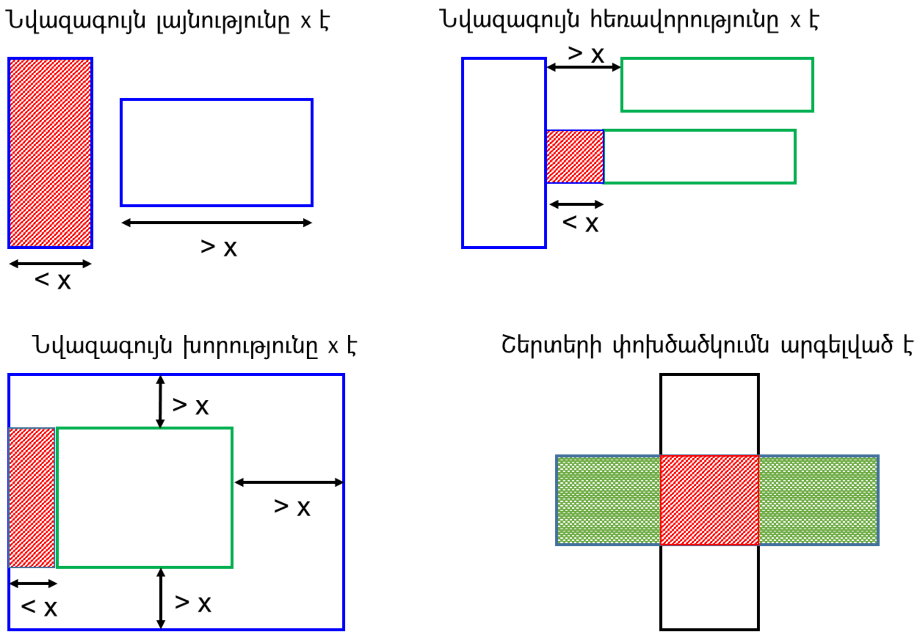
Նախագծման կանոնները [119] ԻՍ-եր արտադրող գործարանի կողմից սահմանված, ֆիզիկական նախագծի մեջ օգտագործվող շերտերի երկրաչափական չափերը և դրանց՝ միմյանց հետ փոխհարաբերությունները կարգավորող օրենքներ են, որոնք ընդհանրացված են նախագծման կանոնների ուղեցույցում (ՆԿՈՒ): Դրանք համապատասխանում են ԻՍ-ի արտադրության ժամանակ կիրառվող ֆոտոշաբլոնների պատուհանների չափերին և վերջիններիս փոխհարաբերություններին, այդ պատճառով ՖՆ-ի՝ նախագծման անգամ մեկ կանոնին չբավարարելը անթույլատրելի է, և այդ նախագիծը ենթակա չէ արտադրության:

Նախագծման կանոնների հիմնական տեսակներն են՝ հեռավորության, լայնության, փոխծածկման և այլ փոխհարաբերությունների ստուգումները: Նախագծման կանոններով ստուգումը (նկ. 2.23) ձևայնացված ստուգում է, այսինքն՝ ստուգվում է նախագծի երկրաչափությունը, այլ ոչ թե ֆունկցիոնալությունը (նկ. 2.24):



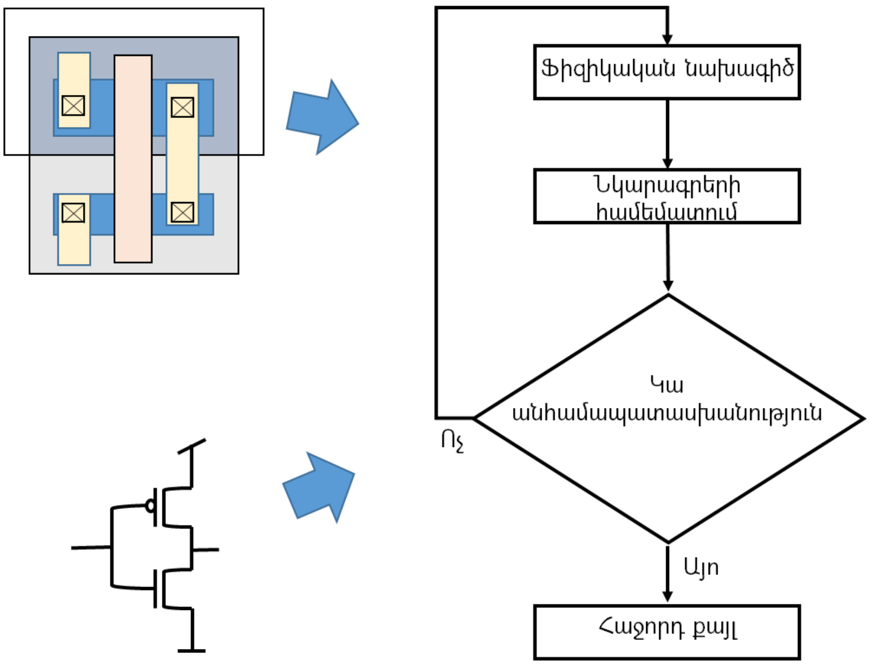
Նկ. 2.23. Ֆիզիկական նախագծման կանոնների ստուգման ընթացակարգը

Այդ ստուգումն իրականացնող ծրագրային գործիքը որպես ելքային արդյունք վերադարձնում է ՖՆ-ի բոլոր այն բազմանկյունները, որոնք չեն բավարարել դրանց շերտերին առաջադրված նախագծման կանոններին:



Նկ. 2.24. Նախագծման կանոնների օրինակները

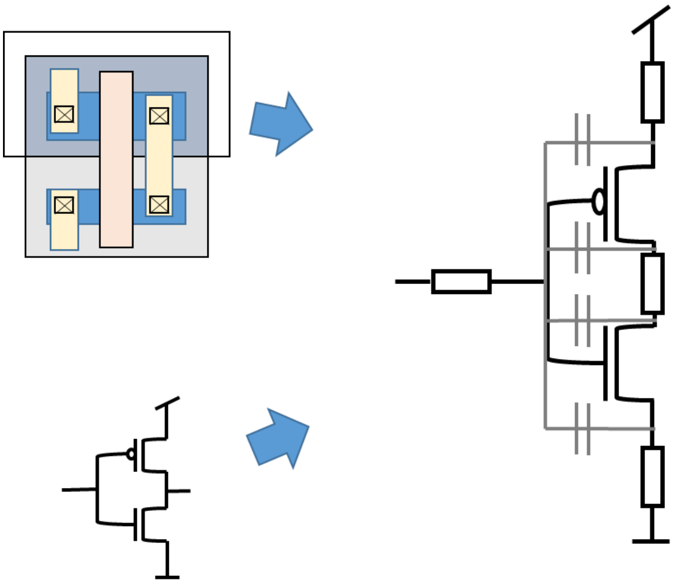
Նախագծման կանոններին ՖՆ-ի համապատասխանությունը ստուգելուց հետո իրականացվում է երկրորդ ձևայնացված ստուգումը, որը համեմատում է ՖՆ-ը դրան համապատասխանող սխեմատեխնիկական նախագծի հետ (նկ. 2.25):



Նկ. 2.25. ՖՆ-ի և սխեմատեխնիկական նախագծի համեմատման ընթացակարգը

Ծրագրային գործիքը ՖՆ-ում նույնականացնում է կիսահաղորդչային սարքերը, դրանց միջմիացումները, երկրաչափական չափերը և համեմատում է սխեմատեխնիկական նախագծում համապատասխան նկարագրերի հետ: Սխեմատեխնիկական նախագծի և ՖՆ-ի համապատասխանությունը հավաստելուց հետո նախագիծը մտնում է պարազիտային քաղվածքի փուլ:

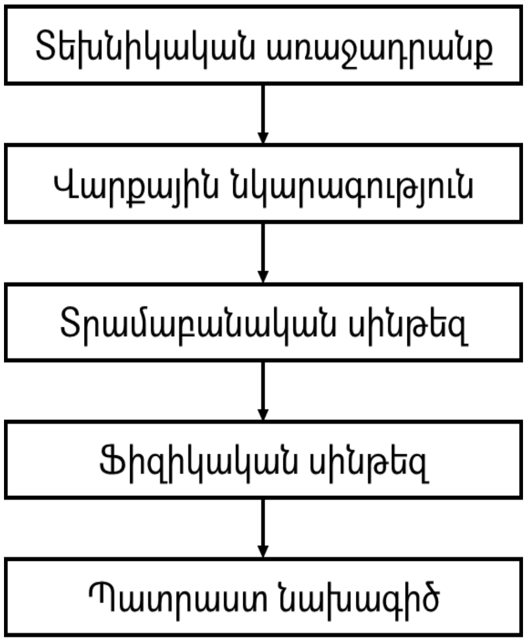
Պարազիտային քաղվածքը (ՊՔ) [120] գործընթաց է (նկ. 2.26), որի ընթացքում տեղի է ունենում կիսահաղորդչային սարքերում և միջմիացումների միջև առկա պարազիտային ունակությունների, դիմադրությունների և ինդուկտիվությունների հաշվարկ:



Նկ. 2.26. Պարազիտային քաղվածքի էությունը

ՊՔ-ի հիմնական նպատակն է ստանալ նախագծված սխեմայի՝ իրականին մոտ մոդելը և տրանզիստորային մակարդակի մոդելավորման միջոցով ստուգել դրա համապատասխանությունը տեխնիկական առաջադրանքին: Հարկ է նշել, որ տեխնոլոգիական մասշտաբավորման հետևանքով պարազիտային տարրերի ազեցությունները սխեմայի վրա դառնում են ավելի շոշափելի: Եթե պարազիտային տարրերի՝ սխեմայում ներառումից հետո վերջինս բավարարում է տեխնիկական առաջադրանքի պահանջները, ապա նախագիծը պատրաստ է: Ավարտուն նախագիծը նախագծի տարբեր տեսքերը ներկայացնող ֆայլերի հավաքածու է: Մասնավորապես, սխեմատեխնիկական և ֆիզիկական տեսքերը ներկայացվում են համապատասխանաբար SPICE և GDSII ձևաչափերով:

ՆՊԵ-ով նախագծվում են ԹՏՏԳ տարրերը, խառը ազդանշանային սխեմաները [116] և ՀԲ-երը: Սակայն միլիարդավոր ԵՏ-եր պարունակող ԻՍ-երը ստեղծվում են վարքային նախագծից ֆիզիկական նախագծի սինթեզման միջոցով: Երթուղին, որով այդ գործընթացն իրականացվում է, կոչվում է ավտոմատացված նախագծման երթուղի (ԱՆԵ) [121], որի փուլերը ներկայացնող բլոկ-դիագրամը բերված է նկ. 2.27

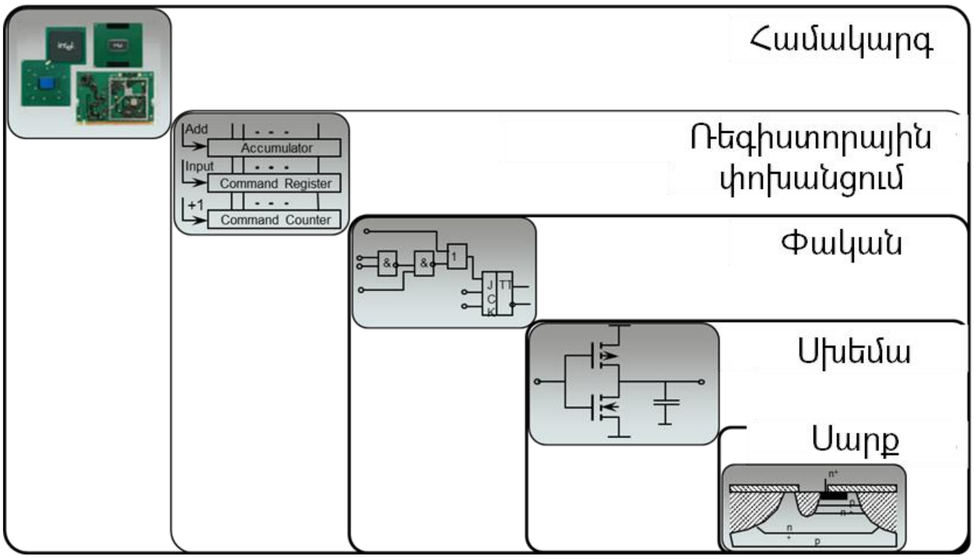


Նկ. 2.27. Ավտոմատացված նախագծման երթուղին

ԱՆԵ-ի իրականացման համար անհրաժեշտ է ունենալ ԹՍՏԳ, որում առկա տարրերը օգտագործվելու են համակարգի սինթեզման համար:

Վարքային նկարագրությունը կամ ռեգիստրային փոխանցման մակարդակը (ՌՓՄ), սարքավորումների նկարագրման որևէ լեզվով (ՍՆԼ) գրված, ԻՍ-ի վարքի նկարագրությունն է: Այն նախագծման գործընթացի վերացարկման բարձր մակարդակն է (նկ. 2.28), որում սխեմայի նկարագրությունը չի պարունակում դրա կառուցվածքի մասին տվյալներ: ՌՓՄ նկարագրության մեջ ընգրկվում են տվյալային կարգեր փոխանակող ռեգիստրները և այդ փոխանակումների իրականացման համար անհրաժեշտ պայմանները:

ԻՍ-երի ՌՓՄ ներակայացման համար կիրառվում են Verilog [122] կամ VHDL [119] ՍՆԼ-երը: ՌՓՄ նկարագրությունը տրվում է տրամաբանական սինթեզ (ՏՍ) իրականացնող ծրագրային գործիքի մուտքին, որն էլ գեներացնում է համապատասխան Verilog կամ VHDL կառուցվածքային՝ փականային մակարդակի (ՓՄ) նկարագրությունը (նկ. 2.29):



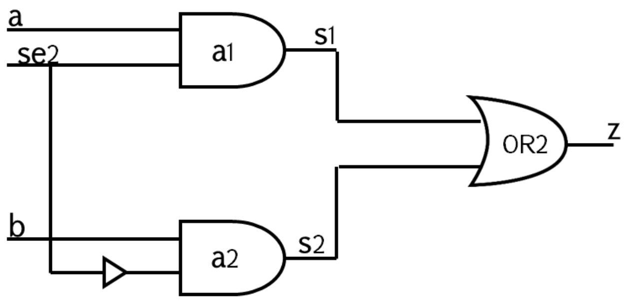
Նկ. 2.28. Նախագծման գործընթացի վերացարկման մակարդակները

ՌՓՄ նկարագրություն	ՓՄ նկարագրություն

```
module mux(out, in1, in2, sel);

output reg out;
input in1, in2, sel;

always @(in1 or in2 or sel) begin
        if (sel) out <= in1;
        elseif (!sel)out <= in2;
end
```



Նկ. 2.29. Վարքային և փականային նկարագրերի կապը

ՏՍ կատարող ծրագրային գործիքը, իր մուտքին ստանալով սխեմայի ՌՓՄ նկարագրությունը, սինթեզված սխեմային առաջադրվող սահմանափակումները և ԹՏՏԳ-ը, վերջինում առկա տարրերից ընտրում է նրանք, որոնց հապաղումը, մակերեսն ու էներգասպառումը բավարարում են առաջադրված պահանջները: Արդյունքում՝ ստացված ՓՄ սխեման տրվում է մեկ այլ ծրագրային գործիքի, որը, մուտքին ստանալով ՓՄ նկարագրություն, ԹՏՏԳ և ՖՆ-ին առաջադրվող սահմանափակումներ, իրականացնում է վերջինիս սինթեզը: ԱՆԵ-ի վերջին փուլում ստուգվում է ՖՆ-ի՝ տեխնիկական առաջադրանքին բավարարելը, որի դրական արդյունքից հետո նախագիծը համարվում է պատրաստ:

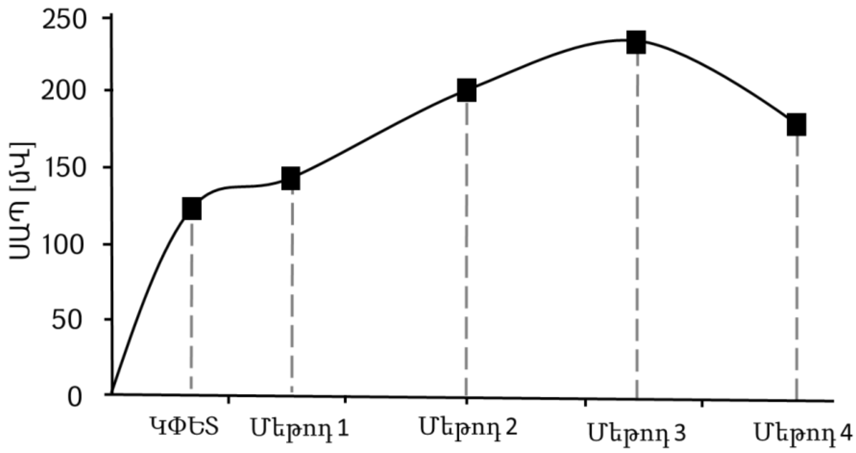
2.4. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման առաջարկվող միջոցների արդյունավետության գնահատումը

§2.2. -ում առաջարկված նախագծման միջոցներն իրականացվել են 14 *նմ* տեխնոլոգիական գործընթացի եռաչափ տրանզիստորներով, և կատարվել է ստացված արդյունքների համեմատում:

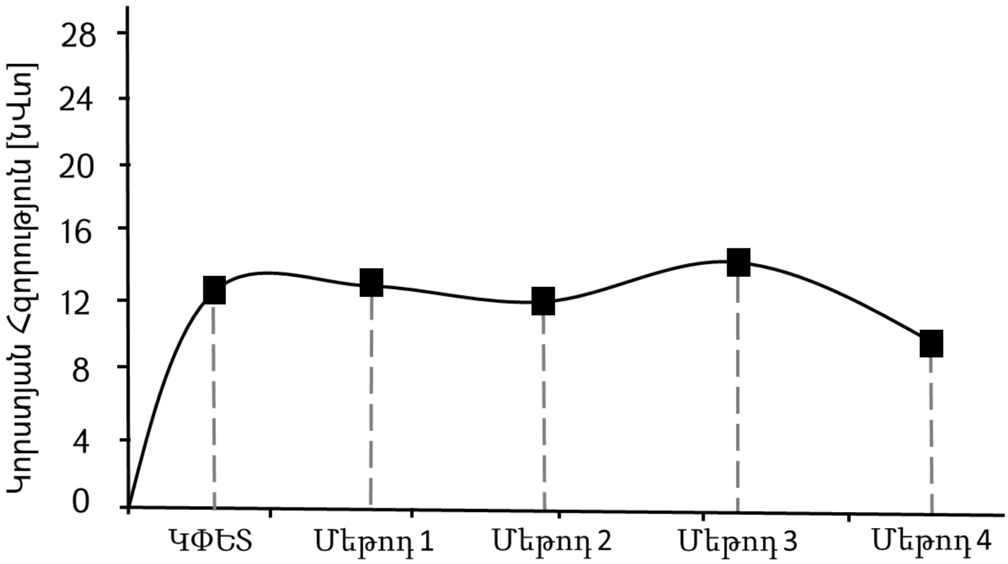
Հետազոտվել են առաջարկված 4 ՍՀԲ-երի ՍԱՊ-ը, կորստի էներգասպառումը և ԸԳ-ի հոսանքը:

<u>ՍԱՊ:</u> ԿՓԵՏ միացմամբ ՀԲ-ի հետ համեմատած՝ նկ. 2.8, 2.9 և 2.10-ում պատկերված մեթոդներն ապահովում են ՍԱՊ-ի արժեքների էական մեծացում (նկ. 2.30):

<u>Կորստի էներգասպառում:</u> Այս մեծությունը որոշվում է ՀԲ-ի տրանզիստորների արդյունարար լայնությամբ և դրանց շեմային լարումներով: ՍԱՊ-ի մեծացման համար բջջի չափերի ավելացումը հանգեցնում է էական կորստային հոսանքների: Առաջարկվող 4 նախագծման միջոցներում տրանզիստորների չափերը նվազագույնն են, իսկ կորստի էներգասպառումը համապատասխանաբար 15,38%, 22,95%, 2,74% և 50% -ներով փոքր է ԿՓԵՏ միացմամբ դեպքի համեմատ (նկ. 2.31):

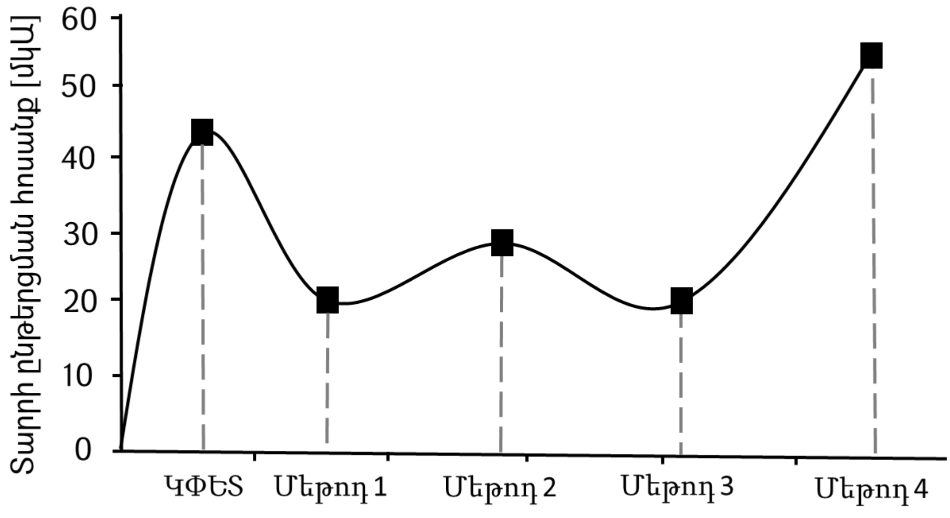


Նկ. 2.30. Դիտարկված նախագծման միջոցների համար ՍԱՊ-ի արժեքները



Նկ. 2.31. Դիտարկված նախագծման միջոցների համար կորստի հզորության արժեքները

ԸԳ-ի հոսանք: Ընթերցման գործողության հոսանքը սահմանվում է որպես ԸԳ-ի ժամանակ ԿԳ-ի հոսող առավելագույն հոսանքը: Նկ. 2.8-ում և 2.10-ում պատկերված միջոցներում, միափական ռեժիմում գործող թույլ հասանելիության տրանզիստորների շնորհիվ, ընթերցման հոսանքը ԿՓԵՏ սխեմայի համեմատ նվազել է մոտ 53,3%-ով, իսկ նկ. 2.9-ի դեպքում, P տիպի հասանելիության տրանզիստորների շնորհիվ, հոսանքը նվազել է 33,3%-ով: Երկփական ռեժիմում աշխատող այդ տրանզիստորները ստեղծում են ավելի մեծ հոսանք, քան մյուս երկու նախագծման միջոցներում N տիպի տրանզիստորները, որի հետևանքով առաջանում է այս տարբերությունը (նկ. 2.32):



Նկ. 2.32. Դիտարկված նախագծման միջոցների համար ընթերցման հոսանքի մեծությունը

Առաջարկվել են ԱՓԵՏ սարքերով ՍՀԲ-երի նախագծման միջոցներ և դրանց հիման վրա մշակվել են 6Տ կառուցվածքով բջիջներ: Վերջիններս նախագծման մյուս հայտնի միջոցների համեմատ ցուցաբերել են միջինում ՍԱՊ-ի 44,4% և ստատիկ էներգասպառման 7,7% բարելավում: ՀԲ-երի նախագծման ներկառուցված հետադարձ կապով միջոցի դեպքում գրանցվել է 36% ՍԱՊ-ի և 25% ստատիկ էներգասպառման բարելավում: Վերոնշյալ ընդհանրացված արդյունքները բերված են աղ. 2.6:

Աղյուսակ 2.6

ՀԲ-երի նախագծման առաջարկված միջոցների մոդելավորման արդյունքները

Հիշողության բջիջը	ՍԱՊ (նՎ)	Կորստի էներգասպառում (նՎտ)	Ընթերցման հոսանք (մկԱ)
Մեթոդ 1	148	13	21,3
Մեթոդ 2	200	12,2	29,8
Մեթոդ 3	237	14,6	20,5
Մեթոդ 4	156	10	55

Դոմինո տրամաբանության լիցքի պահպանման առաջարկված միջոցը, պաշտպանման տրանզիստորի տարբեր չափերի կիրառմամբ, ստուգվել է 4-մուտքանի ԿԱՄ, 16-մուտքանի ԿԱՄ և 16-մուտքանի մուլտիպլեքսորի օրինակների միջոցով: ԿՓԵՏ տրանզիստորներով կառուցված դոմինո տրամաբանության համեմատ, միևնույն ցածր մակարդակի աղմկակայունության պաշարի դեպքում, առաջարկված մեթոդը նվազեցրել է գնահատման հապաղումը 4-մուտքանի ԿԱՄ տարրի դեպքում 36,7%, 16-մուտքանի ԿԱՄ-ի դեպքում 30,27% և 16-մուտքանի մուլտիպլեքսորի դեպքում 21,4%: Հզորության սպառումը վերոհիշյալ սխեմաների դեպքում նվազեցվել է համապատասխանաբար 32,4%, 46,3% և 15% (աղ. 2.7):

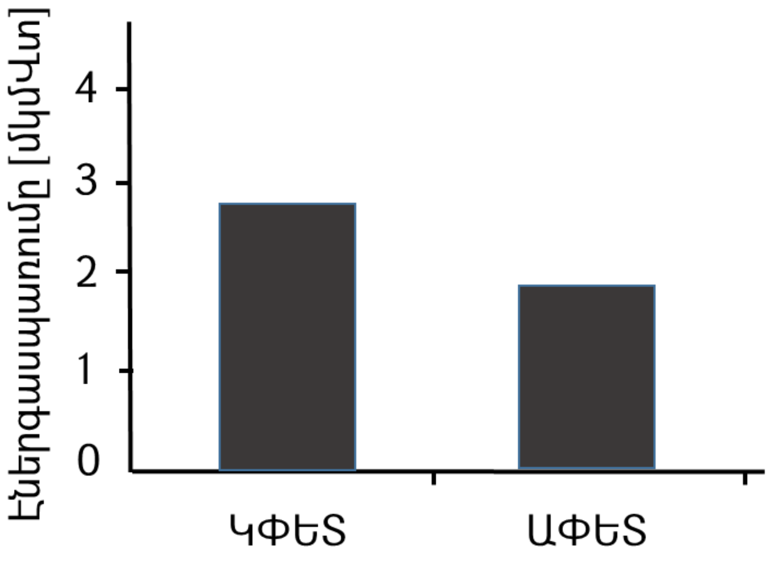
Ստատիկ տրիգերների մոդելավորման ժամանակ ուսումնասիրված սարքերի ելքին միացվել է 0.5 *ֆՖ* ունակային բեռ: Սինքրոազդանշանի սպառած էներգիան չափվել է այն ժամանակ, երբ սարքի մուտքն ու ելքն ունեցել են 0 *Վ* արժեք, և միակ փոխանջատվող ազդանշանը հենց տակտավորող ազդանշանն է եղել:

Աղյուսակ 2.7

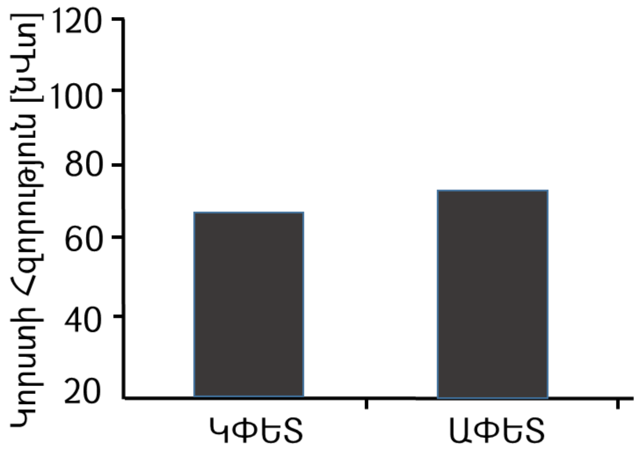
Լիցքի պաշիանման մեթոդով տարբեր փականների մոդելավորումների արդյունքները

Տրամաբանական փականը	ԿՓԵՏ տրանզիստորներով իրականուցումը		Առաջարկվող մեթոդով իրականացումը	
	Էներգասպառում (մկմՎտ)	Հապաղում (պվ)	Էներգասպառում (մկՎտ)	Հապաղում (պվ)
4-մուտքանի ԿԱՄ	7,1	9,5	4,8	6,96
16-մուտքանի ԿԱՄ	8,45	11	4,54	7,67
16-մուտքանի մուլտիպլեքսոր	16,8	8,4	14,3	6,6

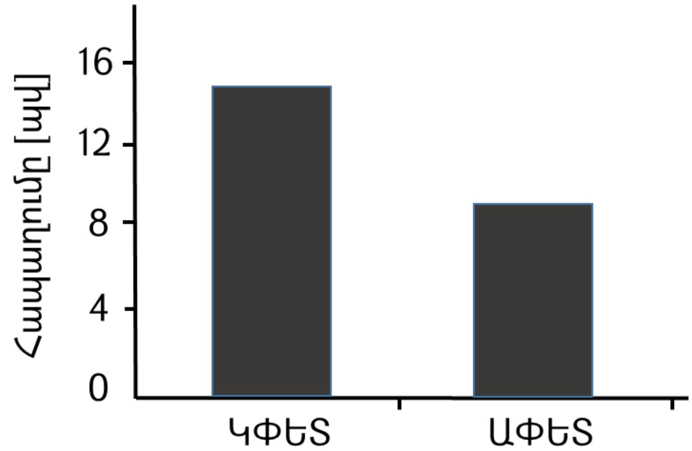
Մոդելավորումների համար առաջարկվող մեթոդի հիման վրա իրագործված տրիգերը համեմատվել է ստանդարտ ԿՓԵՏ միացմամբ տրանզիստորներով տրիգերի հետ: Դիտարկվել են սարքերի հետևյալ բնութագրերը՝ էներգասպառումը, ազդանշանի տարածման հապաղումը, հաստատման ժամանակը և ստատիկ աղմկակայունության պաշարը (նկ. 2.33-2.37):



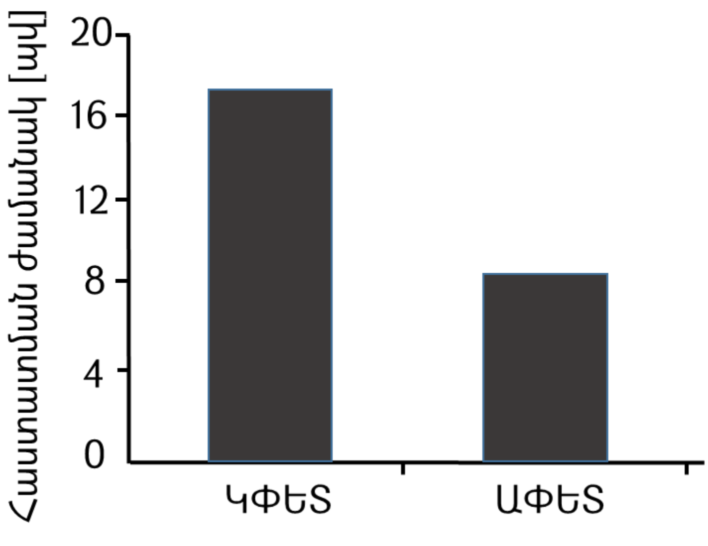
Նկ. 2.33. Դիտարկված տրիգերների էներգասպառումը



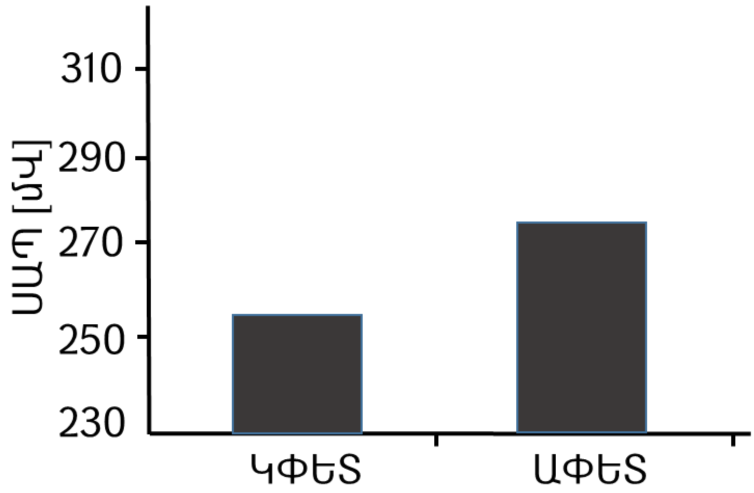
Նկ. 2.34. Դիտարկված տրիգերների կորստի միջինացված էներգասպառումը



Նկ. 2.35. Դիտարկված տրիգերների միջին տարածման հապաղումը



Նկ. 2.36. Դիտարկված տրիգերների հաստատման ժամանակները



Նկ. 2.37. Դիտարկված տրիգերների ՍԱՊ-երի արժեքները

Այսպիսով, եռաչափ տրանզիստորների անկախ փականային միացման առավելությունների շնորհիվ՝ մշակված նախագծման միջոցները, կարճ միացված փականներով կառուցվածքի համեմատ, ապահովում են ստատիկ հիշողության բջիջների, դոմինո տրամաբանական փականների և ստատիկ տրիգերների հիմնական պարամետրերի շահում՝ կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի և նախագծման գործընթացի բարդության ոչ էական ավելացման հաշվին:

Եզրակացություններ

1. Առաջարկվել է եռաչափ տրանզիստորներով ստատիկ հիշողության բջիջների նախագծման 4 մեթոդ, որոնք, օգտագործելով դրանց անկախ փականներ ունենալու հատկությունը, հայտնի լուծումների համեմատ ապահովում են բջջի ստատիկ աղմկակայունության պաշարի մեծ արժեք և փոքր ստատիկ էներգասպառում՝ կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի ոչ էական ավելացման հաշվին:
2. Մշակվել է դոմինո տրամաբանության դինամիկ հանգույցում լիցքի պաշտպանման միջոց, որը, փականի զբաղեցրած մակերեսի մեծացման հաշվին, առկա մոտեցումների համեմատ ապահովում է էապես ցածր էներգասպառում և անհրաժեշտ բարձր արագագործություն:

3. Ստեղծվել է ստատիկ տրիգերների՝ էներգասպառման չնչին ավելացման հաշվին ստատիկ աղմկակայունության պաշարը և արագագործությունը բարելավող մեթոդ, որի հիման վրա տրիգերի հետադարձ կապի շրջիչն իրականացվել է անկախ փականներով եռաչափ տրանզիստորներով:
4. Առաջարկված մեթոդների արդյունավետության վերլուծությունը ցույց է տվել, որ ԻՍ-երի որոշ մակրոբնութագրերի աննշան վատթարացման հաշվին էապես բարելավվում են մյուս բնութագրերը՝ վկայելով մշակված մեթոդների և միջոցների բարձր արդյունավետության մասին:

ԳԼՈՒԽ 3. ԵՌԱՉԱՓ ՏՐԱՆԶԻՍՏՈՐՆԵՐՈՎ ՄԻԿՐՈԷԼԵԿՏՐՈՆԱՅԻՆ ՍԽԵՄԱՆԵՐԻ ՆԱԽԱԳԾՄԱՆ ՄԻՋՈՑՆԵՐԻ ԻՐԱԿԱՆԱՑՈՒՄԸ

ԻՍ-երի մշակման ժամանակ դրանց ենթահանգույցները ձգտում են նախագծել վերաօգտագործելի և հնարավորինս պարամետրացված: Արդեն պատրաստի և ստուգված սխեմաների օգտագործումը հնարավորություն է տալիս էապես կրճատել ընդհանուր համակարգի նախագծման ժամանակը և նվազեցնել նոր սխալների առաջացման հավանականությունը: Վերաօգտագործելիությունը հանդիսանում է մտավոր սեփականության բլոկների ստեղծման հիմնական շարժառիթը:

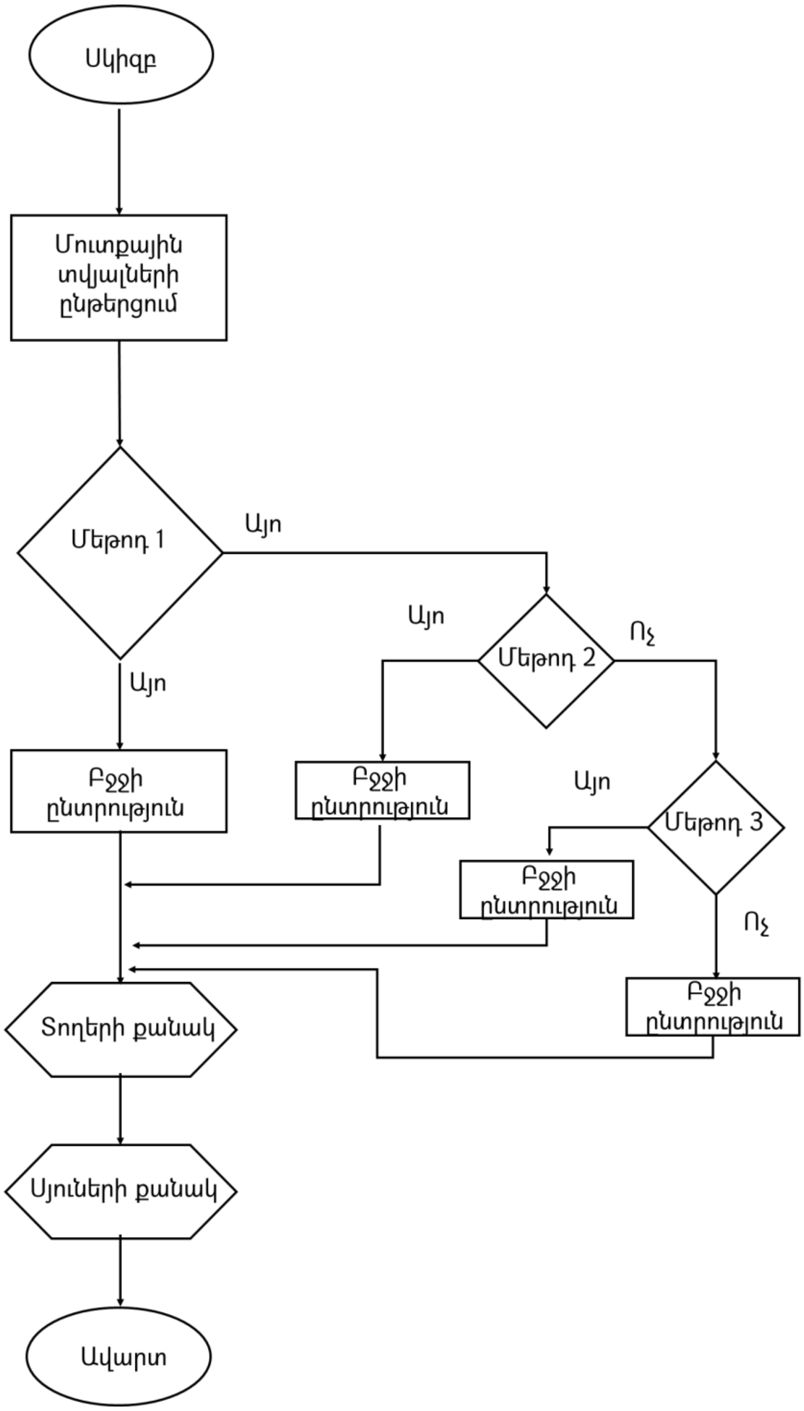
ԿՄՕԿ կիսահաղորդչային տեխնոլոգիաների մասշտաբավորման հետևանքով սահմանված նոր չափերով սխեմաների նախագծումն իրականացնելիս նախորդ տեխնոլոգիայի ԻՍ-երը ենթարկվել են որոշակի փոփոխությունների և համապատասղանեցվել են նոր տեխնոլոգիային: Նման գործընթաց իրագործելը դառնում է աշխատատար և աննպատակահարմար, երբ պլանար տրանզիստորներից անցում է կատարվում եռաչափ տրանզիստորների, որոնք ունեն էական կառուցվածքային տարբերություններ:

Եռաչափ տրանզիստորներով սխեմաների նախագծման գործընթացը հեշտ և ավելի արագ դարձնելու համար մշակվել է 3D-Compiler [109] ծրագրային գործիքը, որն օգտվելով եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների Գլուխ 2-ում մշակված մեթոդներից՝ մուտքային տվյալների հիման վրա նախագծում է հիշողության մատրիցների, դոմինո տրամաբանությամբ կառուցված փականների և տրիգերների տրանզիստորային մակարդակի SPICE նկարագրությունը՝ կիրառելով օգտագործողի կողմից ներմուծված տվյալ տեխնոլոգիայի նվազագույն ֆիների քանակը:

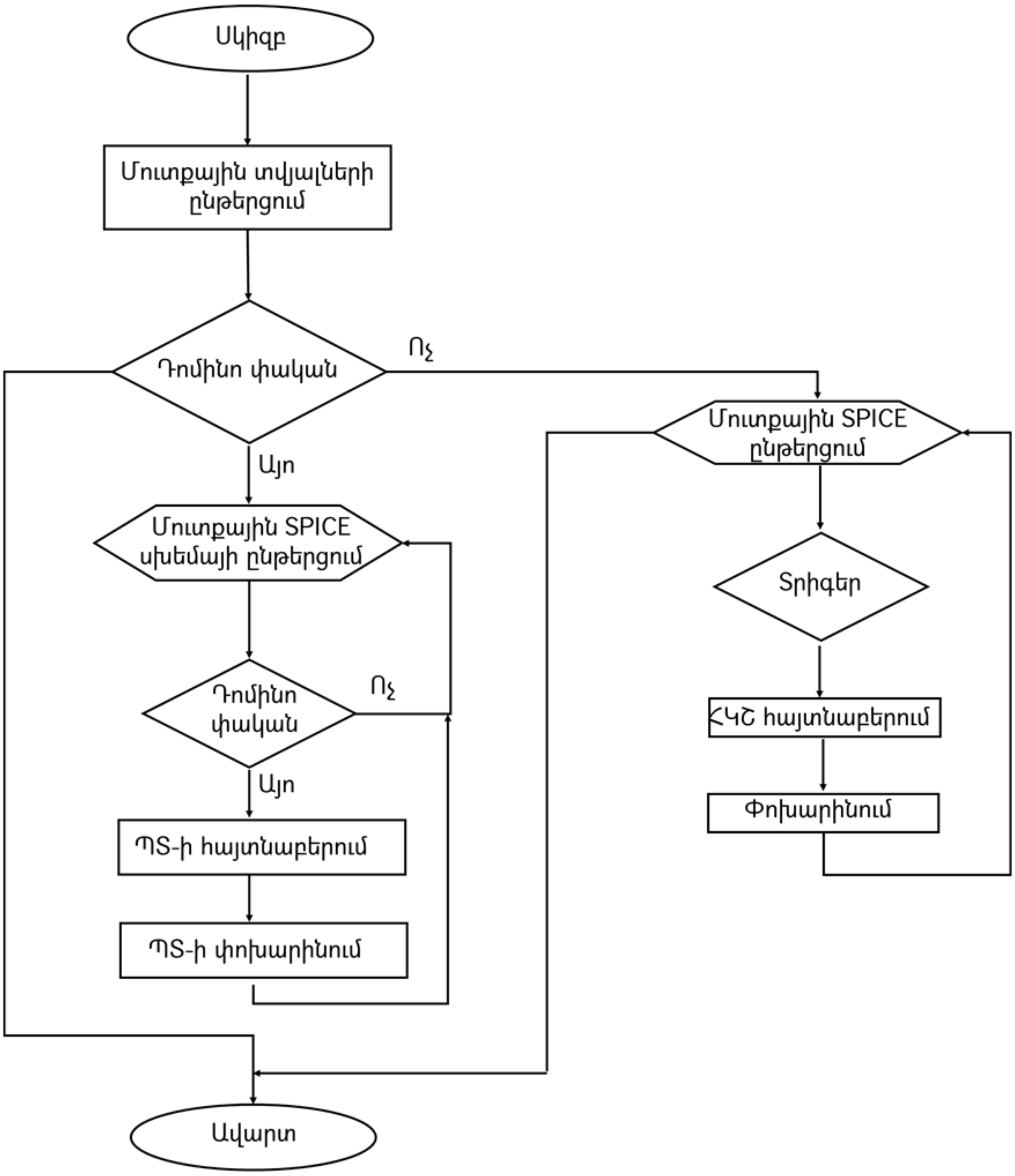
3D-Compiler ծրագրային միջոցը մշակվել է Python [123] ծրագրավորման լեզվով և դրա scypy [124], MySQLdb [125] և PyQT [126] գրադարանների միջոցով և նախատեսված է Linux [127] օպերացիոն համակարգով աշխատելու համար:

Մշակված ծրագրային գործիքով վերոհիշյալ սխեմաների նախագծումը հանգեցնում է նախագծման ժամանակի 10 անգամ կրճատման, իսկ կիրառված նախագծման մեթոդի շնորհիվ՝ սխեմայի էապես փոքր մակերեսի և բարձր արագագործության:

3D-Compiler ծրագիրը, կախված դրված առաջադրանքից, գործում է կամ որպես հիշողության մատրիցների կոմպիլյատոր (նկ. 3.1), կամ նշված դոմինո տրամաբանական փականներն ու տրիգերները փոխարինում է մշակված մեթոդներով իրականացված համապատասխան սխեմաներով (նկ. 3.2):



Նկ. 3.1. 3D-Compiler ծրագրի՝ որպես հիշողության կոմպիլյատորի աշխատանքը

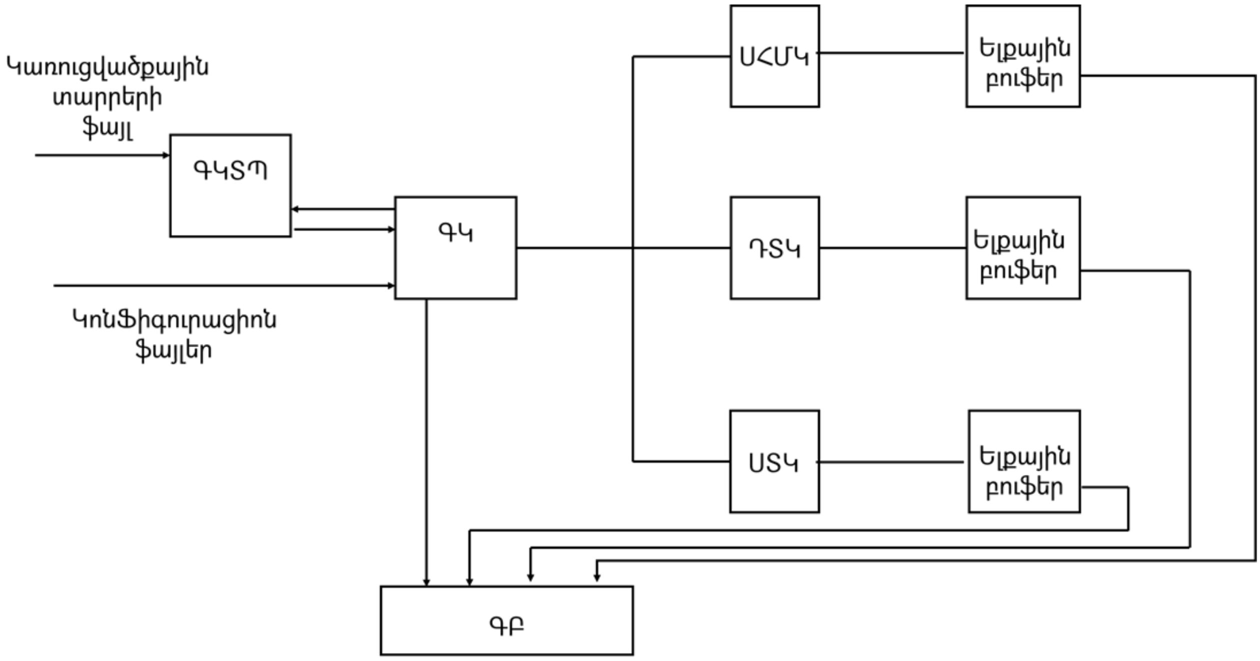


Նկ. 3.2. 3D-Compiler ծրագրի՝ դոմինո փականների և տրիգերների կոմպիլյացման ռեժիմում բլոկ-դիագրամը

3.1. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման ծրագրային միջոցի աշխատանքի սկզբունքը

Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման մշակված 3D-Compiler ծրագրային միջոցն ունի հետևյալ սկզբունքային կառուցվածքը (նկ. 3.3)՝

- գրադարանային կառուցվածքային տարրերի պահոց (ԳԿՏՊ),
- գործողությունների կազմակերպիչ (ԳԿ),
- ստատիկ հիշողության մատրիցի կոմպիլյատոր (ՍՀՄԿ) և ելքային բուֆեր,
- դոմինո տրամաբանության կոմպիլյատոր (ԴՏԿ) և ելքային բուֆեր,
- ստատիկ տրիգերների կոմպիլյատոր (ՍՏԿ) և ելքային բուֆեր,
- գրաֆիկական բլոկ (ԳԲ):

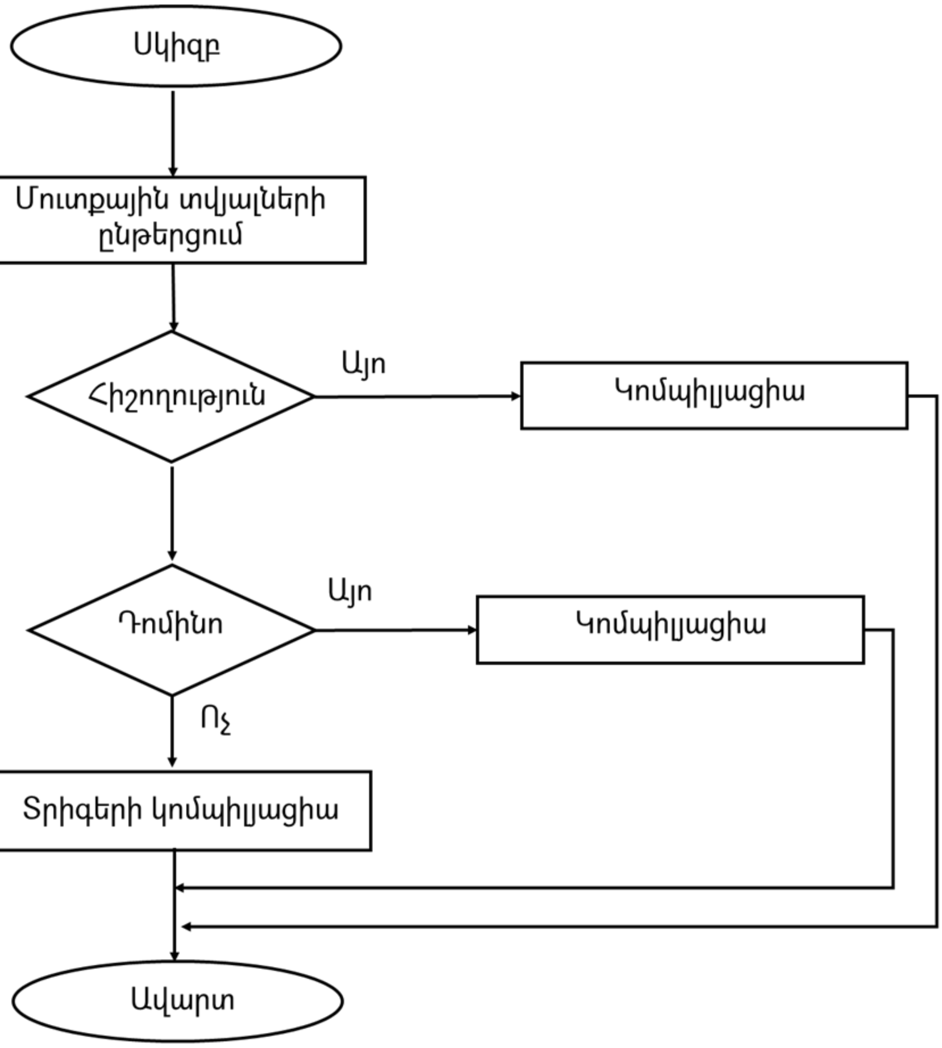


Նկ. 3.3. 3D-Compiler ծրագրային միջոցի սկզբունքային կառուցվածքը

Կոմպիլյացման բոլոր գործողություններն իրականացվում են զուգահեռաբար, ինչի շնորհիվ ծրագիրը 1-2 րոպեի ընթացքում գեներացնում է տարբեր բարդության կառուցվածքներ:

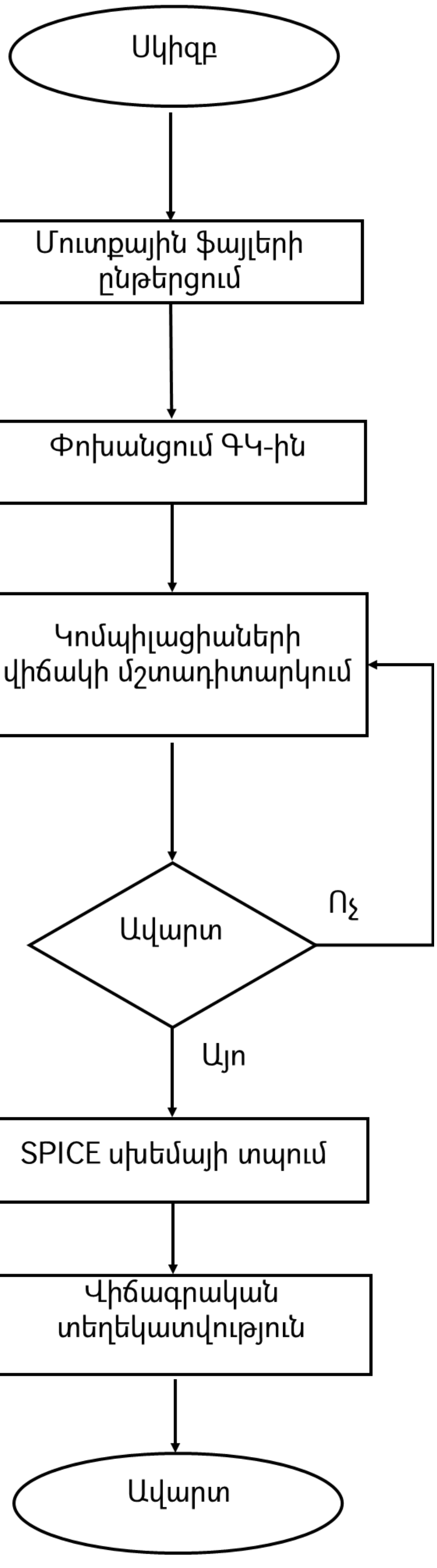
ԳԿՏՊ-ն ներկայացնում է SQL տվյալների հենքով ծրագրի ներքին հիշողությունը, որտեղ պահվում են ստատիկ հիշողության բջջի, դոմինո տրամաբանության լիցքի պաշտպանիչ սխեմայի և տրիգերի հետադարձ կապի շրջիչի՝ առաջարկված մեթոդներին համապատասխանող SPICE նկարագրությունները: Այդ կառուցվածքներին SQL տվյալների բազայում կցված են համապատասխան դրոշմային բառեր՝ mem_cell, domino_charge_protector և feedback_inverter, որոնք արագացնում են համապատասխան սխեմայի կոմպիլյացման համար ճիշտ կառուցվածքի հայտնաբերումը տվյալների բազայից: ԳԿՏՊ-ում պահվող կառուցվածքների ցանկն ու պարունակությունը ենթակա է փոփոխության հատուկ կոնֆիգուրացիոն ֆայլի միջոցով, որում գրված են ԳԿՏՊ-ում նոր տարր ավելացնելու կամ գոյություն ունեցողները փոփոխել/ջնջելու հրահանգներ:

ԳԿ-ն ներկայացնում է որոշում կայացնող ֆունկցիոնալ բլոկ (նկ. 3.4), որը ծրագրի մուտքին տրված հրահանգների հիման վրա գեներացնում է հատուկ հրահանգներ՝ ակտիվացնելով վերոհիշյալ 3 կոմպիլյացիոն բլոկերից միմասը կամ բոլորը: ԳԿ-ը, ստանալով սխեմաների կոմպիլյատորների ընթացքի մասին տեղեկությունը, հրահանգում է ԳԲ-ին արտածել համապատասխան հաղորդագրությունը կամ ԵԲ-ից ընդունել ելքային սխեման:



Նկ. 3.4. Գործողությունների կազմակերպչի բլոկ-դիագրամը

ԳԲ-ը պատասխանատու է կոմպիլյացված սխեմայի կառուցվածքային տեսքի գեներացման և արտապատկերման համար (նկ. 3.5):



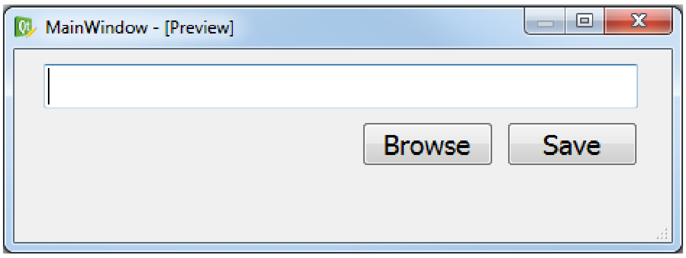
Նկ. 3.5. Գրաֆիկական բլոկի աշխատանքի դիագրամը

Այսպիսով, մշակված ծրագրային գործիքը օժտված է տարբեր սխեմաների կոմպիլյացմանհատկությամբ, թե ունակությամբ օգտվելով Գլուխ 2-ում առաջարկված տարրական կառուցվածքներից:

3.2. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման ծրագրային միջոցի կառուցվածքը

3D-Compiler ծրագրային գործիքի հարմարավետ ինտերֆեյսը թույլ է տալիս հեշտորեն հասկանալ ծրագիրն աշխատեցնելու տարբերակները և հաշված րոպեների ընթացքում ստանալ անհրաժեշտ սխեմայի՝ եռաչափ տրանզիստորներով կառուցված նախագիծը:

Գործիքն առաջին անգամ միացնելիս բացվում է նախնական համալարումների պատուհանը (նկ. 3.6), որի միջոցով տրվում է կառուցվածքային տարրերը պարունակող ֆայլը: OK կոճակը սեղմելուց հետո բացվում է ծրագրի հիմնական պատուհանը (նկ. 3.7):



Նկ. 3.6. Կառուցվածքային տարրերի ֆայլի ներմուծման պատուհանը

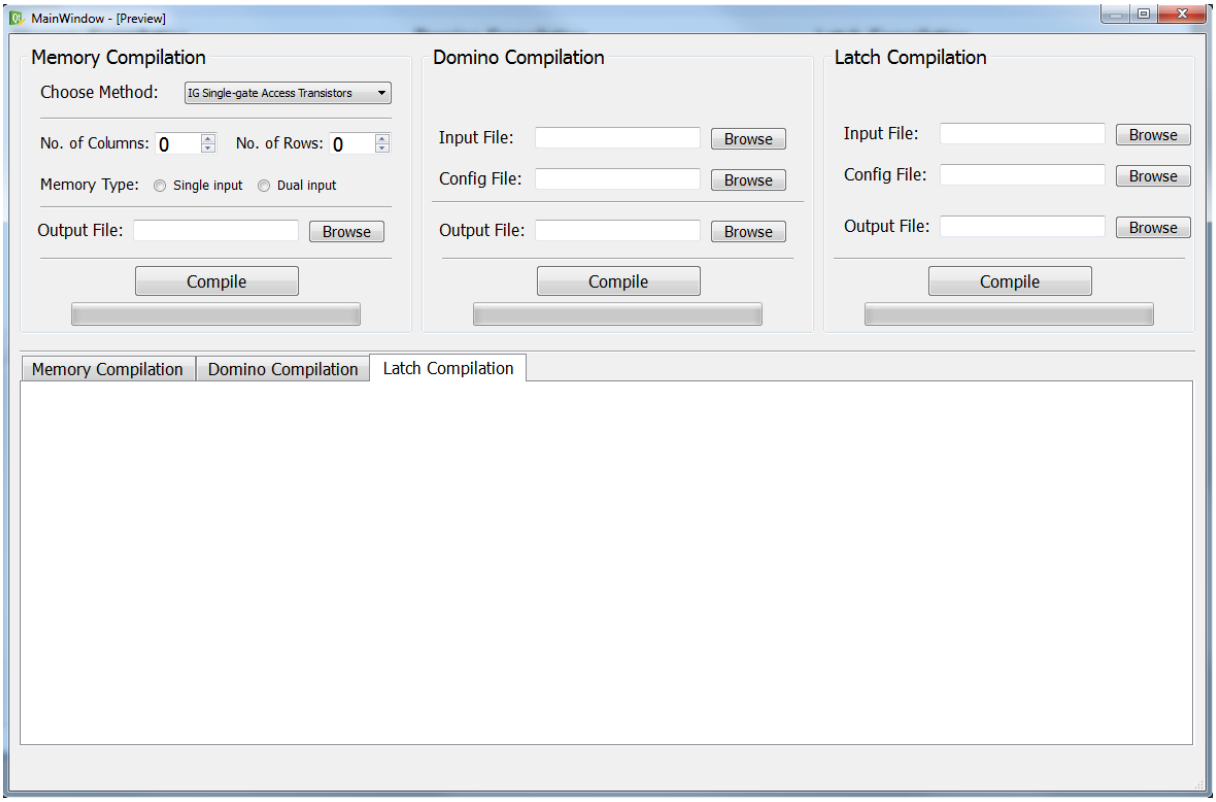
Նկ. 3.7-ում պատկերված գրաֆիկական ինտերֆեյսը բաղկացած է հիշողության մատրիցների, դոմինո փականների և տրիգերների կոմպիլյացման կոնֆիգուրացիոն բլոկերից և արդյունքների պատուհաններից:

Հիշողության մատրիցների կոմպիլյացման ժամանակ անհրաժեշտ է ընտրել ստատիկ հիշողության բջջի տեսակը (նկ. 3.8), մատրիցի տողերի և սյուների քանակը, հիշողության մուտքերի քանակը և ելքային SPICE [128] ֆայլի անունը:

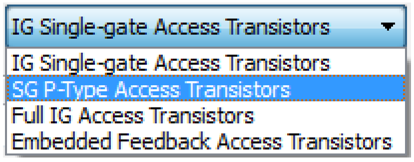
Բջջի տեսակը կարող է լինել Գլուխ 2-ում քննարկված 4 տարբերակներից մեկը՝

- ԱՓԵՏ միափական հասանելիության տրանզիստորով մեթոդ,
- P տիպի ԿՓԵՏ միացմամբ հասանելիության տրանզիստորներով մեթոդ,
- ԱՓԵՏ երկփական հասանելիության տրանզիստորներով մեթոդ,
- Ներկառուցված հետադարձ կապով մեթոդ:

Անհրաժեշտ կոնֆիգուրացիաները կատարելուց հետո պետք է սեղմել Compile կոճակը: Ընթացքի ցուցիչի միջոցով կարելի է տեսնել, թե երբ է ավարտվելու կոմպիլյացման գործողությունը (նկ. 3.9):

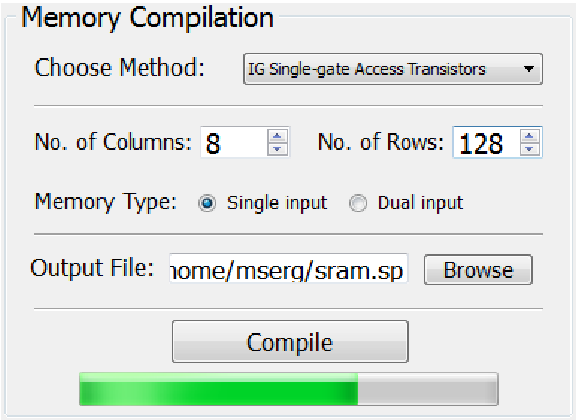


Նկ. 3.7. 3D-Compiler ծրագրի հիմնական պատուհանը

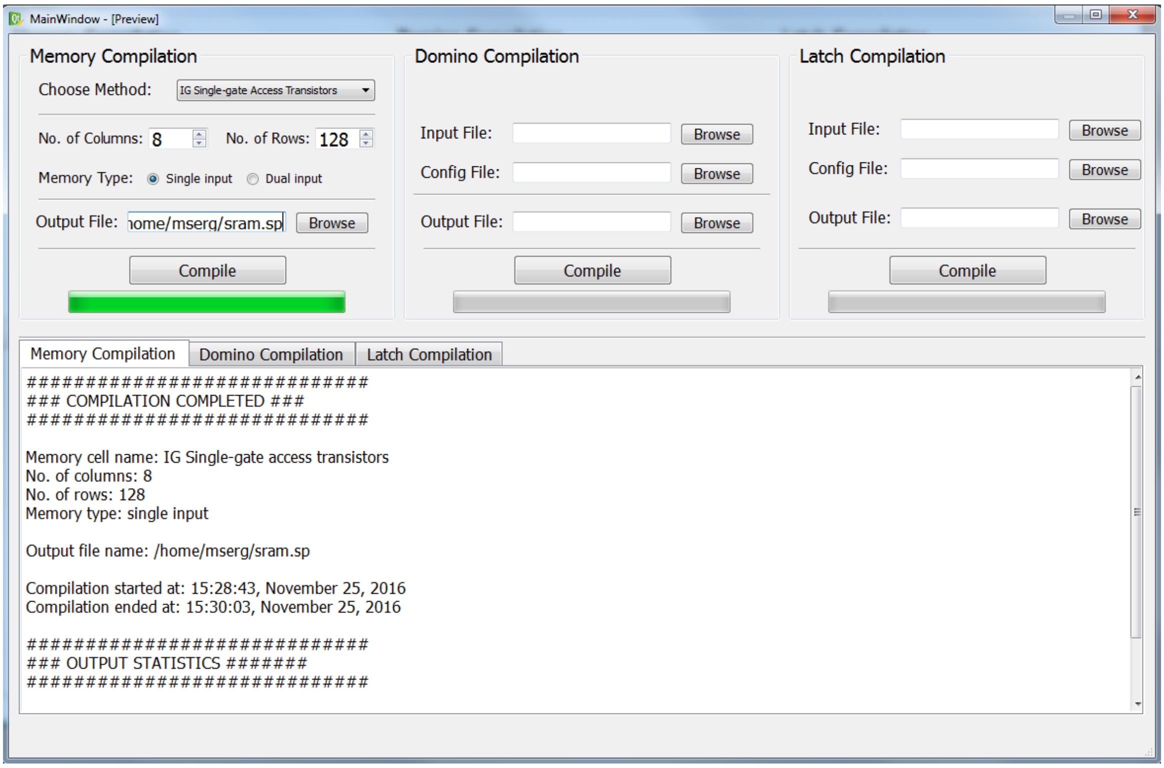


Նկ. 3.8. ՀԲ-երի ընտրության մենյուն

Կոմպիլյացիայի ավարտից հետո Memory Compilation դաշտում հայտնվում է իրականացված գործողության մասին մանրամասն զեկույցը (նկ. 3.10):



Նկ. 3.9. Հիշողության մատրիցի կոմպիլյացիայի կոնֆիգուրացիոն պատուհանը



Նկ. 3.10. Հիշողության մատրիցի կոմպիլյացիայի զեկույցը

Ելքային SPICE նկարագրության ֆայլը բաղկացած է լինում հիշողության բջջի ենթահանգույցների առանձին կիրառումներից.

.subckt 6t_cell a0 a1 wl

xM1 n1 n2 VDD VDD p105 nfin=1 nf=1 l=0.14u m=1

xM2 n2 n1 VDD VDD p105 nfin=1 nf=1 l=0.14u m=1

xM3 a1 wl n1 VSS n105 nfin=1 nf=1 l=0.14u m=1

xM4 n2 wl a0 VSS n105 nfin=1 nf=1 l=0.14u m=1 xM5 n1 n2 VSS VSS n105 nfin=1 nf=1 l=0.14u m=1

xM6 n2 n1 VSS VSS n105 nfin=1 nf=1 l=0.14u m=1

.ends

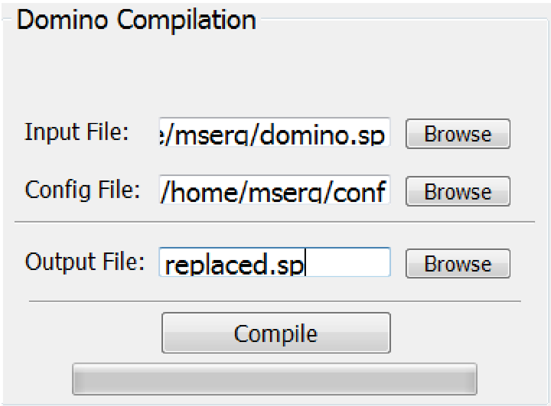
X1 n1 n2 wl 6t_cell

X2 n3 n4 wl 6t_cell

......

.end

Դոմինո տրամաբանության կոմպիլյացիան ներկայացնում է գոյություն ունեցող եռաչափ դոմինո սխեմատեխնիկական բջիջներում պաշտպանիչ տրանզիստորների՝ առաջարկվող մեթոդով փոխարինումը: Կոմպիլյացիան սկսելու համար անհրաժեշտ է լրացնել համապատասխան դաշտերը և սեղմել Compile կոճակը (նկ. 3.11):



Նկ. 3.11. Դոմինո փականների կոմպիլյացիայի կոնֆիգուրացիոն պատուհանը

Կոմպիլյացիայի ավարտից հետո Domino Compilation դաշտում տպվում է հետևյալ հաղորդագրությունը.

##############################

COMPILATION COMPLETED

##############################

Input file name: /home/mserg/domino.sp

Config file name: /home/mserg/conf

Output file name: /home/mserg/replaced.sp

Compilation started at: 15:28:43, December 4, 2016

Compilation ended at: 15:30:03, December 4, 2016

##############################

OUTPUT STATISTICS

##############################

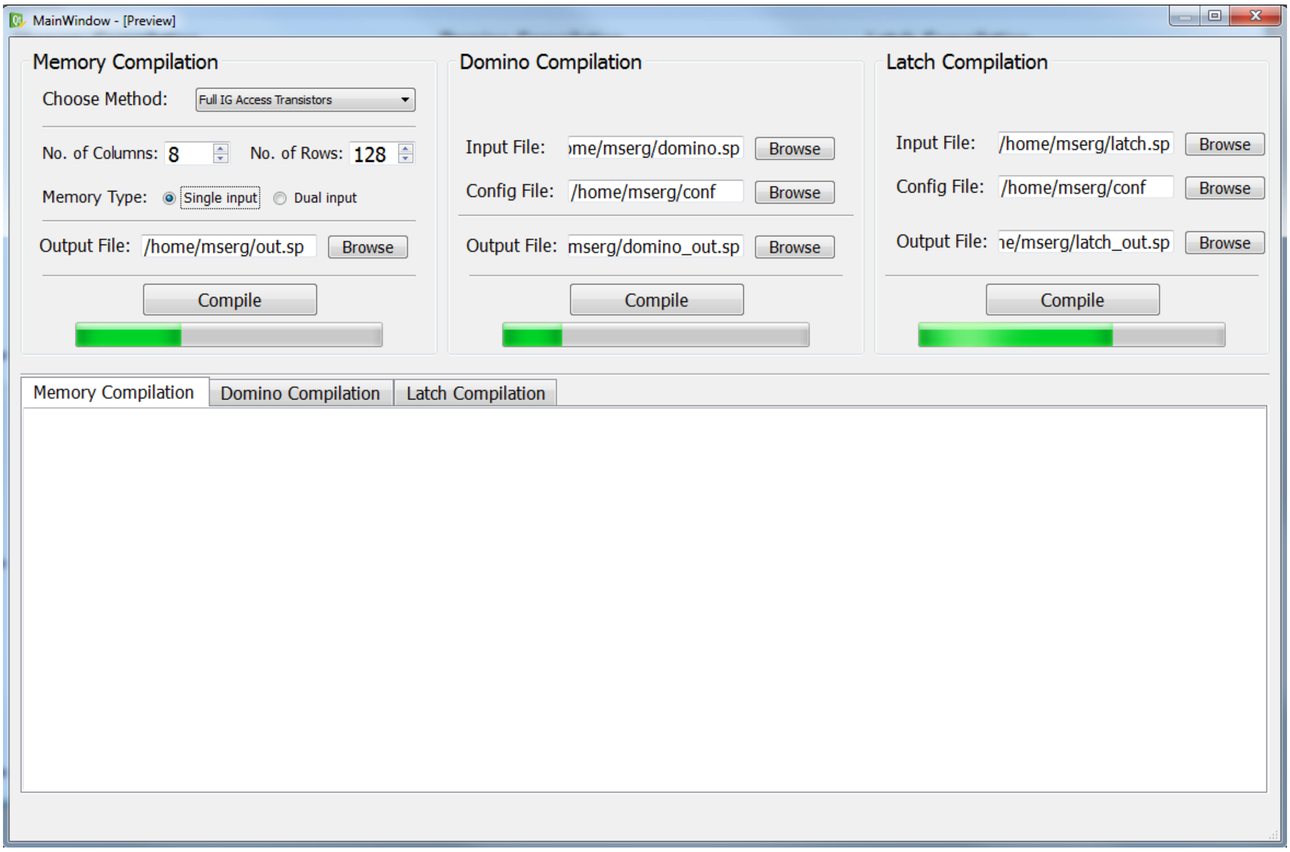
No. of modified cells: 146

No. of cells in netlist: 1564

##############################

Այս դեպքի նմանությամբ է իրականացվում նաև տրիգերների կոմպիլյացիան:

3D-Compiler ծրագրային միջոցը հնարավորություն է տալիս բոլոր 3 կոմպիլյացիաներն իրականացնել զուգահեռաբար: Դա իրականացվում է գործընթացային զուգահեռականության հնարավորության շնորհիվ: Միաժամանակ, բոլոր կոմպիլյացիաները սկսելու համար անհրաժեշտ է լրացնել բոլոր դաշտերը և սեղմել յուրաքանչյուրի Compile կոճակը (նկ. 3.12):



Նկ. 3.12. Զուգահեռ կոմպիլյացիաներ իրականացնող ռեժիմը

Մշակված ծրագրային գործիքը մուտքին ստանում է 3 կոնֆիգուրացիոն ֆայլեր՝

- կառուցվածքային տարրերի նկարագրություններ,
- դոմինո բջիջների անուններ պարունակող կոնֆիգուրացիոն ֆայլ,
- փոփոխության ենթակա ստատիկ տրիգերները նույնականացնող ֆայլ:

Կառուցվածքային տարրերի նկարագրություններով կոնֆիգուրացիոն ֆայլն ունի հետևյալ տեսքը՝

struct_name: ig_single_gate_access

type: memory

bitcell_count: 1

struct_name պարամետրը պարունակում է տարրական կառուցվածքի անունը, type-ը մատնանշում է սխեմայի տեսակը: Այն կարող է ընդունել memory, domino և latch արժեքներից որևէ մեկը, որոնք ցույց են տալիս համապատասխանաբար տարրական սխեմայի հիշողության, դոմինո տրամաբանության և ստատիկ տրիգերների համար նախատեսված լինելը: bitcell_count պարամետրը ցույց է տալիս կառուցվածքում առկա 6S հիշողության բջիջների քանակը: Մեծ հիշողությամբ մատրիցների կառուցումն արագացնելու համար տարրական սխեմաներում պարունակվող ՀԲ-երի թիվը կարող է մեծ լինել: bitcell_count-ը ընդունում է ցանկացած դրական ամբողջ արժեք:

Այս կոնֆիգուրացիոն ֆայլն ընթերցվում է ծրագրի առաջին անգամ միանալու ժամանակ, ինչից հետո ֆայլի պարունակությունը հիշվում է SQL տվյալների բազայի հենքով ծրագրի ներքին հիշողության մեջ:

Դոմինո բջիջների անուններ պարունակող կոնֆիգուրացիոն ֆայլն ունի հետևյալ տեսքը՝

name: AND2X1_DOMINO,

name: OR2X1_DOMINO:

name պարամետրը այս ֆայլի միակ պարամետրն է, որը ստանում է փոփոխության ենթակա դոմինո փականի անունը: Փականների անունները կարելի է նշել առանձին-առանձին կամ մեկ ընդհանուր նշումով՝

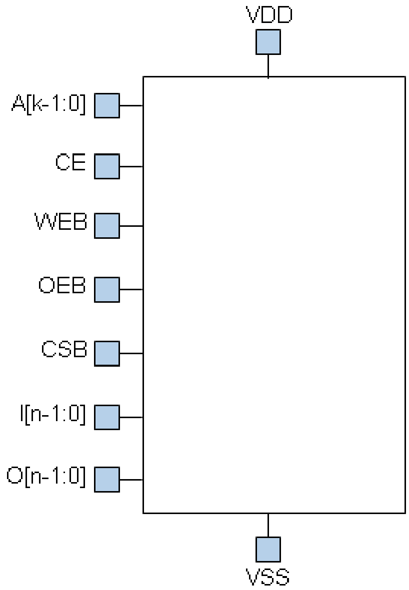
name: *_DOMINO:

Համանման կառուցվածք ունի նաև փոփոխության ենթակա ստատիկ տրիգերները նույնականացնող ֆայլը:

3.3. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման ծրագրային միջոցի արդյունավետության գնահատումը

Մշակված 3D-Compiler ծրագրային միջոցի աշխատանքի արդյունավետությունը գնահատելու նպատակով իրականացվել են եռաչափ տրանզիստորներով հետևյալ սխեմաների նախագծումները՝ 32X128, 64X32, 256X32 ստատիկ հիշողության մատրիցներ, դոմինո ԿԱՄ, ԵՎ, ԵՎ-ԿԱՄ-ՇՐՋՈՒՄ և ԿԱՄ-ԵՎ-ՇՐՋՈՒՄ փականներ և ստատիկ տրիգերներ: Բոլոր սխեմաները նախագծվել են ստանդարտ ԿՓԵՏ միացմամբ տրանզիստորներով և Գլուխ 2-ում նկարագրված համապատասխան մեթոդներով: Իրականացվել են դրանց հապաղումների և էներգասպառման արժեքների գնահատում և համեմատություն:

Նախագծված ստատիկ հիշողությունները տվյալների մեկ մուտքով սխեմաներ են (նկ. 3.13):



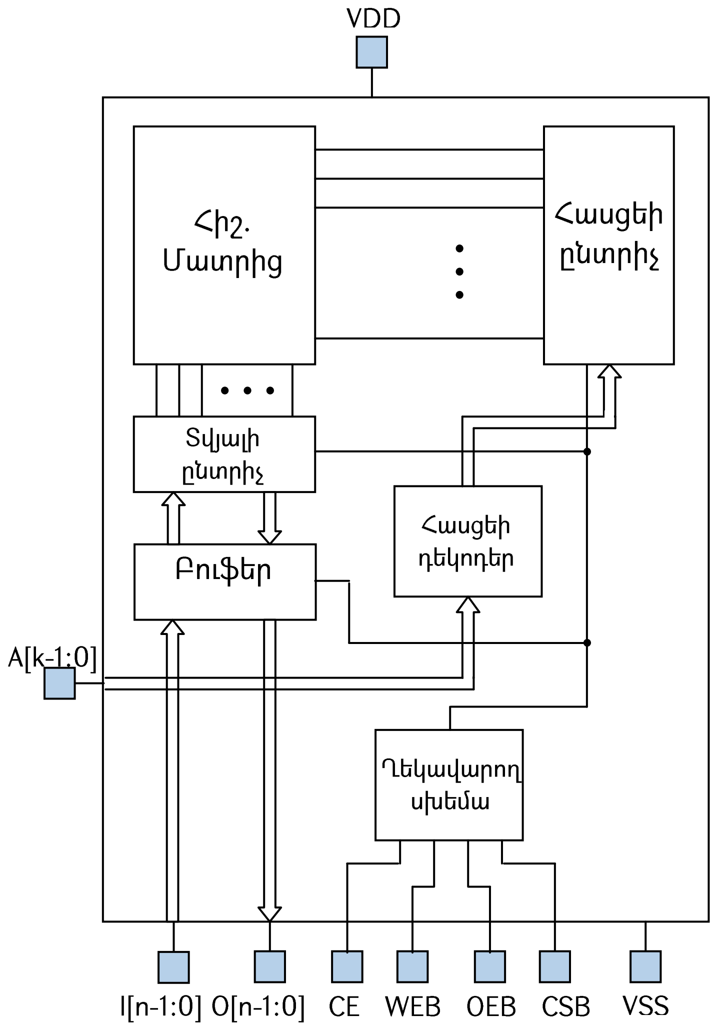
Նկ. 3.13. Ստատիկ հիշող սարքի ելուստները

Նկ. 3.13-ում պատկերված ելուստները հետևյալն են՝

- ընթերցման/գրանցման k-կարգանի A հասցե,
- դրական ակտիվ ճակատով CE սինքրոազդանշան,
- «0» ակտիվ մակարդակով WEB գրանցումը թույլատրող մուտք ,
- «0» ակտիվ մակարդակով OEB ելքային ազդանշանը ակտիվացնող մուտք,
- «0» ակտիվ մակարդակով CSB սխեման ակտիվացնող մուտք,

- մուտքային տվյալների n-կարգանի I դող,
- ելքային տվյալների n-կարգանի O դող,
- տրամաբանական «1» VDD լարում,
- տրամաբանական «0» VSS լարում:

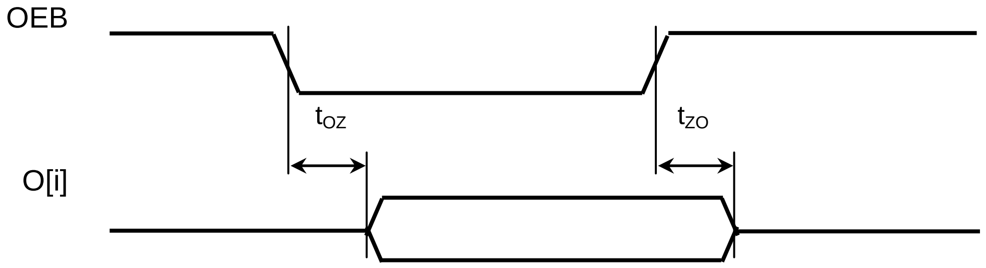
Միամուտք ստատիկ հիշողության սկզբունքային սխեման (նկ. 3.14) ունի հետևյալ կառուցվածքը՝ հիշողության մատրից, հասցեի ընտրիչ, տվյալների ընտրիչ, մուտք/ելք բուֆեր, հասցեի դեկոդավորիչ և ղեկավարող սխեմա:



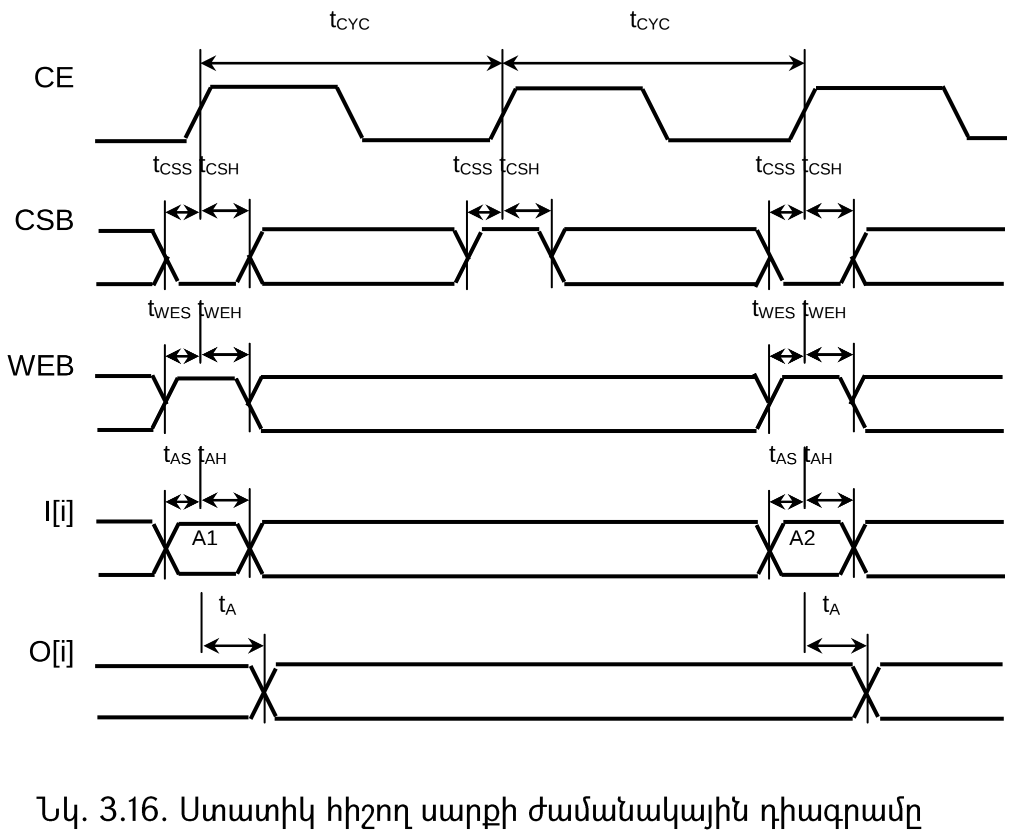
Նկ. 3.14. Ստատիկ հիշող սարքի կառուցվածքը

Ստատիկ հիշողության սխեման մտնում է աշխատանքի գրանցման ռեժիմ, երբ CSB=0 և WEB=0: Գրանցման ռեժիմի ընթացքում I[n-1:0] մուտքի տվյալները գրանցվում են D(A[k 1:0]) հիշողության հասցեում: Երբ OEB=1, ելքային O[n-1:0] դողը գտնվում է բարձր դիմադրության վիճակում: Ընթերցման/գրանցման գործողության ավարտից հետո, երբ OEB=0, տվյալները հայտնվում են ելքային դողի վրա: Ստատիկ սխեմաներում նվազեցված է կորստի էներգասպառումը: Երբ մատրիցի հետ որևէ գործողություն չի կատարվում, CCB=1 ազդանշանի միջոցով այն դրվում է սպասման ռեժիմի մեջ: Այդ դեպքում տվյալային և հասցեի մուտքերը ապաակտիվացվում են, իսկ հիշողության հասցեներում տեղակայված տվյալները պահպանվում են:

Ստատիկ հիշողությունների ժամանակային դիագրամները (նկ. 3.15, 3.16) նկարագրում են դրանց աշխատանքը՝ կախված մուտքային ազդանշանների արժեքներից:



Նկ. 3.15. Ելքային ելուստի ակտիվացում-ապաակտիվացումը



Նկ. 3.16. Ստատիկ հիշող սարքի ժամանակային դիագրամը

3D-Compiler ծրագրային գործիքով նախագծված մյուս սխեմաների պարզությունը հաշվի առնելով՝ դրանք չեն նկարագրվում:

Բոլոր սխեմաները նախագծվել են 0.8 *Վ* տիպային սնուցման լարման, 25° ջերմաստիճանի և 1 *ԳՀց* աշխատանքային հաճախականության պայմաններում:

Այսպիսով, 3D-Compiler ծրագրային գործիքով եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման միջոցների ներկառուցումը տարբեր սխեմաներում ցուցաբերում է կորստի էներգասպառման եւ հապաղումների բարելավում՝ զբաղեցրած մակերեսի եւ նախագծման բարդության չնչին ավելացման հաշվին(աղ. 3.1-3.6):

Աղյուսակ 3.1

Տարբեր մեթոդներով նախագծված ստատիկ հիշողության սխեմաների կորստի էներգասպառումը

Սխեմա	ԿՓԵՏ (նՎտ)	Մեթոդ 1 (նՎտ)	Մեթոդ 2 (նՎտ)	Մեթոդ 3 (նՎտ)	Մեթոդ 4 (նՎտ)
32X128 ՍՀ	25	15	10	16	12
64X32 ՍՀ	21	10	7	12	9
256X32 ՍՀ	27	16	12	17	15

Աղյուսակ 3.2

Տարբեր մեթոդներով նախագծված ստատիկ հիշողության սխեմաների կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսը

Սխեմա	ԿՓԵՏ (մկմ քառ.)	Մեթոդ 1 (մկմ քառ.)	Մեթոդ 2 (մկմ քառ.)	Մեթոդ 3 (մկմ քառ.)	Մեթոդ 4 (մկմ քառ.)
32X128 ՍՀ	57	58,9	59,45	59,2	61
64X32 ՍՀ	46	47,28	47,09	47,38	49
256X32 ՍՀ	98	101,92	103,194	102,41	105,5

Աղյուսակ 3.3

Առաջարկված մեթոդով նախագծված դոմինո տրամաբանական փականների կորստի էներգասպառումը

Սխեմա	ԿՓԵՏ (նՎտ)	Դոմինո մեթոդ	Դոմինո-ԿՓԵՏ
16X1 ԿԱՄ	25,43	16,8	34 %
16X1 ԵՎ	23,9	16,24	32 %
ԿԱՄ-ԵՎ-ՇՐՋՈՒՄ	18	12,33	31,5 %
ԵՎ-ԿԱՄ-ՇՐՋՈՒՄ	15,4	8,56	44,4 %

Աղյուսակ 3.4

Առաջարկված մեթոդով նախագծված դոմինո տրամաբանական փականների կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսը

Սխեմա	ԿՓԵՏ (մկմ քառ.)	Դոմինո մեթոդ	Դոմինո-ԿՓԵՏ
16X1 ԿԱՄ	10	11,5	13 %
16X1 ԵՎ	11,5	12,8	10,16 %
ԿԱՄ-ԵՎ-ՇՐՋՈՒՄ	16	18	11 %
ԵՎ-ԿԱՄ-ՇՐՋՈՒՄ	14,9	16,96	13,8 %

Աղյուսակ 3.5

Առաջարկված մեթոդով նախագծված ստատիկ տրիգերներից (US) բաղկացած սխեմաների հապաղումը

Սխեմա	ԿՓԵՏ (պվ)	Մեթոդ (պվ)	Մեթոդ-ԿՓԵՏ
8-կարգանի US	120	82	31,67 %
16-կարգանի US	243,23	159	34,63 %
32-կարգանի US	478,9	318,45	33,5 %

Աղյուսակ 3.6

Առաջարկված մեթոդով նախագծված սպարիկ փրիգերներից (US) բաղկացած սխեմաների կորսփի էներգասպառումը

Սխեմա	ԿՓԵՏ (նՎտ)	Մեթոդ (նՎտ)	Մեթոդ-ԿՓԵՏ
8-կարգանի US	67	68,4	2,04 %
16-կարգանի US	74	76,62	3,54 %
32-կարգանի US	82	86,86	5,93 %

Այսպիսով, անկախ փականներով եռաչափ տրանզիստորներով սխեմաների նախագծման՝ ատենախոսությունում առաջարկվող մեթոդների իրագործումը 3D-Compiler ծրագրային գործիքով ակնհայտորեն ավելի արդյունավետ է, քան կարճ միացված փականներով տրանզիստորներով նախագծման գոյություն ունեցող մեթոդներինը:

Եզրակացություններ

1. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծումը՝ առաջարկվող մեթոդներով ու միջոցներով իրագործվել է 3D-Compiler ծրագրային գործիքով, որի շնորհիվ հնարավոր է դառնում ատենախոսությունում առաջարկված սխեմաների մշակումն իրականացնել հաշված վայրկյաններից մինչև մի քանի րոպեի ընթացքում:
2. Առաջարկված մեթոդների կիրառմամբ 3D-Compiler ծրագրային գործիքով իրագործված հիշողության սխեմաների էներգասպառումը, կարճ փականային միացմամբ եռաչափ տրանզիստորներով նախագծված սխեմաների հետ համեմատած բարելավվել է մոտ 36-66,7%, կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի 2,38-7,66% վատթարացման հաշվին: Դոմինո փականների նախագծման միջոցի կիրառումը ապահովել է 31,5-44,4% էներգասպառման նվազեցում, ԿԲ-ի վրա զբաղեցրած մակերեսի 11-13,8% մեծացման հաշվին:

3. Ստատիկ տրիգերների հետադարձ կապի շրջիչի նախագծման մեթոդը, 3D-Compiler ծրագրային գործիքով իրագործվելով, կորստի էներգասպառման 2,04-5,93% մեծացման հաշվին ապահովում է տրիգերների շղթայի հապաղման 31,67-33,5% բարելավում՝ համեմատած ԿՓԵՏ միացմամբ տրանզիստորների հետ:

ԵԶՐԱՀԱՆԳՈՒՄ

1. Ձևակերպվել են եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների մշակման միջոցներին առաջադրվող պահանջները, որոնց բավարարումն ապահովում է արագագործության, էներգասպառման և կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի անհրաժեշտ փոխզիջումային արժեքները:
2. Առաջարկվել է եռաչափ տրանզիստորներով ստատիկ հիշողության բջիջների նախագծման 4 մեթոդ, որոնք, օգտագործելով դրանց անկախ փականներ ունենալու հատկությունը, հայտնի լուծումների համեմատ ապահովում են բջջի ստատիկ աղմկակայունության պաշարի անհրաժեշտ մեծ արժեք և փոքր ստատիկ էներգասպառում՝ կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի ոչ էական մեծացման հաշվին:
3. Մշակվել է եռաչափ տրանզիստորներով դոմինո տրամաբանության դինամիկ հանգույցում լիցքի պաշտպանման միջոց, որը, փականի զբաղեցրած մակերեսի աննշան ավելացման հաշվին, առկա մոտեցումների համեմատ ապահովում է էապես ցածր էներգասպառում և անհրաժեշտ բարձր արագագործություն:
4. Ստեղծվել է ստատիկ տրիգերների՝ էներգասպառման չնչին ավելացման հաշվին ստատիկ աղմկակայունության պաշարը և արագագործությունը բարելավող մեթոդ, որի հիման վրա տրիգերի հետադարձ կապի շրջիչն իրականացվել է անկախ փականներով եռաչափ տրանզիստորներով:
5. Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծումը՝ առաջարկվող մեթոդներով ու միջոցներով, իրագործվել է 3D-Compiler ծրագրային գործիքով, որի շնորհիվ հնարավոր է դառնում սխեմաների մշակումն իրականացնել հաշված վայրկյաններից մինչև մի քանի րոպեի ընթացքում: Առաջարկված մեթոդների կիրառմամբ իրագործված հիշողության սխեմաների էներգասպառումը, կարճ փականային միացմամբ եռաչափ տրանզիստորներով նախագծված սխեմաների համեմատ, բարելավվել է մոտ 36 – 66,7% -ով, կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի 2,38-7,66% վատթարացմամբ: Դոմինո փականների նախագծման առաջարկվող միջոցը ԿՓԵՏ կառուցվածքի համեմատ ապահովում է էներգասպառման 31,5-44,4% նվազեցում, ԿԲ-ի վրա զբաղեցրած մակերեսի 11-13,8% մեծացման հաշվին:

Ստատիկ տրիգերների հետադարձ կապի շրջիչի նախագծման մեթոդը, 3D-Compiler ծրագրային գործիքով իրագործվելով, ԿՓԵՏ միացմամբ տրանզիստորների համեմատ ապահովում է տրիգերների շղթայի հապաղման 31,67-33,5% բարելավում՝ կորստի էներգասպառման 2,04-5,93% ավելացման հաշվին:

ՕԳՏԱԳՈՐԾՎԱԾ ԳՐԱԿԱՆՈՒԹՅՈՒՆ

1. Salman E, Friedman E. High Performance Integrated Circuit Design. - McGraw Hill Professional, 2012. -716p.
2. Kelin J. CMOS transistor scaling past 32nm and implications on variation // IEEE/SEMI Advanced Semiconductor Manufacturing Conference. - San Francisco, - 2010. - P.241-246.
3. Han L., Adam T., Peide D. Channel Length Scaling of MOS, MOSFETs // Computer Engineering and Birck Nanotechnology Center / Purdue University. -USA, -2012. -P. 8563–8569.
4. Pavlidis VF, Friedman EG. Three-dimensional integrated circuit design. Morgan Kaufmann, 2010. -308p.
5. Xie Y, Cong J, Sapatnekar SS. Three-dimensional integrated circuit design. 2010. - 284p.
6. Tajalli A. Design Trade-offs in Ultra-Low-Power Digital Nanoscale CMOS // IEEE Transactions on Circuits and Systems I: Regular Papers. 2011.- Vol. 58. - P.2189-2200.
7. Kawamura K, Yanagisawa M, Togawa N. A high-performance circuit design algorithm using data dependent approximation. // InSoC Design Conference (ISOCC). -2016. - P.95-96.
8. Oh H, Han B, McCluskey P, Han C, Youn BD. Physics-of-failure, condition monitoring, and prognostics of insulated gate bipolar transistor modules: A review. // IEEE Transactions on Power Electronics. 2015. -Vol. 30, No. 5. -P.2413-2426.
9. Tsividis Y, McAndrew C. Operation and Modeling of the MOS Transistor. -Oxford Univ. Press; 2011. -723p.
10. Chemning H. FinFET 3D Transistor & the Concepts Behind it . -2011. - 28p. http://www.eecs.berkeley.edu/~hu/
11. Jang SL, Lai WC, Tzeng SS, Hsue CW. A wide-band divide-by-3 injection-locked frequency divider using tunable MOS resistor. // InSolid-State Circuits Conference (A-SSCC), 2015. -P.1-5.

12. Maheshwaram S, Manhas SK, Kaushal G, Anand B, Singh N. Vertical nanowire CMOS parasitic modeling and its performance analysis. // IEEE Transactions on Electron Devices. 2013. -Vol 60, No. 9. -P.2943-2950.
13. Riverola M, Sobreviela G, Uranga A, Barniol N. Intrinsic feedthrough current cancellation in a seesaw CMOS-MEMS resonator for integrated oscillators. // InFrequency Control Symposium (IFCS), 2016. -P.1-4.
14. Lau JH. Overview and outlook of three-dimensional integrated circuit packaging, three-dimensional Si integration, and three-dimensional integrated circuit integration. // Journal of Electronic Packaging. 2014. -Vol. 136, No. 4. 040801. -15p.
15. Otten R. Complexity and diversity in ic layout design. // InProceedings of the 2016 on International Symposium on Physical Design. -2016. -P.115-115.
16. Taur Y, Ning TH. Fundamentals of modern VLSI devices. -Cambridge university press; 2013. -461p.
17. Kannan N, Kumar MJ. Dielectric-modulated impact-ionization MOS transistor as a label-free biosensor. // IEEE Electron Device Letters. -2013. -Vol. 34, No. 12. -P.1575-1577.
18. Khateb F. The experimental results of the bulk-driven quasi-floating-gate MOS transistor. // AEU-International Journal of Electronics and Communications. 2015. -Vol. 69, No. 1. -P.462-466.
19. Hu C. Modern semiconductor devices for integrated circuits. -Prentice Hall; 2010. -384p.
20. Annaratone S. Digital CMOS circuit design. Springer Science & Business Media; 2012. -384p.
21. Masalskis G. Reverse engineering of CMOS integrated circuits. -Elektronika ir elektrotechnika. 2015. -Vol. 88, No. 8. -P.25-28.
22. Baker RJ. CMOS: circuit design, layout, and simulation. -John Wiley & Sons; 2008. -1038p.
23. Yuan HT, McLevige WV, Shih HD. GaAs bipolar digital integrated circuits. // VLSI electronics microstructure science. 2014. -P.173-213.

24. Kang W, Zhao W, Wang Z, Klein JO, Zhang Y, Chabi D, Zhang Y, Ravelosona D, Chappert C. An overview of spin-based integrated circuits. // InDesign Automation Conference (ASP-DAC) IEEE, 2014. -P.676-683.
25. Nadda K, Kumar MJ. Schottky collector bipolar transistor without impurity doped emitter and base: Design and performance. // IEEE transactions on electron devices. 2013. -Vol. 60, No. 9. -P.2956-2959.
26. Lefranc P, Frey D. Design methodology for optimising a high insulation voltage insulated gate bipolar transistor gate driver signal transmission function. // IET Power Electronics. 2015. -Vol. 8, No. 6. -P.1035-1042.
27. Meade R, Tehar-Zahav O, Sternberg Z, Megged E, Sandhu G, Orcutt JS, Ram R, Stojanović V, Michael R, Timurdogan E, Shainline J. Integration of silicon photonics in a bulk CMOS memory flow. // InOptical Interconnects Conference, 2013. IEEE. -P. 114-115.
28. Schwierz F, Pezoldt J, Granzner R. Two-dimensional materials and their prospects in transistor electronics. // Nanoscale. 2015. -Vol. 7, No.18. -P.8261-8283.
29. Bhunia S., Mukhopadhyay S. Low-Power Variation-Tolerant Design in Nanometer Silicon. - New York, 2011. –440p.
30. Ayala J., Alonso A. VLSI-SoC: Forward-Looking Trends in IC and Systems Design // 18th IFIP WG 10.5/IEEE International Conference on Very Large Scale Integration. – New York, 2012. – P.365.
31. Campbell S. Fabrication Engineering at the Micro and Nanoscale. -USA: Oxford University Press, 2007. –672p.
32. Ha YG, Everaerts K, Hersam MC, Marks TJ. Hybrid gate dielectric materials for unconventional electronic circuitry. // Accounts of chemical research. 2014. -Vol. 47, No. 4. -P.1019-1028.
33. Pipes LC, McGill L, Jahagirdar A. NMOS source-drain extension ion implantation into heated substrates. // InIon Implantation Technology (IIT), -2014. -P.1-6.
34. Lee CH, Lee GH, van der Zande A, Chen W, Li Y, Han M, Cui X, Araffe G, Nuckolls C, Heinz TF, Guo J. Charge transport and optoelectronic process in an atomically thin pn junction. // InAPS Meeting Abstracts 2014. Vol. 1. –P.676-681.

35. Mahmud MA, Subrina S. A two dimensional analytical model of drain to source current and subthreshold slope of a triple material double gate MOSFET. InElectrical and Computer Engineering (ICECE). -2014 –P.92-95.
36. Hakim M.M., De Groot C.H., Hall S, Ashburn P. Drain current multiplication in thin pillar vertical MOSFETs due to depletion isolation and charge coupling. Journal of Computational Electronics. -2016, -P.839-849.
37. Wang Z, Shi X, Tolbert LM, Wang F, Liang Z, Costinett D, Blalock BJ. Temperature-dependent short-circuit capability of silicon carbide power MOSFETs. IEEE Transactions on Power Electronics. -2016 –P.1555-1566.
38. Balakrishnan K, Cheng K, Hashemi P, Reznicek A. Channel-last replacement metal-gate vertical field effect transistor. International Business Machines Corporation. United States patent US 9,525,064. 2016.
39. Mao S, Wu T, Lu X, Popovic J, Ferreira J.A. Three-phase active front-end rectifier efficiency improvement with silicon carbide power semiconductor devices. InEnergy Conversion Congress and Exposition (ECCE), -2016 -P.1-8.
40. Terada K, Sanai K, Matsuoka S, Tsuji K. Effective channel length of MOSFET with halo. InIC Design & Technology (ICICDT), -2013 -P.65-68.
41. Jan C.H, Al-Amoody F, Chang H.Y, Chang T, Chen Y.W, Dias N, Hafez W, Ingerly D, Jang M, Karl E, Shi S.Y. A 14 nm SoC platform technology featuring 2 nd generation Tri-Gate transistors, 70 nm gate pitch, 52 nm metal pitch, and 0.0499 um 2 SRAM cells, optimized for low power, high performance and high density SoC products. InVLSI Technology (VLSI Technology), -2015 -P. T12-T13.
42. Sze SM, Ng KK. Physics of semiconductor devices. John wiley & sons, 2006, -832p.
43. Xie Q, Lee CJ, Xu J, Wann C, Sun JY, Taur Y. Comprehensive analysis of short-channel effects in ultrathin SOI MOSFETs. IEEE Transactions on Electron Devices. -2013, -P.1814-1819.
44. Yang X, Li Y, Xie X, Zhang S, Shen Z, Zhou F, He X, Zhao J, Mao G, Wang A, Ju J. Analysis of short channel effects for 14nm and beyond Si-bulk FinFET. InSemiconductor Technology International Conference (CSTIC). China, 2016 -P.1-3.

45. Holtij T, Graef M, Hain FM, Kloes A, Iniguez B. Compact model for short-channel junctionless accumulation mode double gate MOSFETs. IEEE Transactions on Electron Devices. -2014, –P.288-299.
46. Tsague HD, Twala B. INVESTIGATION OF CARRIER MOBILITY DEGRADATION EFFECTS ON MOSFET LEAKAGE SIMULATIONS. International Journal of Computing. -2016. –P.237-247.
47. Lundstrom MS, Antoniadis DA. Compact models and the physics of nanoscale FETs. IEEE Transactions on Electron Devices. -2014, -P.225-233.
48. Serhan A, Lauga-Larroze E, Bourdel S, Fournier JM, Corrao N. Comparison between MOS and bipolar mm-wave power amplifiers in advanced SiGe technologies. InBipolar/BiCMOS Circuits and Technology Meeting (BCTM), -2014, -P.159-162.
49. He L, Chiang TK, Liou JJ, Zheng W, Liu Z. A new analytical subthreshold potential/current model for quadruple-gate junctionless MOSFETs. IEEE Transactions on Electron Devices. -2014, –P.1972-1978.
50. Saramekala GK, Santra A, Kumar M, Dubey S, Jit S, Tiwari PK. Analytical subthreshold current and subthreshold swing models of short-channel dual-metal-gate (DMG) fully-depleted recessed-source/drain (Re-S/D) SOI MOSFETs. Journal of Computational Electronics. -2014, -P.467-476.
51. Mehrad M. Controlling floating body effect in high temperatures: L-shape SiGe region in nano-scale MOSFET. Superlattices and Microstructures. -2015. -P.573-580.
52. Bait AA, Narkhede N, More S, Satkut A, Joshi S. Performance enhancement and supression of short channel effects of 14nm double gate FETs by using gate stacked high-k dielectrics & workfunction variation. InInventive Computation Technologies (ICICT). -2016. –P.1-5.
53. Navarro C, Bawedin M, Andrieu F, Cristoloveanu S. Overestimation of Short-Channel Effects Due to Intergate Coupling in Advanced FD-SOI MOSFETs. IEEE Transactions on Electron Devices. 2014. –P.3274-3281.
54. Dasgupta A, Das R, Chakraborty S, Dutta A, Kundu A, Sarkar CK. Comparisons between Dual and Tri Material Gate on a 32 nm Double Gate MOSFET. Nano 11. 2016. -11p.

55. Liu H, Neal AT, Ye PD. Channel length scaling of MoS2 MOSFETs. ACS Nano. 2012. –P.8563-8569.
56. Chun JW, Chen CR. A novel leakage power reduction technique for CMOS circuit design. InSoC Design Conference (ISOCC), -2010. –P.119-122.
57. Verma A, Mehra R. Design and analysis of conventional and ratioed CMOS logic circuit. IOSR Journal of Engineering. -2013. –P.25-29.
58. Soudris D, Piguet C, Goutis C. Designing CMOS circuits for low power. Springer Publishing Company, Incorporated, 2010. -277p.
59. Rabaey JM, Pedram M, editors. Low power design methodologies. Springer Science & Business Media; 2012. -367p.
60. Ferry D. Semiconductors. IOP Publishing Ltd; 2013. -83p.
61. Olsen DM. *Performance-aware resource management of multi-threaded applications for many-core systems* (Doctoral dissertation, SOUTHERN ILLINOIS UNIVERSITY AT CARBONDALE). 2016. -53p.
62. Turner TE. Interconnect reliability assurance for circuits with billions of transistors. InPhysical and Failure Analysis of Integrated Circuits (IPFA), -2013. –P.149-152.
63. Neisser M, Wurm S. ITRS lithography roadmap: 2015 challenges. Advanced Optical Technologies. 2015. –P.235-240.
64. Meterelliyoz M, Al-amoody FH, Arslan U, Hamzaoglu F, Hood L, Lal M, Miller JL, Ramasundar A, Soltman D, Ifar W, Wang Y. 2 nd generation embedded DRAM with 4X lower self refresh power in 22nm Tri-Gate CMOS technology. InVLSI Circuits Digest of Technical Papers, 2014. –P.1-2.
65. Toledo LE, Petrashin PA, Lancioni WJ, Vazquez CD. Threshold voltage extraction circuit for low voltage CMOS design using basic long-channel MOSFET. InCircuits & Systems (LASCAS), -2015. -P.1-4.
66. Okuhara H, Kitamori K, Fujita Y, Usami K, Amano H. An optimal power supply and body bias voltage for a ultra low power micro-controller with silicon on thin box MOSFET. InLow Power Electronics and Design (ISLPED), -2015. –P.207-212.
67. Jeon DY, Pregl S, Park SJ, Baraban L, Cuniberti G, Mikolajick T, Weber WM. Scaling and Graphical Transport-Map Analysis of Ambipolar Schottky-Barrier Thin-Film

Transistors Based on a Parallel Array of Si Nanowires. Nano letters. -2015. –P.4578-4584.
68. Horowitz M. 1.1 computing's energy problem (and what we can do about it). InSolid-State Circuits Conference Digest of Technical Papers (ISSCC), -2014. –P.10-14.
69. Md Arshad MK, Othman N, Hashim U. Fully depletion of advanced silicon on insulator MOSFETs. Critical Reviews in Solid State and Materials Sciences. -2015. –P.182-196.
70. Chiang TK. A novel scaling theory for fully depleted, multiple-gate MOSFET, including effective number of gates (ENGs). IEEE Transactions on Electron Devices. -2014. –P.631-633.
71. Amiri PK, Alzate JG, Cai XQ, Ebrahimi F, Hu Q, Wong K, Grèzes C, Lee H, Yu G, Li X, Akyol M. Electric-field-controlled magnetoelectric RAM: progress, challenges, and scaling. IEEE Transactions on Magnetics. -2015, –P.1-7.
72. Avci UE, Morris DH, Young IA. Tunnel field-effect transistors: Prospects and challenges. IEEE Journal of the Electron Devices Society. -2015. –P.88-95.
73. Wu J, Min J, Taur Y. Short-channel effects in tunnel FETs. IEEE Transactions on Electron Devices. -2015. –P.3019-3024.
74. Yang W, Feng H, Fang X, Onozawa Y, Tanaka H, Sin JK. A novel edge termination structure for achieving the ideal planar junction breakdown voltage. InPower Semiconductor Devices and ICs (ISPSD), -2016. –P.287-290.
75. Chiang TK. A novel scaling theory for fully depleted omega-gate MOSFETs included with equivalent number of gates. IEEE Transactions on Electron Devices. 2014. –P.926-929.
76. Ilatikhameneh H, Klimeck G, Appenzeller J, Rahman R. Scaling theory of electrically doped 2D transistors. IEEE Electron Device Letters. 2015. –P.726-728.
77. Shin M, Shi M, Mouis M, Cros A, Josse E, Mukhopadhyay S, Piot B, Kim GT, Ghibaudo G. Experimental and theoretical investigation of magnetoresistance from linear regime to saturation in 14-nm FD-SOI MOS devices. IEEE Transactions on Electron Devices. -2015. –P.3-8.
78. Seo KI, Haran B, Gupta D, Guo D, Standaert T, Xie R, Shang H, Alptekin E, Bae DI, Bae G, Boye C. A 10nm platform technology for low power and high performance

application featuring FINFET devices with multi workfunction gate stack on bulk and SOI. InVLSI Technology (VLSI-Technology): Digest of Technical Papers, 2014. –P.1-2.

79. Petrosyants KO, Sambursky LM, Kharitonov IA, Lvov BG. Fault simulation in radiation-hardened SOI CMOS VLSIs using universal compact MOSFET model. InTest Symposium (LATS), -2016. –P.117-122.
80. Cristoloveanu S, Li S. Electrical characterization of silicon-on-insulator materials and devices. Springer Science & Business Media; 2013. -379p.
81. Mehrad M. Controlling floating body effect in high temperatures: L-shape SiGe region in nano-scale MOSFET. Superlattices and Microstructures. -2015. –P.573-580.
82. Märk TD, Dunn GH, editors. Electron impact ionization. Springer Science & Business Media; 2013. -384p.
83. Natarajan S, Agostinelli M, Akbar S, Bost M, Bowonder A, Chikarmane V, Chouksey S, Dasgupta A, Fischer K, Fu Q, Ghani T. A 14nm logic technology featuring 2 nd-generation FinFET, air-gapped interconnects, self-aligned double patterning and a 0.0588 μm 2 SRAM cell size. InElectron Devices Meeting (IEDM). -2014. –P.3-7.
84. Kushwah RS, Akashe S. Analysis of leakage reduction technique on FinFET based 7T and 8T SRAM cells. Radioelectronics and Communications Systems. -2014. –P.383-393.
85. Xie Q, Lin X, Wang Y, Chen S, Dousti MJ, Pedram M. Performance Comparisons Between 7-nm FinFET and Conventional Bulk CMOS Standard Cell Libraries. IEEE Transactions on Circuits and Systems II: Express Briefs. -2015. –P.761-765.
86. Gavrilov S, Ivanova G. Simultaneous logic and layout synthesis for Fin-fet based elements with regular layout in polysilicon and diffusion. -2015. –P.1-4.
87. Chaudhuri SM. Efficient device simulation and power optimization techniques for novel finfet circuit design. 2015. -176p.
88. Seo KI, Haran B, Gupta D, Guo D, Standaert T, Xie R, Shang H, Alptekin E, Bae DI, Bae G, Boye C. A 10nm platform technology for low power and high performance application featuring FINFET devices with multi workfunction gate stack on bulk and SOI. InVLSI Technology (VLSI-Technology): Digest of Technical Papers, -2014. –P.1-2.

89. Liao N, Cui X, Liao K, Ma K, Wu D, Wei W, Li R, Yu D. Low power adiabatic logic based on FinFETs. Science China information sciences. -2014. –P.1-3.
90. Lin X, Wang Y, Pedram M. Joint sizing and adaptive independent gate control for FinFET circuits operating in multiple voltage regimes using the logical effort method. -2013. –P.444-449.
91. Zimpeck AL, Meinhardt C, Posser G, Reis R. FinFET cells with different transistor sizing techniques against PVT variations. InCircuits and Systems (ISCAS), 2016. –P.45-48.
92. Xie Q, Lin X, Wang Y, Dousti MJ, Shafaei A, Ghasemi-Gol M, Pedram M. 5nm FinFET standard cell library optimization and circuit synthesis in near-and super-threshold voltage regimes. InVLSI (ISVLSI), -2014, –P.424-429.
93. Chaudhuri SM, Mishra P, Jha NK. Accurate Leakage/Delay Estimation for FinFET Standard Cells under PVT Variations using the Response Surface Methodology. ACM Journal on Emerging Technologies in Computing Systems (JETC). -2014. –P.19.
94. Burnett D, Parihar S, Ramamurthy H, Balasubramanian S. FinFET SRAM design challenges. InIC Design & Technology (ICICDT). -2014. –P.1-4.
95. Salahuddin SM, Kursun V. Asymmetrical FinFET SRAM cells with wider read noise margin and lower leakage currents. -2015. –P.1-3.
96. Lee JO, Gupta A, Kim DH, inventors; Windbound Electronics Corporation, assignee. NAND flash memory array architecture having low read latency and low program disturb. United States patent US 9,245,639. 2016 Jan 26.
97. Natsui M, Suzuki D, Sakimura N, Nebashi R, Tsuji Y, Morioka A, Sugibayashi T, Miura S, Honjo H, Kinoshita K, Ikeda S. Nonvolatile logic-in-memory array processor in 90nm MTJ/MOS achieving 75% leakage reduction using cycle-based power gating. InSolid-State Circuits Conference Digest of Technical Papers (ISSCC), -2013. –P.194-195.
98. Urteaga M, Hacker J, Griffith Z, Young A, Pierson R, Rowell P, Seo M, Rodwell MJ. A 130 nm InP HBT integrated circuit technology for THz electronics. InElectron Devices Meeting (IEDM), -2016. –P.29-32.

99. Ha D, Lee K, Kang J. Energy efficiency analysis with circuit power consumption in massive MIMO systems. InPersonal Indoor and Mobile Radio Communications (PIMRC), -2013. –P.938-942.
100. Bahr A, Saleh LA, Hinsch R, Schroeder D, Isbrandt D, Krautschneider WH. Small area, low power neural recording integrated circuit in 130 nm CMOS technology for small mammalians. InMicroelectronics (ICM), -2016. –P.349-352.
101. Merino JL, Bota SA, Picos R, Segura J. Alternate characterization technique for static random-access memory static noise margin determination. International Journal of Circuit Theory and Applications. -2013. –P.1085-1096.
102. Zhao J, Mutlu O, Xie Y. FIRM: Fair and high-performance memory control for persistent memory systems. InMicroarchitecture (MICRO), -2014. –P.153-165.
103. Joshi VK, Borkar S. A comparative study of NC and PP-SRAM cells with 6T SRAM cell using 45nm CMOS technology. InAdvances in Electrical, Electronic and Systems Engineering (ICAEES). -2016 –P.58-62.
104. Gong CS. Investigation of Noise-Margin-Enhanced and Low-Power Memory Techniques for SoC Applications. Circuits, Systems, and Signal Processing. -2015. –P.1115-1128.
105. Alorda B, Torrens G, Bota S, Segura J. Adaptive static and dynamic noise margin improvement in minimum-sized 6T-SRAM cells. Microelectronics Reliability. 2014. –P.2613-2620.
106. Sinangil ME, Poulton JW, Fojtik MR, Greer III TH, Tell SG, Gotterba AJ, Wang J, Golbus J, Zimmer B, Dally WJ, Gray CT. A 28 nm 2 Mbit 6 T SRAM with highly configurable low-voltage write-ability assist implementation and capacitor-based sense-amplifier input offset compensation. IEEE Journal of Solid-State Circuits. -2016. –P.557-567.
107. Gosla AF, Mondal C, inventors; Western Digital Technologies, Inc., assignee. Efficient retry mechanism for solid-state memory failures. United States patent US 8,788,880. 2014 Jul 22.
108. Manukyan S. Design of Finfet memory cells // Engineering academy of Armenia, -2017. -P.527-531.

109. Մելիքյան Վ., Խաժակյան Տ., Հախվերդյան Տ., Մանուկյան Ս. Տարբեր հաճախականություններով տակտավորվող թվային սխեմաների համաձայնեցման եղանակ // ՀՀ ԳԱԱ և ՀՊՃՀ տեղեկագիր. Տեխնիկական գիտությունների սերիա.-2016.- Հատոր 69, N4, ISSN 0002-306X.- էջ 381-392:
110. Melikyan V, Babayan E, Khazhakyan T, Manukyan S. Analysis of the impact of metastability phenomenon on the latency and power consumption of synchronizer circuits. InEast-West Design & Test Symposium (EWDTS). -2016. –P.100-102.
111. Krrikyan H, Hovhannisyan T, Manukyan S. Prototyping system for USB3. 0 link layer using synthesizable assertions and partial reconfiguration. // InComputer Science and Information Technologies (CSIT), 2015 Sep 28.-P. 19-22.
112. Melikyan V, Hakhverdyan T, Manukyan S, Gevorgyan A, Babayan D. Low power OpenRISC processor with power gating, multi-VTH and multi-voltage techniques. //InEast-West Design & Test Symposium (EWDTS), 2016 Oct 14.-P. 1-4
113. Melikyan V, Aleksanyan A, Harutyunyan A, Galstyan V, Manukyan S, Ilijin S. Low-power And High Speed High-to-Low level Shifter. In2nd International Conference on Electrical, Electronic and Computing Engineering IcETRAN Silver Lake (Srebrno Jezero) -2015. –P.1-4.
114. Trdatyan D., Trdatyan A., Manukyan S. CMOS master slave flip-flop for low power applications // Semiconductor Micro- & nanoelectronics. -2015. –P.238-341.
115. Andolfatto L, Euzenat C, Vagnoni E, Münch-Alligné C, Avellan F. A mixed standard/custom design strategy to minimize cost and maximize efficiency for picohydro power potential harvesting. InEnergy (IYCE), -2015. –P.1-8.
116. Paydavosi N, Venugopalan S, Chauhan YS, Duarte JP, Jandhyala S, Niknejad AM, Hu CC. BSIM—SPICE models enable FinFET and UTB IC designs. IEEE Access. -2013. –P.201-215.
117. Mitra J, Torres A, Pan DZ. Process, design rule, and layout co-optimization for DSA based patterning of sub-10nm Finfet devices. International Society for Optics and Photonics. -2017, –P.101440G-101440G.
118. Bhattacharya D, Bhoj AN, Jha NK. Design of efficient content addressable memories in high-performance FinFET technology. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. -2015, –P.963-967.

119. Huijsing J, van de Plassche RJ, Sansen W, editors. Analog circuit design: low-noise, low-power, low-voltage; mixed-mode design with CAD tools; voltage, current and time references. Springer Science & Business Media; 2013. -422p.
120. Ziesemer Jr A, Reis R, Moreira MT, Arendt ME, Calazans NL. A design flow for physical synthesis of digital cells with ASTRAN. -2014 –P.245-246.
121. http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=954909
122. https://www.doulos.com/knowhow/vhdl_designers_guide/what_is_vhdl/
123. https://www.python.org/
124. https://www.scipy.org/
125. http://mysql-python.sourceforge.net/MySQLdb.html
126. https://wiki.python.org/moin/PyQt
127. https://www.linux.com
128. http://www.eecg.toronto.edu/~kphang/teaching/spice/

ՀԱՎԵԼՎԱԾ 1. ՆԵՐԴՐՄԱՆ ԱԿՏ

№ 347/17 "28" 04 2017

Հաստատում եմ՝
«ՍԻՆՈՓՍԻՍ ԱՐՄԵՆԻԱ» ՓԲԸ
Գլխավոր տնօրեն՝
Հ. Մուսայելյան

Սերգեյ Հակոբի Մանուկյանի «Եռաչափ տրանզիստորներով միկրոէլեկտրոնային սխեմաների նախագծման միջոցների մշակումը» թեմայով թեկնածուական ատենախոսության արդյունքների

ՆԵՐԴՐՄԱՆ ԱԿՏ

ԵՊՀ «Էլեկտրոնիկա, միկրո և նանոէլեկտրոնիկա» մասնագիտությամբ 3-րդ կուրսի ասպիրանտ Ս. Հ. Մանուկյանի կողմից մշակված նախագծման 3D-Compiler ծրագրային միջոցը ներդրված է «ՍԻՆՈՓՍԻՍ ԱՐՄԵՆԻԱ» ՓԲԸ-ում: Այն օգտագործվում է մեծ ստատիկ աղմկակայունության պաշար պահանջող հիշողության մատրիցների, բարձր արագագործությամբ դոմինո փականների եւ հուսալի ստատիկ տրիգերների նախագծման համար, երբ անհրաժեշտ են սխեմայի մակրոբնութագրերի փոխզիջումային արժեքներ: Մշակված 3D-Compiler ծրագրային միջոցի կիրառումը զգալիորեն պարզեցրել է նման համակարգերի նախագծումը: Ծրագրային միջոցը կազմում է եռաչափ տրանզիստորներով ինտեգրալ սխեմաների նախագծման գործընթացի կարևոր մասը և բավարարում է ժամանակակից էլեկտրոնային նախագծման բնագավառում կիրառվող ծրագրային գործիքային միջոցին ներկայացվող բոլոր պահանջներին:

Հիշողության սխեմաների Ֆիզիկական նախագծման դեպարտամենտի տնօրեն՝

Ա. Հովհաննիսյան

"ՍԻՆՈՓՍԻՍ ԱՐՄԵՆԻԱ" ՓԲԸ
0026, ՀՀ, ԵՐԵՎԱՆ, ԱՐՇԱԿՈՒՆՅԱՑ 41
ՀԵՌ. (+374 10) 49 21 00, ՖԱՔՍ (+374 10) 49 26 96
ՀՎՀՀ 02236362

"SYNOPSYS ARMENIA" CJSC
41 ARSHAKUNYATS AVE., YEREVAN, ARMENIA, 0026
TEL.: (+374 10) 49 21 00, FAX: (+374 10) 49 26 96
TAX PAYER'S ID 02236362

ՀԱՎԵԼՎԱԾ 2. 3D-Compiler ծրագրային գործիքի ելքային SPICE ֆայլերը

Բարձր շեմային լարմամբ հասանելիության տրանզիստորներով հիշողության բջիջը

Xm1 net123 net341 net211 vdd vss nfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm2 net23 net31 net12 vdd vss nfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm3 net13 net1 net12 vdd vss nfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm4 net12 net37 net10 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm5 net131 net3 net20 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm6 net1 net4 net6 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m

P տիպի հասանելիության տրանզիստորներով հիշողության բջիջը

Xm1 net1 net3 net21 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm2 net33 net321 net2 vdd vss pfet w=0.1u l=28n nf=1 nfin=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm3 net135 net15 net12 vdd vss nfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m

Xm4 net12 net37 net10 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm5 net11 net39 net21 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm6 net11 net24 net61 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m

Շեմային լարման դինամիկ ղեկավարմամբ հիշողության բջիջը

Xm1 net100 net3 net1 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm2 net23 net21 net55 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm3 net75 net25 net22 vdd vss nfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm4 net19 net7 net16 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm5 net10 net3 net214 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm6 net111 net94 net21 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m

Ներկառուցված հետադարձ կապով հաստանելիության տրանզիստորներով հիշողության բջիջը

Xm1 net110 net53 net31 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m

Xm2 net2 net51 net52 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm3 net5 net2 net26 vdd vss nfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm4 net18 net76 net36 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm5 net10 net76 net222 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm6 net11 net4 net1 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm7 net13 net26 net22 vdd vss nfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm8 net190 net7 net126 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm9 net17 net93 net24 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm10 net11 net9 net7 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='57f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m

Դոմինո տրամաբանական սխեմաների լիցքի պաշտպանման հանգույցը

Xm1 net110 net53 net31 vdd vss pfet w=0.1u l=28n nfin=1 nf=3 multi=1 ad='50f' as='57f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm2 net2 net51 net52 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=2 ad='50f' as='50f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m

Xm3 net15 net29 net226 vdd vss nfet w=0.1u l=28n nfin=1 nf=3 multi=1 ad='50f' as='50f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm4 net13 net46 net346 vdd vss pfet w=0.1u l=28n nfin=1 nf=3 multi=1 ad='50f' as='50f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm5 net11 net7 net22 vdd vss pfet w=0.1u l=28n nfin=1 nf=5 multi=1 ad='50f' as='50f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm6 net1 net48 net15 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='50f' as='50f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xm7 net13 net6 net2 vdd vss nfet w=0.1u l=28n nf=1 multi=1 ad='50f' as='50f'

Ստատիկ տրիգերի հետադարձ կապի շղթան

Xmp net1 net3 net34 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=1 ad='60f' as='60f'
+ pd=1.9u ps=2.24u nrd=35m nrs=35m
Xmn net4 net1 net5 vdd vss pfet w=0.1u l=28n nfin=1 nf=1 multi=2 ad='60f' as='60f'

ՀԱՎԵԼՎԱԾ 3. 3D-Compiler ԾՐԱԳՐԱՅԻՆ ԳՈՐԾԻՔԻ ՆԿԱՐԱԳՐՈՒԹՅՈՒՆԸ

```
from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1161, 734)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.line = QtWidgets.QFrame(self.centralwidget)
        self.line.setGeometry(QtCore.QRect(10, 300, 1141, 16))
        self.line.setFrameShape(QtWidgets.QFrame.HLine)
        self.line.setFrameShadow(QtWidgets.QFrame.Sunken)
        self.line.setObjectName("line")
        self.groupBox = QtWidgets.QGroupBox(self.centralwidget)
        self.groupBox.setGeometry(QtCore.QRect(10, 10, 381, 281))
        font = QtGui.QFont()
        font.setPointSize(14)
        self.groupBox.setFont(font)
        self.groupBox.setObjectName("groupBox")
        self.label = QtWidgets.QLabel(self.groupBox)
        self.label.setGeometry(QtCore.QRect(20, 30, 131, 31))
        font = QtGui.QFont()
        font.setPointSize(12)
        self.label.setFont(font)
        self.label.setObjectName("label")
        self.comboBox = QtWidgets.QComboBox(self.groupBox)
        self.comboBox.setGeometry(QtCore.QRect(160, 36, 201, 22))
        font = QtGui.QFont()
        font.setPointSize(8)
        self.comboBox.setFont(font)
        self.comboBox.setEditable(False)
        self.comboBox.setMaxVisibleItems(11)
        self.comboBox.setMinimumContentsLength(0)
        self.comboBox.setObjectName("comboBox")
        self.comboBox.addItem("")
        self.comboBox.addItem("")
        self.comboBox.addItem("")
        self.comboBox.addItem("")
        self.label_2 = QtWidgets.QLabel(self.groupBox)
        self.label_2.setGeometry(QtCore.QRect(20, 80, 111, 31))
        font = QtGui.QFont()
        font.setPointSize(11)
        self.label_2.setFont(font)
        self.label_2.setObjectName("label_2")
```

```
self.spinBox = QtWidgets.QSpinBox(self.groupBox)
self.spinBox.setGeometry(QtCore.QRect(131, 85, 61, 22))
self.spinBox.setMaximum(999)
self.spinBox.setObjectName("spinBox")
self.spinBox_2 = QtWidgets.QSpinBox(self.groupBox)
self.spinBox_2.setGeometry(QtCore.QRect(300, 85, 61, 22))
self.spinBox_2.setMaximum(999)
self.spinBox_2.setObjectName("spinBox_2")
self.label_3 = QtWidgets.QLabel(self.groupBox)
self.label_3.setGeometry(QtCore.QRect(210, 80, 91, 31))
font = QtGui.QFont()
font.setPointSize(11)
self.label_3.setFont(font)
self.label_3.setObjectName("label_3")
self.label_4 = QtWidgets.QLabel(self.groupBox)
self.label_4.setGeometry(QtCore.QRect(17, 164, 91, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.label_4.setFont(font)
self.label_4.setObjectName("label_4")
self.lineEdit = QtWidgets.QLineEdit(self.groupBox)
self.lineEdit.setGeometry(QtCore.QRect(110, 170, 161, 21))
font = QtGui.QFont()
font.setPointSize(12)
self.lineEdit.setFont(font)
self.lineEdit.setObjectName("lineEdit")
self.line_2 = QtWidgets.QFrame(self.groupBox)
self.line_2.setGeometry(QtCore.QRect(20, 64, 341, 16))
self.line_2.setFrameShape(QtWidgets.QFrame.HLine)
self.line_2.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line_2.setObjectName("line_2")
self.label_5 = QtWidgets.QLabel(self.groupBox)
self.label_5.setGeometry(QtCore.QRect(20, 120, 101, 31))
font = QtGui.QFont()
font.setPointSize(11)
self.label_5.setFont(font)
self.label_5.setObjectName("label_5")
self.radioButton = QtWidgets.QRadioButton(self.groupBox)
self.radioButton.setGeometry(QtCore.QRect(130, 126, 91, 21))
font = QtGui.QFont()
font.setPointSize(10)
self.radioButton.setFont(font)
self.radioButton.setObjectName("radioButton")
self.radioButton_2 = QtWidgets.QRadioButton(self.groupBox)
self.radioButton_2.setGeometry(QtCore.QRect(231, 126, 91, 21))
font = QtGui.QFont()
font.setPointSize(10)
self.radioButton_2.setFont(font)
self.radioButton_2.setObjectName("radioButton_2")
self.line_3 = QtWidgets.QFrame(self.groupBox)
```

```
self.line_3.setGeometry(QtCore.QRect(20, 150, 341, 16))
self.line_3.setFrameShape(QtWidgets.QFrame.HLine)
self.line_3.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line_3.setObjectName("line_3")
self.pushButton = QtWidgets.QPushButton(self.groupBox)
self.pushButton.setGeometry(QtCore.QRect(280, 170, 75, 23))
font = QtGui.QFont()
font.setPointSize(11)
self.pushButton.setFont(font)
self.pushButton.setObjectName("pushButton")
self.line_4 = QtWidgets.QFrame(self.groupBox)
self.line_4.setGeometry(QtCore.QRect(20, 200, 341, 16))
self.line_4.setFrameShape(QtWidgets.QFrame.HLine)
self.line_4.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line_4.setObjectName("line_4")
self.progressBar = QtWidgets.QProgressBar(self.groupBox)
self.progressBar.setGeometry(QtCore.QRect(50, 250, 281, 23))
self.progressBar.setProperty("value", 35)
self.progressBar.setTextVisible(False)
self.progressBar.setObjectName("progressBar")
self.pushButton_2 = QtWidgets.QPushButton(self.groupBox)
self.pushButton_2.setGeometry(QtCore.QRect(111, 214, 161, 31))
palette = QtGui.QPalette()
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)
brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)
brush = QtGui.QBrush(QtGui.QColor(6, 224, 121))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Light, brush)
brush = QtGui.QBrush(QtGui.QColor(5, 186, 101))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Midlight, brush)
brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Dark, brush)
brush = QtGui.QBrush(QtGui.QColor(2, 99, 54))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Mid, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Text, brush)
brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.BrightText, brush)
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ButtonText, brush)
```

```
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Base, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Window, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Shadow, brush)
        brush = QtGui.QBrush(QtGui.QColor(129, 202, 168))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.AlternateBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ToolTipBase, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ToolTipText, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.WindowText, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Button, brush)
        brush = QtGui.QBrush(QtGui.QColor(6, 224, 121))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Light, brush)
        brush = QtGui.QBrush(QtGui.QColor(5, 186, 101))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Midlight, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Dark, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 99, 54))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Mid, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Text, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.BrightText, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ButtonText, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Base, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
```

```
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Window, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Shadow, brush)
        brush = QtGui.QBrush(QtGui.QColor(129, 202, 168))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.AlternateBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ToolTipBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ToolTipText,
brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.WindowText, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Button, brush)
        brush = QtGui.QBrush(QtGui.QColor(6, 224, 121))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Light, brush)
        brush = QtGui.QBrush(QtGui.QColor(5, 186, 101))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Midlight, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Dark, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 99, 54))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Mid, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Text, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.BrightText, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ButtonText, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Base, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Window, brush)
```

```
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Shadow, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.AlternateBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ToolTipBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ToolTipText,
brush)
        self.pushButton_2.setPalette(palette)
        font = QtGui.QFont()
        font.setPointSize(12)
        self.pushButton_2.setFont(font)
        self.pushButton_2.setObjectName("pushButton_2")
        self.groupBox_2 = QtWidgets.QGroupBox(self.centralwidget)
        self.groupBox_2.setGeometry(QtCore.QRect(400, 10, 381, 281))
        font = QtGui.QFont()
        font.setPointSize(14)
        self.groupBox_2.setFont(font)
        self.groupBox_2.setObjectName("groupBox_2")
        self.lineEdit_2 = QtWidgets.QLineEdit(self.groupBox_2)
        self.lineEdit_2.setGeometry(QtCore.QRect(110, 80, 161, 21))
        font = QtGui.QFont()
        font.setPointSize(12)
        self.lineEdit_2.setFont(font)
        self.lineEdit_2.setObjectName("lineEdit_2")
        self.pushButton_3 = QtWidgets.QPushButton(self.groupBox_2)
        self.pushButton_3.setGeometry(QtCore.QRect(280, 80, 75, 23))
        font = QtGui.QFont()
        font.setPointSize(11)
        self.pushButton_3.setFont(font)
        self.pushButton_3.setObjectName("pushButton_3")
        self.label_7 = QtWidgets.QLabel(self.groupBox_2)
        self.label_7.setGeometry(QtCore.QRect(17, 74, 91, 31))
        font = QtGui.QFont()
        font.setPointSize(12)
        self.label_7.setFont(font)
        self.label_7.setObjectName("label_7")
        self.pushButton_4 = QtWidgets.QPushButton(self.groupBox_2)
        self.pushButton_4.setGeometry(QtCore.QRect(280, 170, 75, 23))
        font = QtGui.QFont()
        font.setPointSize(11)
        self.pushButton_4.setFont(font)
        self.pushButton_4.setObjectName("pushButton_4")
```

```
self.label_8 = QtWidgets.QLabel(self.groupBox_2)
self.label_8.setGeometry(QtCore.QRect(17, 164, 91, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.label_8.setFont(font)
self.label_8.setObjectName("label_8")
self.lineEdit_3 = QtWidgets.QLineEdit(self.groupBox_2)
self.lineEdit_3.setGeometry(QtCore.QRect(110, 170, 161, 21))
font = QtGui.QFont()
font.setPointSize(12)
self.lineEdit_3.setFont(font)
self.lineEdit_3.setObjectName("lineEdit_3")
self.label_9 = QtWidgets.QLabel(self.groupBox_2)
self.label_9.setGeometry(QtCore.QRect(17, 114, 91, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.label_9.setFont(font)
self.label_9.setObjectName("label_9")
self.pushButton_5 = QtWidgets.QPushButton(self.groupBox_2)
self.pushButton_5.setGeometry(QtCore.QRect(280, 120, 75, 23))
font = QtGui.QFont()
font.setPointSize(11)
self.pushButton_5.setFont(font)
self.pushButton_5.setObjectName("pushButton_5")
self.lineEdit_4 = QtWidgets.QLineEdit(self.groupBox_2)
self.lineEdit_4.setGeometry(QtCore.QRect(110, 120, 161, 21))
font = QtGui.QFont()
font.setPointSize(12)
self.lineEdit_4.setFont(font)
self.lineEdit_4.setObjectName("lineEdit_4")
self.line_5 = QtWidgets.QFrame(self.groupBox_2)
self.line_5.setGeometry(QtCore.QRect(10, 143, 361, 20))
self.line_5.setFrameShape(QtWidgets.QFrame.HLine)
self.line_5.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line_5.setObjectName("line_5")
self.line_6 = QtWidgets.QFrame(self.groupBox_2)
self.line_6.setGeometry(QtCore.QRect(20, 200, 341, 16))
self.line_6.setFrameShape(QtWidgets.QFrame.HLine)
self.line_6.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line_6.setObjectName("line_6")
self.pushButton_6 = QtWidgets.QPushButton(self.groupBox_2)
self.pushButton_6.setGeometry(QtCore.QRect(111, 214, 161, 31))
palette = QtGui.QPalette()
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)
brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)
brush = QtGui.QBrush(QtGui.QColor(6, 224, 121))
```

```
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Light, brush)
        brush = QtGui.QBrush(QtGui.QColor(5, 186, 101))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Midlight, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Dark, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 99, 54))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Mid, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Text, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.BrightText, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ButtonText, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Base, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Window, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Shadow, brush)
        brush = QtGui.QBrush(QtGui.QColor(129, 202, 168))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.AlternateBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ToolTipBase, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ToolTipText, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.WindowText, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Button, brush)
        brush = QtGui.QBrush(QtGui.QColor(6, 224, 121))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Light, brush)
        brush = QtGui.QBrush(QtGui.QColor(5, 186, 101))
        brush.setStyle(QtCore.Qt.SolidPattern)
```

```
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Midlight, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Dark, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 99, 54))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Mid, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Text, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.BrightText, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ButtonText, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Base, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Window, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Shadow, brush)
        brush = QtGui.QBrush(QtGui.QColor(129, 202, 168))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.AlternateBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ToolTipBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ToolTipText,
brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.WindowText, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Button, brush)
        brush = QtGui.QBrush(QtGui.QColor(6, 224, 121))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Light, brush)
        brush = QtGui.QBrush(QtGui.QColor(5, 186, 101))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Midlight, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
```

```
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Dark, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 99, 54))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Mid, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Text, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.BrightText, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ButtonText, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Base, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Window, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Shadow, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.AlternateBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ToolTipBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ToolTipText,
brush)
        self.pushButton_6.setPalette(palette)
        font = QtGui.QFont()
        font.setPointSize(12)
        self.pushButton_6.setFont(font)
        self.pushButton_6.setObjectName("pushButton_6")
        self.progressBar_2 = QtWidgets.QProgressBar(self.groupBox_2)
        self.progressBar_2.setGeometry(QtCore.QRect(50, 250, 281, 23))
        self.progressBar_2.setProperty("value", 20)
        self.progressBar_2.setTextVisible(False)
        self.progressBar_2.setObjectName("progressBar_2")
        self.groupBox_3 = QtWidgets.QGroupBox(self.centralwidget)
        self.groupBox_3.setGeometry(QtCore.QRect(790, 10, 361, 281))
        font = QtGui.QFont()
        font.setPointSize(14)
        self.groupBox_3.setFont(font)
```

```
self.groupBox_3.setObjectName("groupBox_3")
self.label_10 = QtWidgets.QLabel(self.groupBox_3)
self.label_10.setGeometry(QtCore.QRect(20, 110, 91, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.label_10.setFont(font)
self.label_10.setObjectName("label_10")
self.lineEdit_5 = QtWidgets.QLineEdit(self.groupBox_3)
self.lineEdit_5.setGeometry(QtCore.QRect(113, 116, 161, 21))
font = QtGui.QFont()
font.setPointSize(12)
self.lineEdit_5.setFont(font)
self.lineEdit_5.setObjectName("lineEdit_5")
self.pushButton_7 = QtWidgets.QPushButton(self.groupBox_3)
self.pushButton_7.setGeometry(QtCore.QRect(283, 166, 75, 23))
font = QtGui.QFont()
font.setPointSize(11)
self.pushButton_7.setFont(font)
self.pushButton_7.setObjectName("pushButton_7")
self.label_11 = QtWidgets.QLabel(self.groupBox_3)
self.label_11.setGeometry(QtCore.QRect(20, 70, 91, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.label_11.setFont(font)
self.label_11.setObjectName("label_11")
self.pushButton_8 = QtWidgets.QPushButton(self.groupBox_3)
self.pushButton_8.setGeometry(QtCore.QRect(283, 76, 75, 23))
font = QtGui.QFont()
font.setPointSize(11)
self.pushButton_8.setFont(font)
self.pushButton_8.setObjectName("pushButton_8")
self.pushButton_9 = QtWidgets.QPushButton(self.groupBox_3)
self.pushButton_9.setGeometry(QtCore.QRect(283, 116, 75, 23))
font = QtGui.QFont()
font.setPointSize(11)
self.pushButton_9.setFont(font)
self.pushButton_9.setObjectName("pushButton_9")
self.line_7 = QtWidgets.QFrame(self.groupBox_3)
self.line_7.setGeometry(QtCore.QRect(10, 200, 341, 16))
self.line_7.setFrameShape(QtWidgets.QFrame.HLine)
self.line_7.setFrameShadow(QtWidgets.QFrame.Sunken)
self.line_7.setObjectName("line_7")
self.pushButton_10 = QtWidgets.QPushButton(self.groupBox_3)
self.pushButton_10.setGeometry(QtCore.QRect(101, 214, 161, 31))
palette = QtGui.QPalette()
brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
brush.setStyle(QtCore.Qt.SolidPattern)
palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.WindowText, brush)
brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
brush.setStyle(QtCore.Qt.SolidPattern)
```

```
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Button, brush)
        brush = QtGui.QBrush(QtGui.QColor(6, 224, 121))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Light, brush)
        brush = QtGui.QBrush(QtGui.QColor(5, 186, 101))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Midlight, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Dark, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 99, 54))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Mid, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Text, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.BrightText, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ButtonText, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Base, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Window, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.Shadow, brush)
        brush = QtGui.QBrush(QtGui.QColor(129, 202, 168))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.AlternateBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ToolTipBase, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Active, QtGui.QPalette.ToolTipText, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.WindowText, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Button, brush)
        brush = QtGui.QBrush(QtGui.QColor(6, 224, 121))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Light, brush)
```

```
        brush = QtGui.QBrush(QtGui.QColor(5, 186, 101))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Midlight, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Dark, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 99, 54))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Mid, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Text, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.BrightText, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ButtonText, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Base, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Window, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.Shadow, brush)
        brush = QtGui.QBrush(QtGui.QColor(129, 202, 168))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.AlternateBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ToolTipBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Inactive, QtGui.QPalette.ToolTipText,
brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.WindowText, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Button, brush)
        brush = QtGui.QBrush(QtGui.QColor(6, 224, 121))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Light, brush)
        brush = QtGui.QBrush(QtGui.QColor(5, 186, 101))
        brush.setStyle(QtCore.Qt.SolidPattern)
```

```
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Midlight, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Dark, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 99, 54))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Mid, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Text, brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 255))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.BrightText, brush)
        brush = QtGui.QBrush(QtGui.QColor(2, 74, 40))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ButtonText, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Base, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Window, brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.Shadow, brush)
        brush = QtGui.QBrush(QtGui.QColor(4, 149, 81))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.AlternateBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(255, 255, 220))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ToolTipBase,
brush)
        brush = QtGui.QBrush(QtGui.QColor(0, 0, 0))
        brush.setStyle(QtCore.Qt.SolidPattern)
        palette.setBrush(QtGui.QPalette.Disabled, QtGui.QPalette.ToolTipText,
brush)
        self.pushButton_10.setPalette(palette)
        font = QtGui.QFont()
        font.setPointSize(12)
        self.pushButton_10.setFont(font)
        self.pushButton_10.setObjectName("pushButton_10")
        self.progressBar_3 = QtWidgets.QProgressBar(self.groupBox_3)
        self.progressBar_3.setGeometry(QtCore.QRect(40, 250, 281, 23))
        self.progressBar_3.setProperty("value", 64)
        self.progressBar_3.setTextVisible(False)
        self.progressBar_3.setObjectName("progressBar_3")
        self.lineEdit_6 = QtWidgets.QLineEdit(self.groupBox_3)
        self.lineEdit_6.setGeometry(QtCore.QRect(113, 76, 161, 21))
        font = QtGui.QFont()
```

```
font.setPointSize(12)
self.lineEdit_6.setFont(font)
self.lineEdit_6.setObjectName("lineEdit_6")
self.lineEdit_7 = QtWidgets.QLineEdit(self.groupBox_3)
self.lineEdit_7.setGeometry(QtCore.QRect(113, 166, 161, 21))
font = QtGui.QFont()
font.setPointSize(12)
self.lineEdit_7.setFont(font)
self.lineEdit_7.setObjectName("lineEdit_7")
self.label_12 = QtWidgets.QLabel(self.groupBox_3)
self.label_12.setGeometry(QtCore.QRect(20, 160, 91, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.label_12.setFont(font)
self.label_12.setObjectName("label_12")
self.tabWidget = QtWidgets.QTabWidget(self.centralwidget)
self.tabWidget.setGeometry(QtCore.QRect(10, 310, 1141, 381))
font = QtGui.QFont()
font.setPointSize(12)
self.tabWidget.setFont(font)
self.tabWidget.setObjectName("tabWidget")
self.tab = QtWidgets.QWidget()
self.tab.setObjectName("tab")
self.textEdit_2 = QtWidgets.QTextEdit(self.tab)
self.textEdit_2.setGeometry(QtCore.QRect(0, 0, 1141, 351))
self.textEdit_2.setObjectName("textEdit_2")
self.tabWidget.addTab(self.tab, "")
self.tab_2 = QtWidgets.QWidget()
self.tab_2.setObjectName("tab_2")
self.textEdit_3 = QtWidgets.QTextEdit(self.tab_2)
self.textEdit_3.setGeometry(QtCore.QRect(0, 0, 1141, 351))
self.textEdit_3.setObjectName("textEdit_3")
self.tabWidget.addTab(self.tab_2, "")
self.tab_3 = QtWidgets.QWidget()
self.tab_3.setObjectName("tab_3")
self.textEdit = QtWidgets.QTextEdit(self.tab_3)
self.textEdit.setGeometry(QtCore.QRect(0, 0, 1131, 351))
self.textEdit.setObjectName("textEdit")
self.tabWidget.addTab(self.tab_3, "")
MainWindow.setCentralWidget(self.centralwidget)
self.menubar = QtWidgets.QMenuBar(MainWindow)
self.menubar.setGeometry(QtCore.QRect(0, 0, 1161, 21))
self.menubar.setObjectName("menubar")
MainWindow.setMenuBar(self.menubar)
self.statusbar = QtWidgets.QStatusBar(MainWindow)
self.statusbar.setObjectName("statusbar")
MainWindow.setStatusBar(self.statusbar)

self.retranslateUi(MainWindow)
self.comboBox.setCurrentIndex(0)
```

```
        self.tabWidget.setCurrentIndex(0)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.groupBox.setTitle(_translate("MainWindow", "Memory Compilation"))
        self.label.setText(_translate("MainWindow", "Choose Method:"))
        self.comboBox.setItemText(0, _translate("MainWindow", "IG Single-gate
Access Transistors"))
        self.comboBox.setItemText(1, _translate("MainWindow", "SG P-Type Access
Transistors"))
        self.comboBox.setItemText(2, _translate("MainWindow", "Full IG Access
Transistors"))
        self.comboBox.setItemText(3, _translate("MainWindow", "Embedded Feedback
Access Transistors"))
        self.label_2.setText(_translate("MainWindow", "No. of Columns:"))
        self.label_3.setText(_translate("MainWindow", "No. of Rows:"))
        self.label_4.setText(_translate("MainWindow", "Output File:"))
        self.label_5.setText(_translate("MainWindow", "Memory Type:"))
        self.radioButton.setText(_translate("MainWindow", "Single input"))
        self.radioButton_2.setText(_translate("MainWindow", "Dual input"))
        self.pushButton.setText(_translate("MainWindow", "Browse"))
        self.pushButton_2.setText(_translate("MainWindow", "Compile"))
        self.groupBox_2.setTitle(_translate("MainWindow", "Domino Compilation"))
        self.pushButton_3.setText(_translate("MainWindow", "Browse"))
        self.label_7.setText(_translate("MainWindow", "Input File:"))
        self.pushButton_4.setText(_translate("MainWindow", "Browse"))
        self.label_8.setText(_translate("MainWindow", "Output File:"))
        self.label_9.setText(_translate("MainWindow", "Config File:"))
        self.pushButton_5.setText(_translate("MainWindow", "Browse"))
        self.pushButton_6.setText(_translate("MainWindow", "Compile"))
        self.groupBox_3.setTitle(_translate("MainWindow", "Latch Compilation"))
        self.label_10.setText(_translate("MainWindow", "Config File:"))
        self.pushButton_7.setText(_translate("MainWindow", "Browse"))
        self.label_11.setText(_translate("MainWindow", "Input File:"))
        self.pushButton_8.setText(_translate("MainWindow", "Browse"))
        self.pushButton_9.setText(_translate("MainWindow", "Browse"))
        self.pushButton_10.setText(_translate("MainWindow", "Compile"))
        self.label_12.setText(_translate("MainWindow", "Output File:"))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab),
_translate("MainWindow", "Memory Compilation"))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_2),
_translate("MainWindow", "Domino Compilation"))
        self.tabWidget.setTabText(self.tabWidget.indexOf(self.tab_3),
_translate("MainWindow", "Latch Compilation"))
```

ՀԱՎԵԼՎԱԾ 4. ՆԿԱՐՆԵՐԻ ՑԱՆԿ

ՀԱՎԵԼՎԱԾ 5. ԱՂՅՈՒՍԱԿՆԵՐԻ ՑԱՆԿ

ՀԱՎԵԼՎԱԾ 6. ՀԱՊԱՎՈՒՄՆԵՐԻ ՑԱՆԿ

ԻՍ - ինտեգրալ սխեմա

ՄՕԿ - մետաղ-օքսիդ-կիսահաղորդիչ

ՍՄՎ - սիլիցիումը մեկուսիչի վրա

ԵՏ - եռաչափ տրանզիստոր

ՇԼ - շեմային լարում

ԼՂԴ - լարմամբ ղեկավարվող դիմադրության

ՓԱԼ - փական-ակունք լարումը

ԵՀ - ենթաշեմային հոսանքը

ԿՄՕԿ - կոմպլեմենտար մետաղ-օքսիդ-կիսահաղորդիչ

ՀԴՄ - հաստատուն դաշտով մասշտաբավորում

ԼԱ - լրիվ աղքատացած

ՄԱ - մասնակի աղքատացած

ԲՓ - եռաչափ բազմափական

ԱՓԵՏ - անկախ փականով եռաչափ տրանզիստոր

ԿՓԵՏ - կարճ փականային միացմամբ եռաչափ տրանզիստոր

ՏՓ - տրամաբանական փական

ՑԷՄ - ցածր-էներգասպառմամբ միացում

ԹՍՏԳ - թվային ստանդարտ տարրերի գրադարան

ԷՀԱ - էներգիայի և հապաղման արտադրյալը

ՍԼՋ - սնուցման լարման և ջերմաստիճանի

ՏՋ - տրամաբանական ջանքի մեթոդ

ՆՉԼ - նվազագույն չափերով լավարկման մեթոդ

ՏՕԼ - տրանզիստորների օպտիմալացված լավարկման

ԵԱՖ - ելքի աշխատանքների ֆլուկտուացիա

ՍՀ - ստատիկ հիշող սարքեր

ՍՀԲ - ստատիկ հիշող սարքերի բջիջ

ՍՀՄ - ստատիկ հիշող սարքերի մատրից

ԸԳԱԿ - ընթերցման գործողության աղմկակայունությունը

ԳԳԱԿ - գրելու գործողության աղմկակայունությունը

ՄԿՎ - Մոնտե Կառլո վերլուծություն

ՍԱՊ - ստատիկ աղմկակայունության պաշար

ԿԳ - կարգային գիծ

ԲԳ - բառի գիծ

ՀՀ - հիշողության հանգույց

ԳԳ - գրանցման գործողություն

ԸԳ - ընթերցման գործողություն

ՊՏ - պաշտպանիչ տրանզիստոր

ՑԱՊ - տրամաբանական «ցածր» մակարդակի աղմկակայունության պաշարը

ՀՏ - հողանցող տրամաբանությունը

ՈՒՄ - ուժային մեթոդ

ՀԿՇ - հետադարձ կապի շրջիչից

ՆՊԵ - նախագծման պատվիրովի երթուղի

ԳՆՀ - գործընթացի նախագծման հավաքածու

ՖՆ - ֆիզիկական նախագիծը

ՆԿՈՒ - նախագծման կանոնների ուղեցույց

ՊՔ - պարազիտային քաղվածքը

ՌՓՄ - ռեգիստրային փոխանցման մակարդակը

ՍՆԼ - սարքավորումների նկարագրում որևէ լեզվով

ՏՍ - տրամաբանական սինթեզ

ՓՄ - փականային մակարդակ

ԳԿՏՊ - գրադարանային կառուցվածքային տարրերի պահոց

ԳԿ - գործողությունների կազմակերպիչ

ՍՀՄԿ - ստատիկ հիշողության մատրիցի կոմպիլյատոր

ԴՏԿ - դոմինո տրամաբանության կոմպիլյատոր

ՍՏԿ - ստատիկ տրիգերների կոմպիլյատոր

ԳԲ - գրաֆիկական բլոկ

ՍՏ - ստատիկ տրիգեր