ՀՀ ԿՐԹՈՒԹՅԱՆ ԵՎ ԳԻՏՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ
ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ
ՀԱՄԱԼՍԱՐԱՆ

ԴԱՎԻԹ ՌՈԲԵՐՏԻ ԲԱԲԱՅԱՆ

ՊՐՈՑԵՍՈՐԱՅԻՆ ԻՆՏԵԳՐԱԼ ՍԽԵՄԱՆԵՐԻ ԷՆԵՐԳԱՍՊԱՌՄԱՆ ՆՎԱԶԵՑՄԱՆ ՄԻՋՈՑՆԵՐԻ ՄՇԱԿՈՒՄԸ ԵՎ ՀԵՏԱԶՈՏՈՒՄԸ

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> Ժամանակակից կոմպլեմենտար մետաղ-օքսիդ-կիսահաղորդիչ (ԿՄՕԿ) ինտեգրալ սխեմաներում (ԻՍ) տրանզիստորների հոսքուղու երկարության անընդմեջ նվազման արդյունքում այժմ այն հասել է ընդհուպ մինչև 7 *նմ*-ի: Դրան զուգընթաց՝ բարձրանում են ԻՍ-ում ինտեգրման աստիճանը և ստատիկ հզորության արժեքը (պայմանավորված ԿՄՕԿ տեխնոլոգիական առանձնահատկություններով պասսիվ վիճակում կորստի հոսանքներով): Այժմ ԻՍ-երի էներգասպառման նվազեցումը նախագծման հիմնական մարտահրավերն է: Դրանց բացարձակ էներգասպառումը հասել է ընդհուպ մինչև մի քանի հարյուր *Վտ*-ի, իսկ տեսակարար էներգասպառումը մոտեցել է միջուկային ռեակտորի համանման մեծություններին: Այդ մարտահրավերն առավել արդիական է պրոցեսորային ԻՍ-երի պարագայում, քանի որ դրանք պարունակում են ամենաշատ քանակով տրանզիստորները մեկ առանձնացված հանգույցում:

Գերմեծ պրոցեսորային ԻՍ-ում առավել մեծ կարևորություն ունի հաճախությունը, որը պարբերաբար աճում է՝ բավարարելու համար տեխնիկական առաջադրանքը և արդի ծրագրային մարտահրավերները: Հայտնի է, որ էներգասպառումը համեմատական է հաճախությանը: Պրոցեսորային համակարգերը հիմնականում կազմված են թվային սինքրոն հանգույցներից, որոնք կառավարվում են սինքրոն-ազդանշանի միջոցով, որն ունի ամենամեծ ճյուղավորման գործակիցը սխեմայում, կատարում է ամենաշատ փոխանջատումը, և հետևաբար՝ ամենամեծ դինամիկ հզորության բաժինը բաշխվում է հենց դրա վրա:

Հայտնի է, որ էներգասպառումը համեմատական է նաև սնուցման լարման քառակուսուն, որի արժեքը շարունակաբար փոփոխվելով՝ հասել է մինչև 0,65 *Վ*-ի, և դրա հետագա նվազեցումը սահմանափակվում է տեխնոլոգիական գործընթացով ու շեմային լարման արժեքով:

Այդ հիմնախնդրի լուծմանն են ուղղված տարբեր հեղինակային խմբերի կատարած հետազոտությունները, և ըստ դրանց՝ վերջին շրջանում մշակվել են տարբեր եղանակներ՝ խնդիրը լուծելու համար: Մասնավորապես՝ բազմասնուցում, բազմաշեմային, տակտային ազդանշանի շրջափակում և հզորության շրջափակում մեթոդների օգտագործումը փոքրացրել է էներգասպառումը ԻՍ-երում:

Այնուամենայնիվ, առկա մեթոդները չեն բավարարում գործող պահանջները, որն էլ անհրաժեշտություն է ստեղծում նոր հետազոտությունների կատարման համար, որոնք կապահովեն արագագործության և կիսահաղորդչային բյուրեղի վրա զբաղեցրած մակերեսի վրա ոչ էական մեծացումը:

Ատենախոսությունը նվիրված է պրոցեսորային ԻՍ-ի էներգասպառման նվազեցման հիմնահարցերի լուծմանը:

<u>Հետազոտության առարկան:</u> RISC ճարտարապետության պրոցեսորային ԻՍ-ի սնման և շեմային լարումների տատանումների ազդեցությունը էներգասպառման և ժամանակային պարամետրերի վրա, դրանցում ստատիկ և դինամիկ բաղադրիչների միաժամանակ նվազարկման եղանակները և դրանց ազդեցությունը կիսահաղորդչային բյուրեղում (ԿԲ-ում) ԻՍ-ի զբաղեցրած մակերեսի վրա:

<u>Աշխատանքի նպատակը:</u> Պրոցեսորային ԻՍ-երում էներգասպառման նվազեցմանը միտված մեթոդների, եղանակների և համապատասխան սխեմատեխնիկական լուծումների մշակումը և հետազոտումը, որոնք ուղղված կլինեն պրոցեսորային ԻՍ-ում միասնական դինամիկ և ստատիկ հզորությունների նվազարկմանը: Դրանք միաժամանակ պետք է ապահովեն նվազագույն ազդեցություն կիսահաղորդչային բյուրեղի վրա ԻՍ-ի զբաղեցրած մակերեսի և արագագործության արժեքների վրա:

<u>Հետազոտության մեթոդները:</u> Ատենախոսության կատարման ընթացքում օգտագործվել են ցածր էներգասպառմամբ նախագծման առկա մեթոդները, RISC պրոցեսորային ԻՍ-երի միկրոճարտարապետության դասական տարբերակները, սխեմաների նկարագրման լեզուները, թվային նախագծման գործընթացը:

<u>Գիտական նորույթը.</u>

- Առաջարկվել են պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկման սկզբունքներ, որոնց հիման վրա նախագծումն ապահովում է սնման աղբյուրից ծախսվող հոսանքների բավարար չափով փոքրացում:
- Առաջարկվել է RISC ճարտարապետությամբ պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկման եղանակ, որում ցածր հզորության նախագծման մեթոդների համադրության շնորհիվ առկա լուծումների համեմատ ապահովվում է էներգասպառման ժամանակակից պահանջները բավարարող չափով փոքրացում:

- Մշակվել է հզորության շրջափակման մեթոդի տարատեսակ, որի դեպքում ցածր էներգասպառման ռեժիմում վիճակի պահպանման և վերականգնման ֆունկցիայի՝ պրոցեսորային ինտեգրալ սխեմայի ներսից դեպի եզրագծեր տեղափոխման հաշվին ստացվում է էներգասպառման ստատիկ բաղադրիչի զգալի նվազեցում՝ կիսահաղորդչային բյուրեղի վրա ինտեգրալ սխեմայի զբաղեցրած մակերեսի թույլատրելի մեծացման պարագայում:
- Հզորության շրջափակման մեթոդում առաջարկված մոտեցման կիրառությունն ապահովելու նպատակով կառուցվել են ելքային ազդանշանի 0,5…1,5 *ԳՀց* հաճախականային միջակայքում աշխատող, վիճակի պահպանման և վերականգնման հնարավորությամբ հատուկ լարման մակարդակի փոխակերպիչներ, որոնք լրացուցիչ լիցքաթափում իրականացնող տրանզիստորային հանգույցի հաշվին ապահովում են զգալիորեն ցածր էներգասպառում, բարձր արագագործություն և պահպանում են համակարգի կայունությունը՝ բացառելով մետակայուն վիճակում հայտնվելու հնարավորությունը:
- Առաջարկվել է պրոցեսորային ինտեգրալ սխեմաների դինամիկ լարման և հաճախության մասշտաբավորման եղանակ, որի դեպքում ջերմաստիճանի կախվածությունը հաշվի առնելով՝ ստացվել է էներգասպառման էական նվազարկում, միաժամանակ մեծացնելով կիսահաղորդչային բյուրեղում ինտեգրալ սխեմայի զբաղեցրած մակերեսը:

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

- պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկման սկզբունքները,
- RISC ճարտարապետությամբ պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկման եղանակը,
- հզորության շրջափակման մեթոդի տարատեսակը,
- վիճակի պահպանման և վերականգնման հնարավորությամբ հատուկ լարման մակարդակի փոխակերպիչները,
- դինամիկ լարման և հաճախության մասշտաբավորման եղանակը:

Աշխատանքի գործնական արժեքը:

Ատենախոսությունում առաջարկված պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկմանն ուղղված մեթոդների, եղանակների և սխեմատեխնիկական հանգույցների հիման վրա մշակված PWR Compiler ծրագրային միջոցի արդյունավետության գնահատման նպատակով այն փորձարկվել է բազմաթիվ սխեմաներում, մասնավորապես՝ RISC ճարտարապետության պրոցեսորային ինտեգրալ սխեմայի վրա: Ժամանակային հապաղումների՝ գործնականում անփոփոխ մնալու պարագայում ընդհանուր էներգասպառումը կրճատվել է մոտ 54%-ով՝ կիսահաղորդչային բյուրեղում ինտեգրալ սխեմայի զբաղեցրած մակերեսի ընդամենը 3,8% աճի դեպքում, միաժամանակ ապահովելով նախագծի կատարման ժամկետի մինչև յոթ անգամ կրճատում:

Գիտական դրույթների հավաստիությունը: Առաջարկված եղանակներով և միջոցներով նախագծված ԻՍ-երի հիմնական բնութագրերը և տեխնիկական պարամետրերը համադրելի են արտադրված ԻՍ-երի համանման ցուցանիշների հետ:

Ներդրումը: PWR Compiler ծրագրային միջոցը ներդրված է <<Սինոփսիս Արմենիա>> ՓԲԸ-ում: Մշակված ծրագրային միջոցը պրոցեսորային ԻՍ-երի ցածր էներգասպառմամբ նախագծում իրականացնելիս կիրառվում է որպես թվային ավտոմատ նախագծման գործընթացի բաղկացուցիչ մաս: RISC ճարտարապետության տարբեր պրոցեսորային ԻՍ-եր և ՀՀԴ թվային միջուկներ սինթեզվել են առաջարկված ցածր էներգասպառման մեթոդներով՝ օգտագործելով PWR Compiler ծրագրային միջոցը: Ծրագրային միջոցը լայնորեն կիրառվում է այն հանգույցների նախագծման ժամանակ, որտեղ օգտագործվում են միասնական հզորության ձևաչափեր: Առաջարկված PWR Compiler-ը ինտեգրված է Սինոփսիս ընկերության ծրագրային փաթեթում:

Հրապարակումներ: Ատենախոսության հիմնական դրույթները տպագրված են 10 գիտական հոդվածներում:

Ատենախոսության կառուցվածքը: Ատենախոսությունը բաղկացած է՝ ներածությունից, 3 գլուխներից, եզրահանգումից, 110 անուն գրականության ցանկից և 4 հավելվածներից: 3 գլուխների ընդհանուր էջերի քանակը 137 է, հավելվածի հետ մեկտեղ կազմում է 165 էջ: Ատենախոսության մեջ ներառված են 76 նկարներ և 20

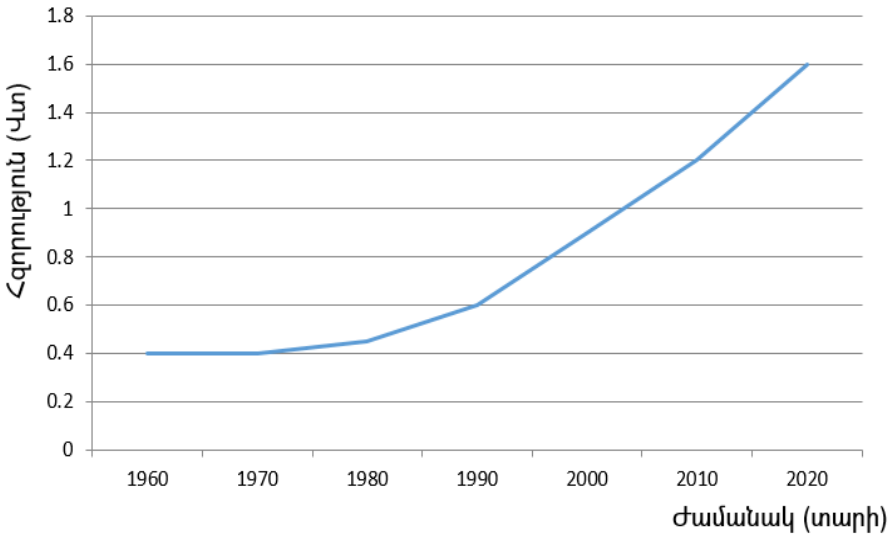
աղյուսակներ: Հավելվածները ներառում են՝ 1) PWR Compiler ծրագրային միջոցի ներդրման ակտ <<Սինոփսիս Արմենիա>> ՓԲԸ-ում, 2) RISC պրոցեսորային ԻՍ-ի կառուցվածքային նկարագիրը Verilog սխեմաների նկարագրման լեզվով, 3) PWR Compiler ծրագրային միջոցի նկարագրությունից հատվածներ, 4) օգտագործված նկարների, աղյուսակների, հապավումների ցանկերը և դրանց բացատրությունները:

ԳԼՈՒԽ 1. ՊՐՈՑԵՍՈՐԱՅԻՆ ԻՆՏԵԳՐԱԼ ՍԽԵՄԱՆԵՐԻ ԷՆԵՐԳԱՍՊԱՌՄԱՆ ՆՎԱԶԵՑՄԱՆ ԸՆԴՀԱՆՈՒՐ ՀԱՐՑԵՐԸ

1.1. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման անհրաժեշտությունը

Կոմպլեմենտար մետաղ-օքսիդ-կիսահաղորդիչ (ԿՄՕԿ) տեխնոլոգիական գործընթացները ներկայումս անընդմեջ ենթարկվում են փոփոխությունների՝ ապահովելով դրանցով նախագծված տրանզիստորի չափերի նվազեցում և մեկ բյուրեղում ինտեգրված տրանզիստորների քանակի աճ [1,2]: Տրանզիստորների չափերի անընդմեջ նվազումը և միաժամանակ մեկ բյուրեղում ինտեգրված տրանզիստորների քանակի աճը բնութագրվում է Գորդոն Մուրի օրենքով [3]: Ըստ այդ օրենքի՝ 18 ամիսը մեկ բյուրեղում ինտեգրված տարրերի քանակը կրկնապատկվում է: Միաժամանակ, տեղի է ունենում տրանզիստորների չափերի կրկնակի նվազում: Զարգացման այս միտումների պարագայում մեծանում է ԻՍ-երում էներգասպառման արդյունավետ օգտագործման անհրաժեշտությունը: Բազմաթիվ սարքավորումներում, որտեղ ներդրված են ԻՍ-եր, այժմ օգտագործվում են դյուրակիր մարտկոցներ՝ որպես դրանց սնման հիմնական աղբյուր [1,4]:

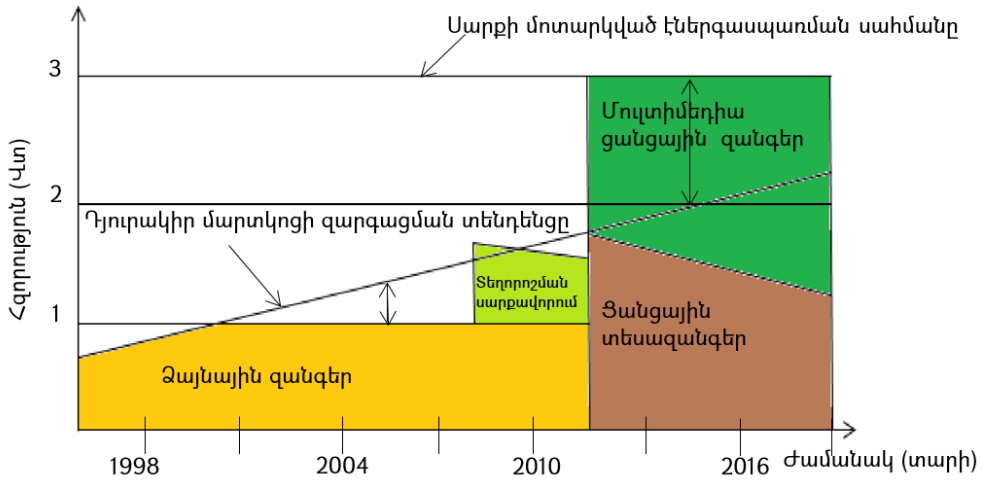
Սկսած 1990թ. –ից (նկ. 1.1)՝ դյուրակիր մարտկոցների հզորության աճի տեմպերն արագացել են, ինչը զիջում է ԻՍ–երի զարգացմամբ պայմանավորված էներգասպառման աճին [1,4]:



Նկ. 1.1. Դյուրակիր մարտկոցների հզորության աճի տեմպերը

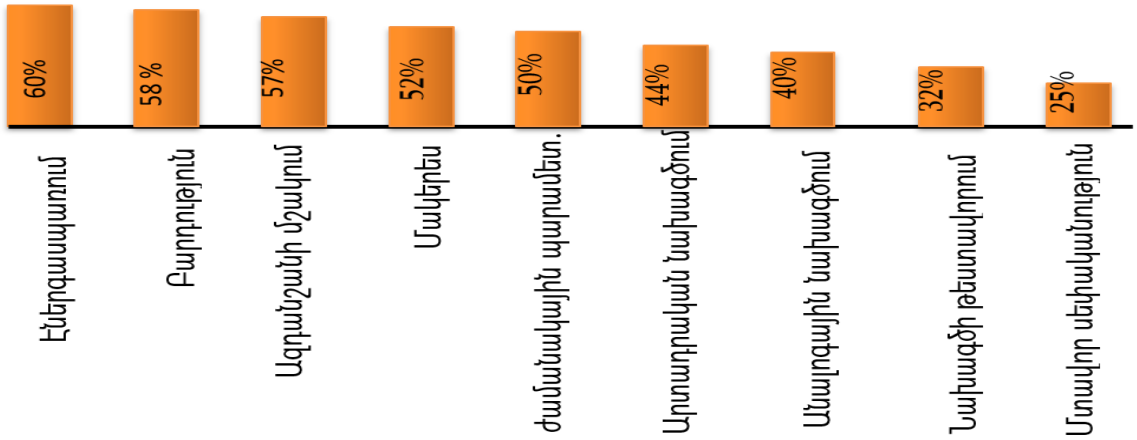
Էներգասպառման նվազարկման անհրաժեշտությունը դառնում է էլ ավելի արդիական այն սարքերում, որոնք աշխատում են միայն դյուրակիր մարտկոցներով, որտեղ կա անհրաժեշտություն՝ մեծացնելու օգտակար աշխատաժամանակը:

Սկսած 2010թ.-ից (նկ. 1.2)՝ դյուրակիր մարտկոցների զարգացման տեմպերը անտարակույս զիջում են արդի սարքերի զարգացման համանման պարամետրերին և հետևաբար՝ նաև էներգասպառմանը, որը հիմնականում պայմանավորված է այդ սարքերում նոր հնարավորությունների ավելացմամբ [1]:



Նկ. 1.2. Էներգասպառման և դյուրակիր մարտկոցների զարգացման միտումների հարաբերակցությունը

Ճարտարագետների շրջանում անցկացված հարցման արդյունքներով ստացվել են նմանատիպ ցուցանիշներ (նկ. 1.3) [1,5]. հարցվածների 60%-ը էներգասպառումը նշում են որպես նախագծման գերխնդիր, որը և հարցման արդյունքներով զբաղեցնում է առաջին հորիզոնականը:



Նկ. 1.3. Ճարտարագետների շրջանում կատարված հարցման արդյունքները

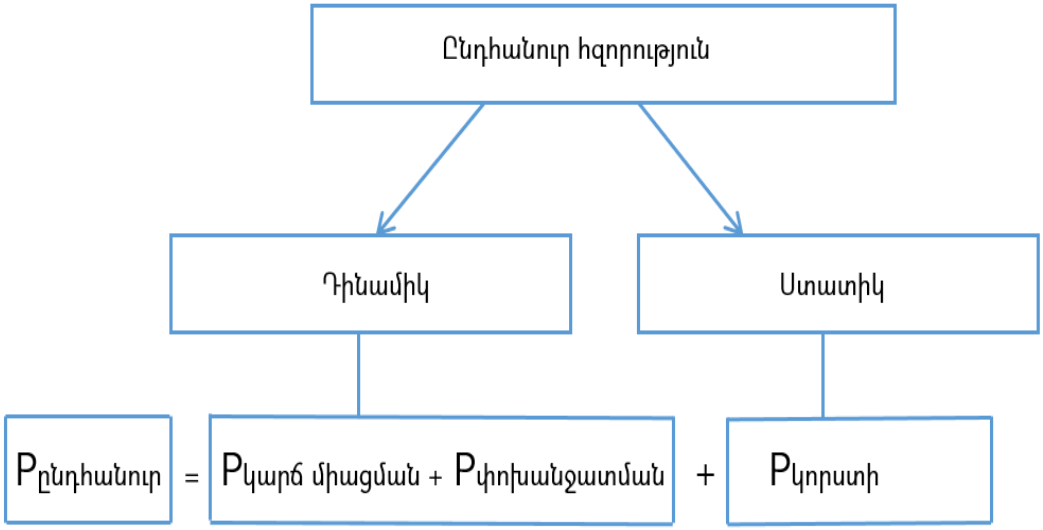
Նախագծման կարևորագույն հիմնահարցերից են նաև ժամանակային պարամետրերը, մակերեսը, ազդանշանի մշակումը, նախագծի թեստավորումը և այլն [2]:

1.1.1. ԿՄՕԿ ինտեգրալ սխեմաներում էներգասպառման հիմնական աղբյուրները

ԻՍ-երի էներգասպառման նվազարկումը հնարավորություն է ընձեռում ապահովել ավելի շատ ակտիվ աշխատաժամանակ դյուրակիր մարտկոցների համար, որոնց տեխնիկական բնութագրերը հասել են իրենց առավելագույն ցուցանիշներին: ԻՍ-երի զարգացմանը զուգահեռ՝ աճում է նաև ԻՍ-ի տակտային հաճախությունը, որը, ինչպես հայտնի է, համեմատական է հզորությանը և անջատվող ջերմաքանակին, և որն էլ էներգասպառման հիմնական աղբյուրներից մեկն է [5]: Էներգասպառման աճին զուգահեռ՝ աճում է նաև անջատվող ջերմաքանակը ԻՍ-ի ամբողջ մակերեսով, որն իր հերթին հանգեցնում է ԻՍ-ի տաքացմանը, ինչն էլ կարող է հանգեցնել ԻՍ-ի աշխատանքի կայունության խափանմանը [5,6]: Հզորության աճի հիմնական աղբյուրների շարքն է դասվում նաև ինտեգրման աստիճանի բարձրացումը՝ պայմանավորված միավոր մակերեսում ներդրված տարրերի թվաքանակի աճով [4]: Արդի ԻՍ–երի զարգացումը հանգեցնում է ֆունկցիոնալ նոր հնարավորությունների ավելացմանը, որն իր հերթին բարդացնում է ԻՍ–ի կառուցվածքը և հետևաբար՝ մեծացնում էներգասպառումը: Տեխնոլոգիական գործընթացների երկրաչափական չափերի նվազման հետ մեկտեղ՝ ի հայտ են գալիս նոր երևույթներ, որոնք մեծացնում են ստատիկ հզորության առանձին բաղադրիչները՝ ստեղծելով բավականին մեծ կշիռ ընդհանուր էներգասպառման մեջ [7]: Պրոցեսորային ԻՍ-երում առկա են թվարկված բոլոր բաղադրիչները՝ բարձր տակտային հաճախություն, գերբարձր ինտեգրման աստիճան, կորստի հզորության մեծ կշիռ [6]: Տվյալ բաղադրիչների առկայությունը պրոցեսորային ԻՍ-երում ստեղծում է դրանցում էներգասպառման նվազարկման նոր մեթոդների և մոտեցումների կիրառման անհրաժեշտություն:

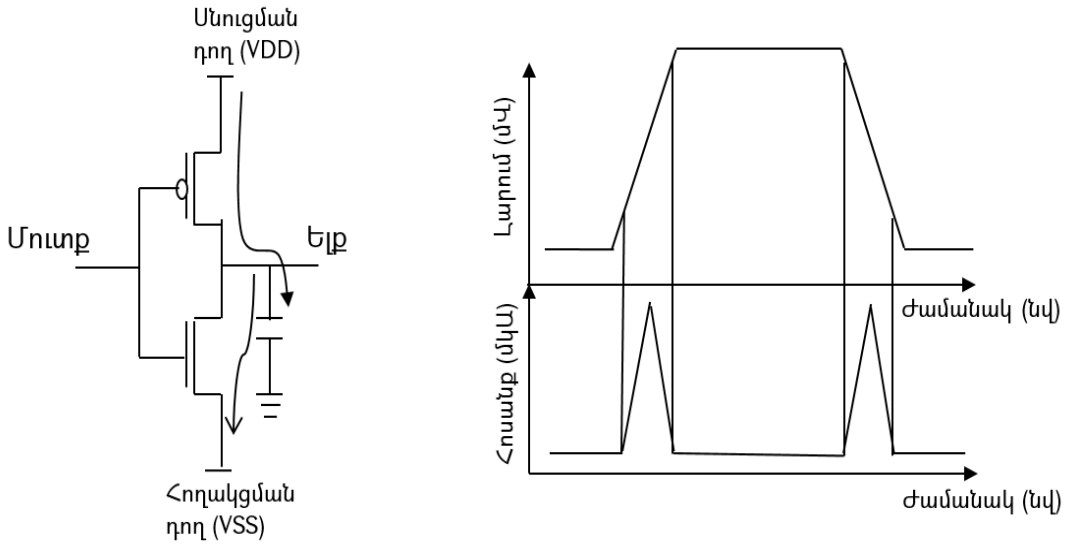
1.1.2. ԿՄՕԿ ինտեգրալ սխեմաներում դինամիկ, ստատիկ հզորությունները և դրանց բաղադրիչները

ԻՍ–երի ընդհանուր էներգասպառումը բաղկացած է երկու՝ դինամիկ և ստատիկ բաղադրիչներից (նկ. 1.4) [7,8]: Դինամիկ հզորությունը իր հերթին բաժանվում է երկու մասի՝ փոխանջատման և կարճ միացման հզորությունների: ԿՄՕԿ տեխնոլոգիական գործընթացում ստատիկ բաղադրիչը հիմնականում պայմանավորված է սխեմայի կորստի հզորությամբ [8]:



Նկ. 1.4. ԿՄՕԿ ԻՍ-երում ընդհանուր հզորության կառուցվածքը

Փոխանջատման հզորությունը սխեմայի փոխանջատման ընթացքում առաջացած էներգասպառումն է, երբ ազդանշանի մեկ մակարդակից անցում է կատարվում մեկ ուրիշ մակարդակի: Փոխանջատման հզորության օրինակը բերված է ԿՄՕԿ շրջիչի հիման վրա (լիցքավորման և լիցքաթափման ժամանակ) (նկ. 1.5ա) [8]: Փոխանջատման ժամանակ ազդանշանի անցումը ցածրից բարձր մակարդակի և հակառակը, միաժամանակ այդ ընթացում նաև հոսանքի բնութագիրը շրջիչի հիման վրա ներկայացված է նկ. 1.5բ-ում [8,9]: Ինչպես նկարագրված է բնութագրում, հոսանքն ունենում է կտրուկ աճ լիցքավորման (աճման) և լիցքաթափման (նվազման) ընթացքում, որն էլ հիմնականում պայմանավորված է ԻՍ-ում անցողիկ գործընթացներով: Դինամիկ հզորության ձևավորման գործընթացում հիմնական մասը պայմանավորված է փոխանջատման երևույթներով [9]:



ա) բ)

Նկ. 1.5. ա) ԿՄՕԿ շրջիչի հիման վրա փոխանջատման հզորության օրինակը, բ) շրջիչում փոխանջատման Ժամանակից հոսանքի և լարման կախվածությունները

Միավոր էներգիան փոխանջատման ընթացքում հավասար է [10]՝

$$Q_{փ} = C_{բ} * VDD^2, \quad (1.1)$$

որտեղ $C_{բ}$–ն սխեմայի ելքային ունակություն է, VDD-ն՝ սնուցման լարումը: Համապատասխան ձևափոխություններից հետո կարող է ստացվել հետևյալ բանաձևը [10] ՝

$$P_{դին} = Q_{փ} * F = C_{բ} * VDD^2 * P_{փոխ} * F_{տակտ}, \quad (1.2)$$

որտեղ F-ը փոխանջատման հաճախությունն է, $P_{փոխ}$-ը՝ դրա ելքային հավանականությունը, $F_{տակտ}$–ը՝ տակտային ազդանշանի հաճախությունը: Շարունակելով ձևափոխությունը և կիրառելով հետևյալ բանաձևը, կստացվի [10] ՝

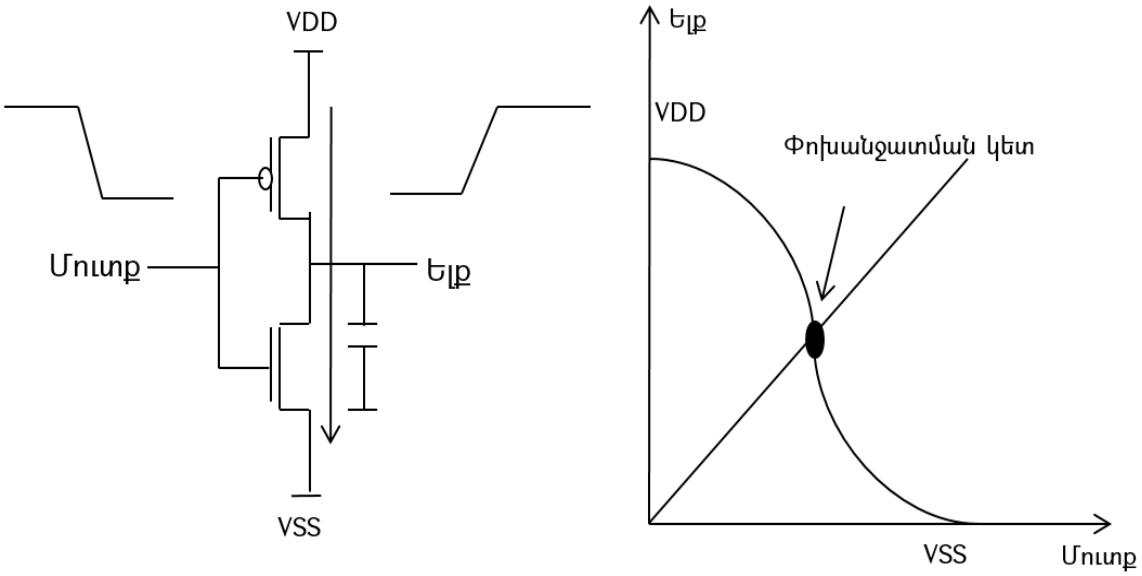
$$C_{արդյունավար} = P_{փոփ} * C_{բ}, \quad (1.3)$$

$$P_{դին} = C_{արդյունավար} * VDD^2 * F_{տակտ}: \quad (1.4)$$

Փոխանջատման հզորությունը կախվածության մեջ է գտնվում փոխանջատման ակտիվությունից և ելքային ունակային բեռից [10]:

ԿՄՕԿ ԻՍ–երում կարճ միացման հզորությունը այն էներգասպառումն է, երբ միաժամանակ կիսով չափ բաց են և P-ՄՕԿ և N-ՄՕԿ տրանզիստորները (կամ տրանզիստորային հանգույցները), որի ընթացքում տեղի է ունենում սնուցման և հողակցման դողերի միջև ուղիղ միացում (նկ. 1.6ա) [10]:

ԿՄՕԿ շրջիչի մուտք/ելք փոխանցման բնութագրում փոխանջատման կետը ցույց է տալիս առավելագույն կարճ միացման հոսանքը, որը սնուցման դողից հոսում է դեպի հողակցման դող, քանի որ այդ կետում N-ՄՕԿ և P-ՄՕԿ տրանզիստորներն ունեն ամենամեծ հավասարաչափ բացվածքը (նկ. 1.6բ) [8-10]:



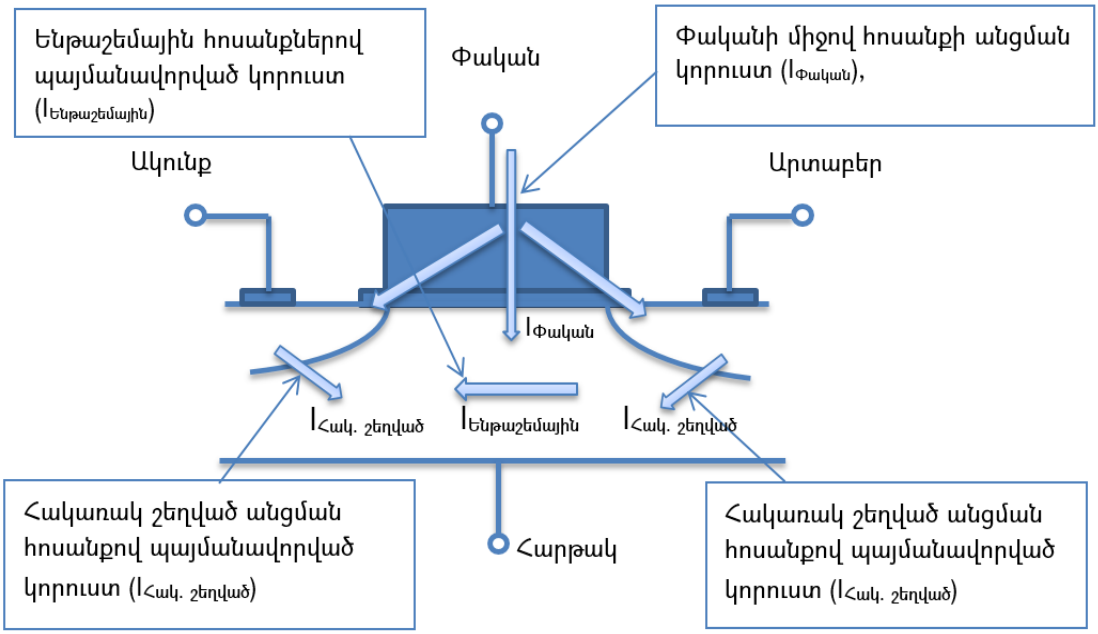
ա) բ)

Նկ. 1.6. ա) ԿՄՕԿ շրջիչի հիման վրա կարճ միացման հզորության օրինակը, բ) ԿՄՕԿ շրջիչի փոխանցման բնութագիրը (պատկերված է նաև փոխանջատման կետը)

ԿՄՕԿ ԻՍ–երում կորստի հզորությունը այն էներգասպառումն է, երբ սարքը միացված է սնուցմանը, բայց ոչ մի ազդանշան չի փոփոխվում, և ոչ մի ակտիվ գործողություն չի կատարվում. այն ձևավորվում է միայն ԿՄՕԿ սխեմայի կորստի հոսանքների հիման վրա, որոնք հիմնականում երեքն են (նկ. 1.7) [9].

- Ենթաշեմային հոսանքներով պայմանավորված կորուստը ($I_{\text{Ենթաշեմային}}$): Դա այն հոսանքն է, որը հոսում է արտաբերից դեպի ակունք՝ թույլ ինվերս շերտի արդյունքում [8-11]:
- Փականի միջով հոսանքի անցման կորուստը ($I_{\text{Փական}}$): Դա այն հոսանքն է, որը հոսում է փականից մեկուսչի միջով դեպի հարթակ, արտաբեր և ակունք՝ փականի օքսիդի թունելավորման և տաք լիցքակիրների ինժեկցիայի պատճառով [8-12]:
- Հակառակ շեղված անցման հոսանքով պայմանավորված կորուստը ($I_{\text{Հակ. շեղված}}$): Այն հիմնականում բաղկացած է էլեկտրական դաշտի ոչ հիմնական

լիցքակիրներով պայմանավորված կորստի հոսանքի և աղքատացված տիրույթում ոչ հիմնական լիցքակիրների գեներացիայով պայմանավորված հոսանքների գումարից [8-11]:



Նկ. 1.7. Տրանզիստորի լայնական կտրվածքի հիման վրա կորստի հոսանքների տարատեսակները

Ենթաշեմային հոսանքներով պայմանավորված կորուստը ($I_{ենթ}$) հաշվարկվում է [10]՝

$$I_{ենթաշեմային} = I_0 exp^{q\frac{V_{փական/ակունք}-V_{շեմային}}{nkT}} * (1 - exp^{q\frac{-v_{արտաբեր/ակունք}}{kT}}) \ , \qquad (1.5)$$

որտեղ I_0 է [8-10] ՝

$$I_0 \approx \beta \left(\frac{kT}{q}\right)^2 e^{1.8} \ , \qquad (1.6)$$

որտեղ β է [11] ՝

$$\beta = \mu C_{օքսիդ} \frac{W}{L} \ : \qquad (1.7)$$

Տեղադրելով I_0 և β –ն (1.5) –ում, կստացվի [10] ՝

$$I_{ենթաշեմային} = \mu C_{օքսիդ} \frac{W}{L} \left(\frac{kT}{q}\right)^2 e^{1.8} exp^{q\frac{V_{փական/ակունք}-V_{շեմային}}{nkT}} * (1 - exp^{q\frac{-v_{արտաբեր/ակունք}}{kT}}), \qquad (1.8)$$

որտեղ W –ն տրանզիստորի լայնությունն է, L-ը՝ տրանզիստորի երկարությունը, kt/q արժեքը՝ 25,9 *մՎ* և n-ը կախված է տեխնոլոգիական գործընթացից և տատանվում է 1-ից 3-ի միջակայքում: Փականի միջով հոսանքի անցման կորուստը ($I_{փական}$) հաշվարկվում է [12] ՝

$$I_{\text{փական}} = I_0\, exp\left\{-\frac{-8\pi\sqrt{2q}}{3h}\left(m_{\text{արդյունավետ}} * \Phi_{\text{բ}}\right)^{0.5} Kt_{\text{օքսիդ}}\right\}: \qquad (1.9)$$

Հակառակ շեղված անցման հոսանքով պայմանավորված կորուստը ($I_{\text{հակ. շեղված}}$) հաշվարկվում է [10-12] `

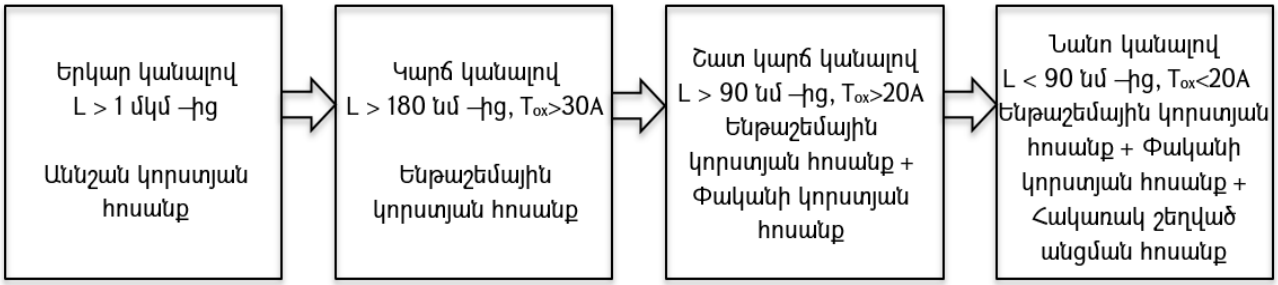
$$I_{\text{հակ.շեղված}} = A * J_u\left(e^{\frac{qVapp}{kT}} - 1\right): \qquad (1.10)$$

Այսպիսով, ընդհանուր կորստի հոսանքը կլինի [10-13] `

$$I_{\text{ընդհանուր}} = I_{\text{ենթաշեմային}} + I_{\text{փական}} + I_{\text{հակ.շեղված}}: \qquad (1.11)$$

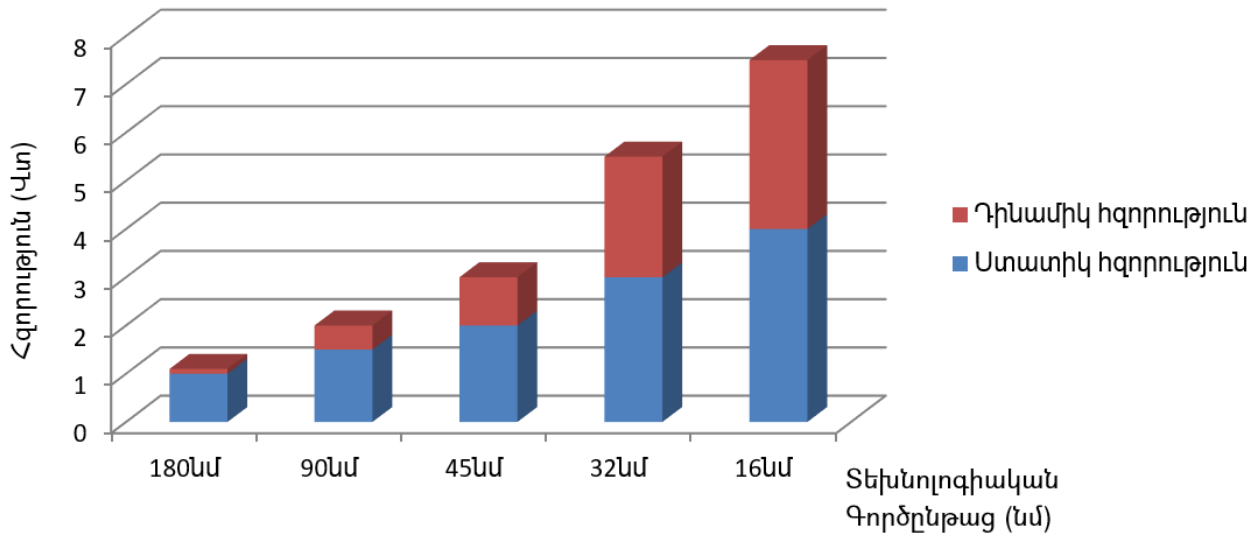
Տեխնոլոգիական գործընթացով նախագծված տրանզիստորների չափերի նվազմանը զուգահեռ՝ փոխվել է նաև կորստի հոսանքներով պայմանավորված կորստի հզորության դերը ընդհանուր էներգասպառման մեջ [13]:

Կորստի հոսանքները 1 *մկմ*-ից բարձր տեխնոլոգիական գործընթացում երկար հոսքուղու դեպքում հիմնականում անտեսվում են (նկ. 1.8)՝ պայմանավորված դրանց աննշան լինելով:



Նկ. 1.8. Տեխնոլոգիական գործընթացի փոփոխություններից կախված կորստի հոսանքների տարատեսակները

180 *նմ*-ից բարձր տեխնոլոգիական գործընթացում, երբ $T_{\text{օքսիդ}}$ մեծ է 30A–ից, կարճ հոսքուղու դեպքում հաշվի է առնվում ենթաշեմային կորստի հոսանքի ազդեցությունը [13]: 90 *նմ*-ից բարձր տեխնոլոգիական գործընթացում, երբ $T_{\text{օքսիդ}}$ մեծ է 20A–ից, շատ կարճ կանալի դեպքում ենթաշեմային կորստի հոսանքին գումարվում է նաև փականի կորստի հոսանքի ազդեցությունը: 90 *նմ* –ից ցածր տեխնոլոգիական գործընթացում, երբ $T_{\text{օքսիդ}}$ փոքր է 20A –ից, նանո կանալի դեպքում ենթաշեմային կորստի հոսանքին, փականի կորստի հոսանքին գումարվում է նաև հակառակ շեղված անցման հոսանքի ազդեցությունը: Այսպիսով, տեխնոլոգիական գործընթացից կախված՝ դինամիկ և ստատիկ բաղադրիչների համամասնությունը փոփոխվում է ընդհանուր հզորության ձևավորման մեջ՝ ի նպաստ ստատիկ բաղադրիչի (նկ. 1.9) [14]:

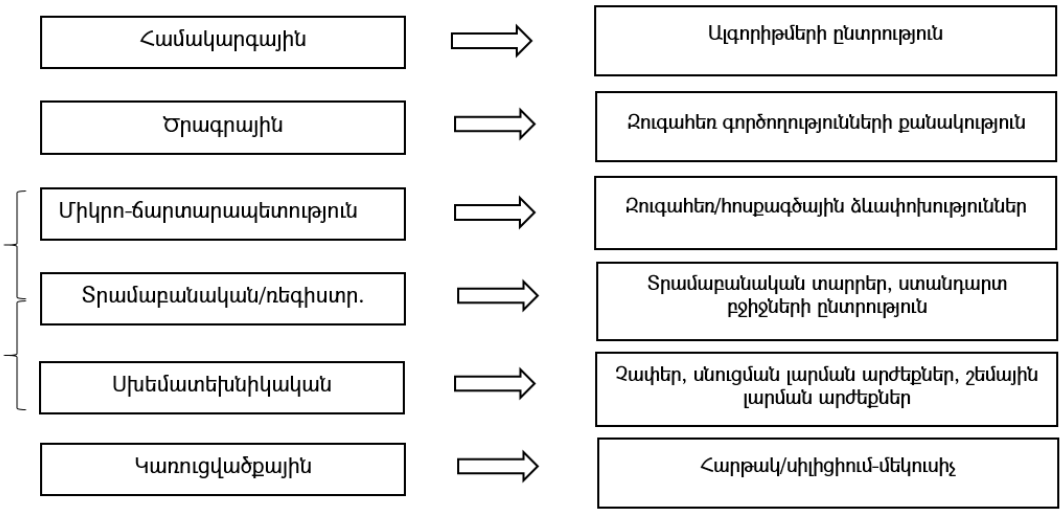


Նկ. 1.9. Տեխնոլոգիական գործընթացից կախված ստատիկ և դինամիկ հզորությունների արժեքները

1.2. ԿՄՕԿ ինտեգրալ սխեմաներում էներգասպառման նվազեցման արդի լուծումները և հիմնահարցերը

Գոյություն ունեն ԻՍ–ում էներգասպառման նվազարկման բազմաթիվ մեթոդներ, որոնք գործում են ԻՍ–ի տարբեր աբստրակցիայի մակարդակներում, ամենացածր մակարդակից՝ տոպոլոգիական, մինչև ամենաբարձր մակարդակը՝ ծրագրային: Ցանկացած մակարդակում նվազարկումը ներկայումս ունի իր դժվարությունները, քանի որ այն հիմնական պարամետրերը, որոնց վրա հիմնվելով կատարվում է էներգասպառման նվազարկումը, անընդմեջ ենթարկվում են փոփոխության, օրինակ՝ տրանզիստորի չափերը, սնման և շեմային լարումները, տրամաբանական մակարդակի նախագծման ձևերը, ֆիզիկական նախագծումը և միկրո-ճարտարապետությունը [15]: Տեսականորեն հնարավոր է հաշվի առնել բոլոր պարամետրերի ազդեցությունը մեկ մակարդակում նվազարկում իրականացնելիս, բայց գործնականում խնդրի բարդության պատճառով դա դառնում է անիրագործելի, որը լուծելու նպատակով էներգասպառման նվազարկման գործընթացը բաշխվում է ԻՍ-երի նախագծման տարբեր աբստրակցիայի մակարդակներում (նկ. 1.10) [15]: Ընտրված պարամետրերը, որոնք ազդեցություն են ունենում նվազարկման վրա, հնարավոր է հաշվի առնել ընտրված մեկ մակարդակում, օրինակ՝ ընտրությունը տարբեր հրահանգային հավաքածուների միջև տիպիկ միկրո-ճարտարապետության օրինակ է, իսկ տարբեր

շեմային լարումներով աշխատող սարքերի ընտրությունը լավագույնս նկարագրվում է սխեմատեխնիկական մակարդակում կատարված նվազարկմամբ [14]:



Նկ. 1.10. Նախագծման տարբեր մակարդակները և դրանցում էներգասպառման հնարավոր նվազեցումը

Առկա է տեսակետ, որ տարբեր մակարդակներում կատարված նվազարկումները անկախ են միմյանցից, որը սխալ է, քանի որ, օրինակ՝ տարբեր ենթաշեմային լարումներով սարքերի ընտրություն կատարելիս սխեմատեխնիկական մակարդակում այն ուղղակի ազդեցություն է ունենում տրամաբանական կամ միկրո-ճարտարապետության մակարդակներում: Նմանապես փոփոխություններ կատարելով միկրո-ճարտարապետության մակարդակում, օրինակ՝ հոսքագծային ձևափոխությունները կարող են մեծացնել օպտիմալացման ծավալը սխեմատեխնիկական մակարդակում: Հետևաբար՝ օպտիմալացումը պետք է ներառի բոլոր մակարդակները և ապահովի կապը դրանց միջև: Սովորաբար նախագծի օպտիմալացումը կատարվում է <<հանդիպում կենտրոնում>> սկզբունքով, երբ տեխնիկական առաջադրանքները և պահանջները տարածվում են բարձրից դեպի ցածր մակարդակներ, և սահմանափակումները տարածվում են ցածրից դեպի բարձր մակարդակներ [16]:

Ինչպես հայտնի է [17], սնման լարումն ունի քառակուսային կախվածություն հզորությունից [17].

$$P_{ընդհանուր} = I_{միջին} * U_{սնման} \quad , \tag{1.12}$$

$$P_{ընդհանուր} = \frac{U^2_{սնման}}{R} \quad : \tag{1.13}$$

(1.13)-ը նշանակում է, որ սնուցման լարման փոփոխությամբ կարելի է ազդել ընդհանուր հզորության ձևավորման վրա: Միևնույն ժամանակ, հայտնի է (1.14), որ սնուցման լարումը պայմանավորված է ԻՍ–ի ժամանակային հապաղումներով, և դրա փոքրացումը կարող է հանգեցնել ժամանակային պարամետրերի վատթարացմանը [18].

$$t_{բարձրից/ցածր} = 0.7 * R_n * C_բ = \frac{VDD * C_բ}{\beta_n (VDD - V_{շեմային})^2} \quad , \tag{1.14}$$

$$t_{ցածրից/բարձր} = 0.7 * R_p * C_բ = \frac{VDD * C_բ}{\beta_p (VDD - V_{շեմային})^2} \quad , \tag{1.15}$$

ինչը նշանակում է, որ հարկավոր է սնման լարման արժեքի փոփոխությունից առաջ հաշվի առնել նաև սխեմայի արագագործությունը և նրա հնարավոր շեղումները: Անհրաժեշտ է գտնել ընդունելի տարբերակ թե՛ հզորության և թե՛ արագագործության տեսանկյունից: [18]:

(1.14) և (1.15)-ից հետևում է, որ ԻՍ–երում հապաղումը հակադարձ կախվածության մեջ է գտնվում շեմային լարումից, ինչը և հնարավորություն է տալիս, ազդելով շեմային լարման արժեքի վրա, փորձել կարգավորել ԻՍ-ի հապաղումները, որոնք ենթարկվում են փոփոխության՝ կախված սնման լարման տատանումներից [17-18]:

Համաձայն վերլուծության՝ հիմնական նվազեցման առարկան կարող է լինել սնման լարումը, քանի որ ունի քառակուսային կախվածություն հզորությունից. մյուս պարամետրերն ավելի փոքր ազդեցություն ունեն: Բացի հիմնական պարամետրերից, որոնք բերված են բանաձևերում, կարևոր է նաև ուսումնասիրել ԻՍ-երի մյուս կարևոր բաղադրիչների վրա ազդեցությունը, ինչպիսիք են ԻՍ-ի մակերեսը և հուսալիությունը, օրինակ՝ ԻՍ–ի ոչ կրիտիկական ուղիներում տրանզիստորների չափերի փոքրացումը կարող է հանգեցնել մակերեսի, հզորության փոքրացմանը, միաժամանակ՝ հապաղումների մեծացմանը [19]: Հապաղումների շատ մեծացումը մոտեցնում է ԻՍ-ի առավելագույն թույլատրելի հապաղման չափին, որը և ԻՍ–ն դարձնում է ավելի զգայուն տեխնոլոգիական գործընթացի շեղումների նկատմամբ:

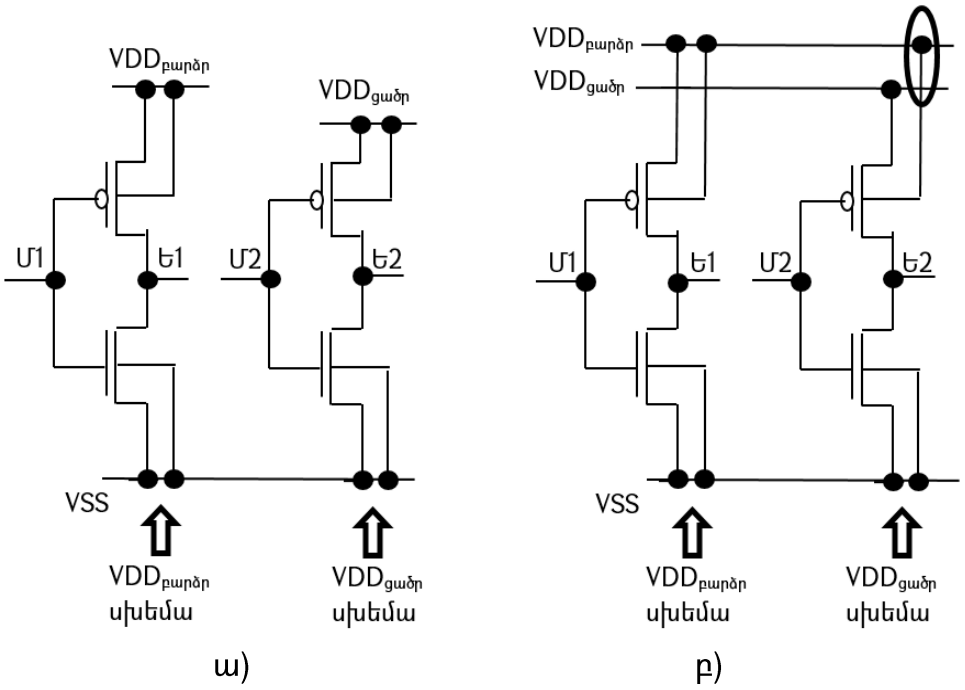
Ամփոփելով ներկայացված արդյունքները՝ կարելի է ասել, որ սնման և շեմային լարումներն ունեն առաջնային ազդեցություն ԻՍ–ի էներգասպառման ձևավորման գործընթացում [20]: Շեմային և սնման լարման հիման վրա իրականացված օպտիմալացումը համարվում է սխեմատեխնիկական/տրամաբանական մակարդակում

կատարված գործողություն: Հաշվի առնելով այն վարկածը, որ կարելի է փոփոխել միայն ոչ սահմանային ուղիների սնման լարումը, ստացվում է [21], որ պետք է նախագիծ իրականացնելիս օգտագործել երկու կամ ավելի սնման լարումներ և տեղաբաշխել դրանք ԻՍ-ում, համապատասխանաբար, ըստ ընտրված ուղիների: Այս մեթոդը կոչվում է <<բազմասնուցում նախագծում>> [21,22]: Երկրորդ մոտարկումն իրականացվել է շեմային լարման հնարավոր փոփոխության դեպքում, որը պետք է կատարել՝ վերլուծելով ընտրված հանգույցների հզորության/արագագործության/մակերեսի/հուսալիության տվյալները: Այնուհետև անհրաժեշտ է կատարել նախապես ընտրված ԻՍ-ի հանգույցներում տրանզիստորների փոխատեղում: Այս մեթոդը կոչվում է <<բազմակի շեմային լարումներով նախագծում>> [23]:

1.2.1. Բազմասնուցմամբ նախագծման մեթոդի կիրառմամբ պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցումը

Ինչպես արդեն քննարկվել է, բազմակի լարումների օգտագործումը ԻՍ-ում կարող է արդյունավետ լինել էներգասպառման նվազարկման տեսանկյունից: Անհրաժեշտ երկու լարման մակարդակներն ապահովելու համար պետք է կատարել փոփոխություններ նախագծման տարբեր մակարդակներում՝ սկսած հատուկ սխեմատեխնիկական լուծումներից, վերջացրած միկրո-ճարտարապետության փոփոխությամբ [24]: Սխեմատեխնիկական մակարդակում ներդրման համար հնարավոր է կիրառել երկու տարբերակներ՝ դասական և միացյալ N-գրպանիկով: Դասական տարբերակում օգտագործվում են տարբեր լարումներով աշխատող տարրեր, որոնց փականներին տրվում են տարբեր լարման մակարդակներ (նկ. 1.11ա), և դրանք տոպոլոգիական տեսանկյունից գտնվում են առանձնացված N-գրպանիկում: Այս տարբերակը չի պահանջում ստանդարտ բջիջների նոր նախագծում, բայց և պահանջում է ավելի մեծ մակերես ներդրման ժամանակ՝ պայմանավորված տարբեր լարումներով N-գրպանիկների միջև անհրաժեշտ տարածություն պահպանելու տեխնոլոգիական պահանջով [25]: Միացյալ N-գրպանիկով տարբերակում բոլոր տարրերը սնուցվում են միաժամանակ երկու բարձր և ցածր սնման լարման մակարդակներով (նկ. 1.11բ): Այս տարբերակի առանձնահատկությունն այն է, որ ցածր

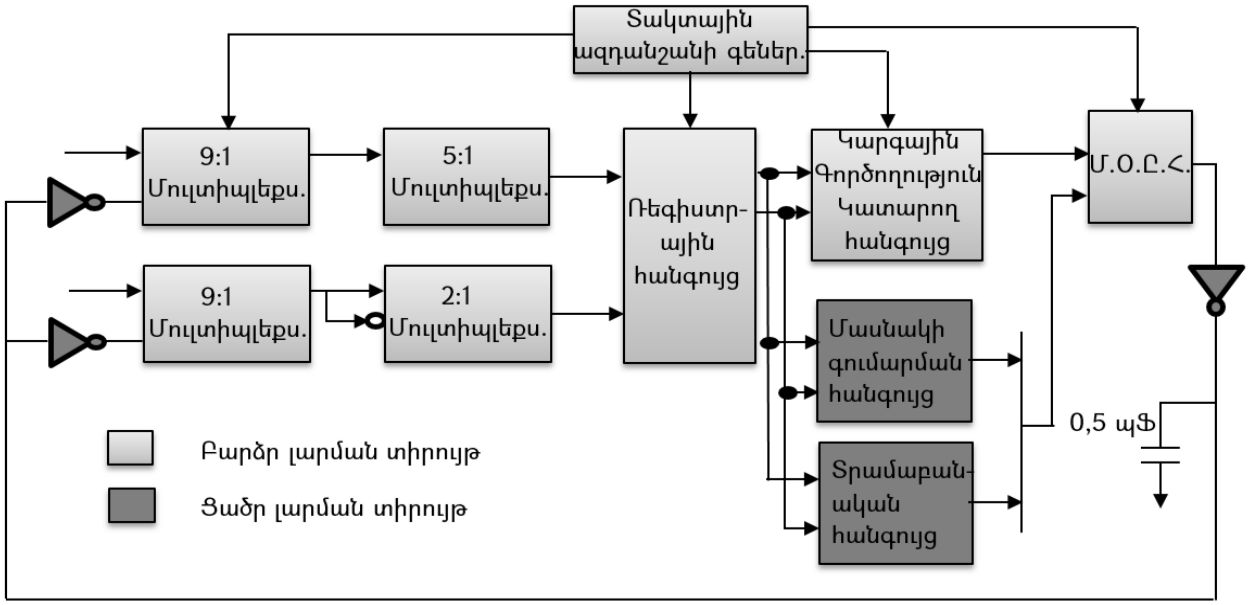
սնման լարմանը միացված տարրերի հարթակը միացվում է բարձր պոտենցիալին՝ դրանով իսկ ընձեռելով հնարավորություն կիսելու ընդհանուր N-գրպանիկը: Այս տարբերակը պահանջում է ստանդարտ բջիջների նոր նախագիծ, իսկ մակերեսի տեսանկյունից կարող է պահանջվել լրացուցիչ ծրագծում սնման լարման դողերի համար [26,27]:



Նկ. 1.11. ա) ԿՄՕԿ շրջիչի հիման վրա դասական տարբերակի իրականացումը, բ) ԿՄՕԿ շրջիչի հիման վրա միացյալ N-գրպանիկով տարբերակի իրականացումը

Դիտարկված մեթոդները առավելապես վերաբերում են ֆիզիկական նախագծմանը: Միացյալ N-գրպանիկով տարբերակը կիրառված է 64-կարգանի միկրոպրոցեսորի թվաբանական տրամաբանական գործողությունների հանգույցի (ԹՏԳՀ) վրա, որը հնարավորություն է ընձեռում պատկերացում կազմելու բազմակի լարումների կիրառման արդյունավետության մասին համակարգային բլոկ-սխեմայի մակարդակում: Միկրոպրոցեսորի ԹՏԳՀ-ի սխեման (նկ. 1.12) կազմված է հետևյալ հանգույցներից՝ մուտքային օպերանդներ ընտրող, կարգային գործողություն կատարող, մասնակի գումարման և ռեգիստրային հանգույցներից: Էներգասպառումը հիմնականում ձևավորվում է ելքային մեծ ունակային բեռով, որն էլ պայմանավորված է ելքի տանող ուժի բարձր աստիճանով [27]: Այսպիսով, իջեցնելով ելքային ուղու վրա գտնվող սխեմատեխնիկական հանգույցների սնման լարումները, կարելի է ապահովել

էներգասպառման զգալի նվազարկում: Ինչպես հայտնի է, կարգային գործողություն կատարող հանգույցը միկրոպրոցեսորային համակարգի կարևորագույն մասն է, որը տվյալ պարագայում միացվել է բարձր սնման լարման տիրույթին՝ ապահովելու համար բարձր արագագործություն և արտադրողականություն: Միաժամանակ, մասնակի գումարման և տրամաբանական հանգույցները միացվել են ցածր լարման տիրույթին, հաշվի առնելով այն փաստը, որ դրանք միացված են ելքային մեծ բեռին: Լարման մակարդակի փոխարկումները ցածրից բարձր և հակառակը կատարվում են մուտքային օպերանդներ ընտրող հանգույցի ելքում և 9:1 մուլտիպլեքսորի մուտքում տեղակայված լարման մակարդակի փոխակերպիչներով:



Նկ. 1.12. 64–կարգանի միկրոպրոցեսորի ԹՏԳՀ բլոկ-սխեման, այդ թվում՝ առանձնացված բարձր և ցածր լարման տիրույթները

Ստորև բերված են 64–կարգանի միկրոպրոցեսորի ԹՏԳՀ հանգույցի փորձարարական արդյունքները (աղ. 1.1): Մուտքային տվյալներն են՝ տակտային ազդանշանի հաճախությունը՝ 1,6 *ԳՀg*, ցածր սնման լարման արժեքը՝ 1,2*Վ* է, իսկ բարձրինը՝ 1,8 *Վ*:

Աղյուսակ 1.1

Բազմասնուցում նախագծման միացյալ N-գրպանիկով տարբերակը

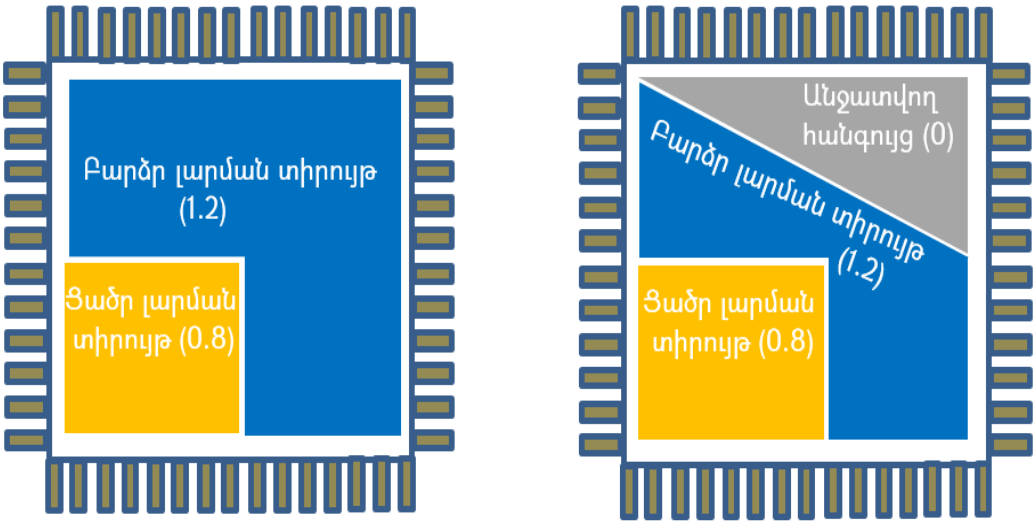
Պարամետրերը	Բազմասնուցում նախագծման միացյալ N-գրպանիկով տարբերակ
Հաճախություն *(ՄՀg)*	1600

Սնման լարում *(Վ)*	1,2…1,8
Ընդհանուր հզորություն *(մլՎտ)*	83
Ժամանակային հապաղումներ *(նվ)*	15,7

Արդյունքում, ի համեմատ դասական նախագծի, ստացվել է 20,9%-ի էներգասպառման նվազեցում, միևնույն ժամանակ, հապաղումն աճել է 8,3%-ով՝ ստեղծելով բավականին մեծ անհրաժեշտություն՝ նվազարկելու ստացված ժամանակային արդյունքները, ինչը հիմնականում պայմանավորված է ավելացված լարման մակարդակի փոխակերպիչներով [27]:

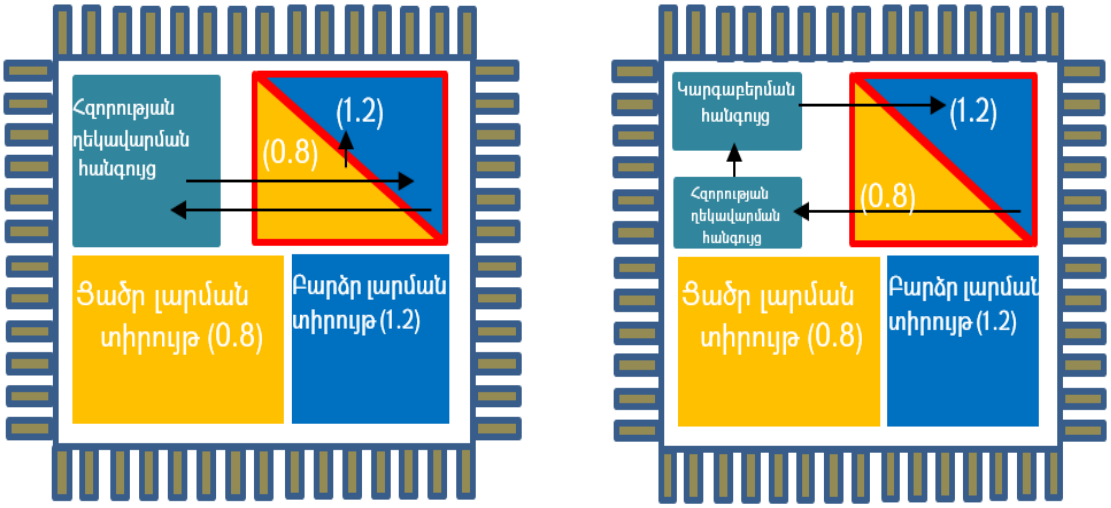
Ինչպես երևում է վերը շարադրվածից, բազմասնուցում նախագծման ժամանակ օգտագործվում են տարբեր լարումներով աշխատող տիրույթներ, որոնք անվանվում են լարման կամ հզորության կղզյակներ: Համակարգային տեսանկյունից միկրոպրոցեսորը և նրա մեջ գտնվող հանգույցներն ունեն ժամանակային տարբեր պահանջներ, որն էլ հնարավորություն է ընձեռում ունենալ լարման լայն տիրույթ բոլոր այդ հանգույցների համար [28]: Օրինակ՝ թվային միջուկը, տեխնիկական առաջադրանքից ելնելով, պետք է աշխատի տեխնոլոգիական թույլատրելի առավելագույն արագագործությամբ և հետևաբար՝ սնուցվի առավելագույն լարմամբ, բայց, միևնույն ժամանակ, օպերացիոն հիշողության հանգույցն ունի ավելի մեղմ պահանջներ արագագործության նկատմամբ, օրինակ, երբ անհրաժեշտ է միայն կատարել հիշելու գործողություն, կարող է սնուցվել ավելի ցածր լարումով, իսկ երբ կատարվում է գրել-կարդալու գործողությունը, պետք է սնուցվի բարձր լարումով: Բազմասնուցում նախագծումը կատարվում է մի քանի եղանակներով [29].

- ստատիկ լարման մասշտաբավորում (նկ. 1.13ա),
- բազմամակարդակ լարման մասշտաբավորում (նկ. 1.13բ),
- դինամիկ լարման և հաճախության մասշտաբավորում (նկ. 1.13գ),



➢ ադապտացվող լարման մասշտաբավորում (նկ. 1.13դ):

ա) բ)



գ) դ)

Նկ. 1.13. Բազմասնուցում նախագծման եղանակները՝ ա) ստատիկ լարման մասշտաբավորումը, բ) բազմամակարդակ լարման մասշտաբավորումը, գ) դինամիկ լարման և հաճախության մասշտաբավորումը, դ) ադապտացվող լարման մասշտաբավորումը

Ստատիկ լարման մասշտաբավորումը դասական տարբերակն է, երբ առանձնացված կղզյակները սնուցվում են տարբեր լարումներով, և այդ լարումների արժեքները մնում են անփոփոխ աշխատանքի ընթացքում: Բազմամակարդակ լարման մասշտաբավորումը այն տարբերակն է, երբ առանձնացված լարման կղզյակը կարող է

ունենալ մեկ և ավելի լարման մակարդակներ և փոխանջատել դրանց միջև՝ կախված աշխատանքային ռեժիմից [30]: Դինամիկ լարման և հաճախության մասշտաբավորումը հնարավորություն է ընձեռում բազմաթիվ լարման մակարդակների միջև կատարել դինամիկ փոփոխություններ՝ հիմք ընդունելով օպերացիոն ծանրաբեռնվածությունը: Ադապտացվող լարման մասշտաբավորումը նախորդի ընդլայնված տարբերակն է, որը պարունակում է հզորության կարգավորման հանգույց, որն էլ կարգավորում է լարման և հաճախության արժեքները [31]: Այնուամենայնիվ, գոյություն ունեն բազմասնուցում նախագծման բազմաթիվ մարտահրավերներ, որոնք առաջանում են նախագծման գործընթացում անգամ ամենապարզ տարբերակի կիրառության դեպքում.

- Ազդանշանի փոխանցումը մեկ լարման կղզյակից մեկ այլ լարման կղզյակ իրականացնելու համար անհրաժեշտ են լարման մակարդակի փոխակերպիչներ, որոնք բարդացնում են նախագծման գործընթացը [32]:
- Ժամանակային վերլուծության իրականացումը բավականին բարդանում է այն դեպքում, երբ օգտագործվում են լարման տարբեր մակարդակներ:
- Մի քանի լարման կղզյակների ներդրումն ավելի մանրակրկիտ հատակագծում է պահանջում, առաջանում է անհրաժեշտություն կատարելու հզորության պլանավորում, սնուցման բաշխման ցանցը դառնում է ավելի ծավալուն և բարդ:
- Համակարգային մակարդակում ավելացված սնուցումներն ապահովելու համար նույնպես առաջանում են դժվարություններ՝ լարման աղբյուրների և կարգաբերման համակարգերի ներդրման պատճառով:
- Անջատման վիճակին (անջատված սնուցում) բերելու և այնուհետև վերականգնման (միացված սնուցման) վիճակին հասնելու համար անհրաժեշտ կլինի ժամանակ և սխեմաների աշխատանքի համաձայնեցում՝ խուսափելու համար համակարգային սխալներից:

Ժամանակային խնդիրները բազմասնուցում նախագծման դեպքում ունեն բավականին մեծ նշանակություն, քանի որ կարող են հանգեցնել ԻՍ–ի աշխատանքում սխալների: Պրոցեսորային ԻՍ-երում այս հարցն էլ ավելի արդիական է՝ հաշվի առնելով այն փաստը, որ համակարգի մեծ մասը սինքրոն է և կառավարվում է տակտային ազդանշանով, և ժամանակային բոլոր վերլուծություններն ու հաշվարկները

կատարվում են՝ հիմք ընդունելով տակտային ազդանշանը [33]: Այն փաստը, որ տակտային ազդանշանը պետք է բաշխվի ամբողջ սխեմայով և անցնի տարբեր լարման կղզյակներով, էլ ավելի է բարդացնում տակտային ազդանշանի ծառի կառուցումը [34]:

Հզորության պլանավորումը և սնման լարման ցանցի մշակումը ԻՍ-ում նախագծման կարևորագույն մասերից են, և այն փաստը, որ օգտագործվում են տարբեր լարման կղզյակներ, էլ ավելի է մեծացնում դրա նշանակությունը: Ցանկացած լարման կղզյակ պետք է ունենա իր սեփական անկախ սնուցման ցանցը՝ ուղիղ միացմամբ դեպի մուտք/ելք հանգույց: Գնդային կառուցվածքով ԻՍ-երի դեպքում այս հարցը ավելի պարզ է, քանի որ առանձնացված մուտք/ելք սնուցման հանգույցը գտնվում է հենց լարման կղզյակում [30-34]: Ավանդական կառուցվածքի պարագայում, երբ սնուցման լարումը պետք է ստացվի շրջագծում գտնվող մուտք/ելք հանգույցներից, նախագծում կարող է սահմանափակվել լարման կղզյակների քանակը՝ ապահովելու համար անհրաժեշտ ծրագծման անցկացումը [35]:

Համակարգային մակարդակում դժվարություններ առաջանում են հիմնականում կարգաբերման ժամանակ, երբ անհրաժեշտ է լինում համապատասխանեցնել աշխատանքը պրոցեսորային ԻՍ-ի տարբեր ենթաբլոկները սինքրոնացնելիս: Օրինակ՝ բոլոր հանգույցները մինչև զրոյացման գործողությունը պետք է լինեն միացված սնման լարմանը և գտնվեն ակտիվ ռեժիմում: Այդպիսի դեպքերում կամ ավելի բարդ սինքրոնացումներ ապահովելու համար կարող են օգտագործվել լրացուցիչ կառավարման հանգույցներ, օրինակ՝ <<ձեռքսեղմում>> մեթոդի տարբեր կիրառությունները [36]: Նմանատիպ բարդ ինտեգրումներ ապահովելու համար շատ հաճախ օգտագործվում են հզորության կառավարման հանգույցներ:

Ստորև բերված են դինամիկ լարման և հաճախության մասշտաբավորմամբ (ԴԼՀՄ) նախագծված 128-կարգանի միկրոպրոցեսորի փորձարարական արդյունքները (աղ. 1.2): Տակտային ազդանշանը աշխատանքի ընթացքում փոփոխվում է 5 *ՄՀց*-ից մինչև 130 *ՄՀց*, 0,5 *ՄՀց* քայլով: Սնման լարման արժեքը փոփոխվում է 0,9 *Վ*–ից մինչև 1,6 *Վ*, 0,5 *մլՎ* քայլով: Որպես փոփոխվող լարման և հաճախության տիրույթ է ընտրվել MPEG4 ենթաբլոկը [37]:

Աղյուսակ 1.2

Բազմասնուցում նախագծման, դինամիկ լարման և հաճախության մասշտաբավորման եղանակով կատարված փորձարարական արդյունքների ամփոփումը

Պարամետրերը	Ստանդարտ տարբերակ	Բազմասնուցում նախագծման ԴԼՀՄ տարբերակ
Հաճախություն *(ՄՀց)*	123	5…130
Սնման լարում *(Վ)*	1,2	0,9…1,6
Ընդհանուր հզորություն *(մլՎտ)*	741	350
Հապաղումներ *(նվ)*	89	138

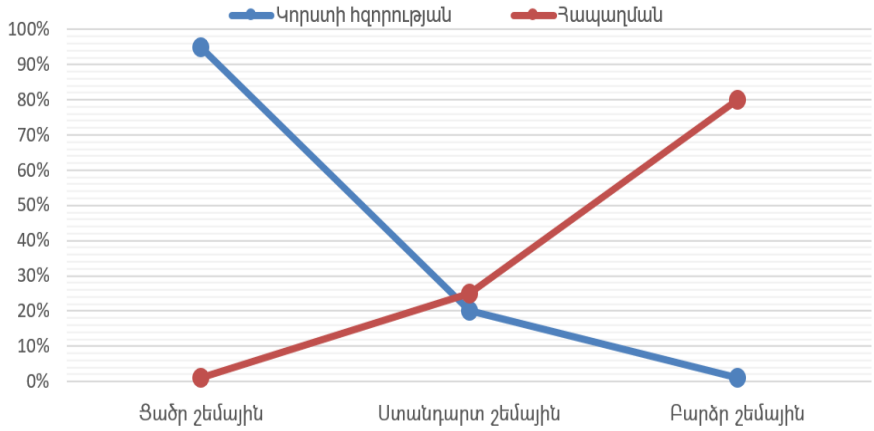
Տվյալ դեպքում ստացվել է ընդհանուր հզորության նվազարկում մոտ 53%-ով, որը բավականին բարձր ցուցանիշ է, բայց նաև կա մեծ ազդեցություն մակերեսի և արագագործության վրա, որոնց մեծացումը պայմանավորված է ԴԼՀՄ կառավարման հանգույցով, որը բավականին բարդ կառուցվածք ունի և աշխատանքի ընթացքում օգտագործում է ԻՍ-ի տարբեր հանգույցներից ստացված տվյալները՝ հանգեցնելով մեծ հապաղումների: Այսպիսով, ամփոփելով ստացված արդյունքները և այս բաժնում տեղ գտած հիմնավորումները՝ կարելի է եզրակացնել, որ բազմասնուցում նախագծման մեթոդն իր տարբեր եղանակներով բավականին արդյունավետ է պրոցեսորային համակարգերում կիրառելու համար, բայց և ունի թերություններ, ինչպիսիք են՝ ժամանակային հապաղումները և դրանց վերլուծությունը, մակերեսը և հատակագծումը, նախագծման բարդությունը և տարբեր սինքրոն համակարգերի ժամանակային համադրությունը:

1.2.2. Բազմակի շեմային լարումներով նախագծման մեթոդի կիրառմամբ պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցումը

Ոչ վաղ անցյալում շեմային լարումները ԻՍ-ում բավականին մեծ արժեք ունեին, դրանք տատանվում էին 0,8…1,2 Վ լարման միջակայքում: Դա հնարավոր էր, քանի որ սնման լարումները տատանվում էին 2,5…3,3 Վ սահմաններում: Արդի տեխնոլոգիական գործընթացում սնման լարման արժեքն ունի կրկնակի անկում և տատանվում է 0,9…1,2 Վ լարման տիրույթում այնքան, որքան ժամանակին շեմային

լարման արժեքն էր, հետևաբար՝ անկում է ապրել նաև շեմային լարման արժեքը, որն այժմ տատանվում է 0,25…0,4 Վ լարման տիրույթում [38]:

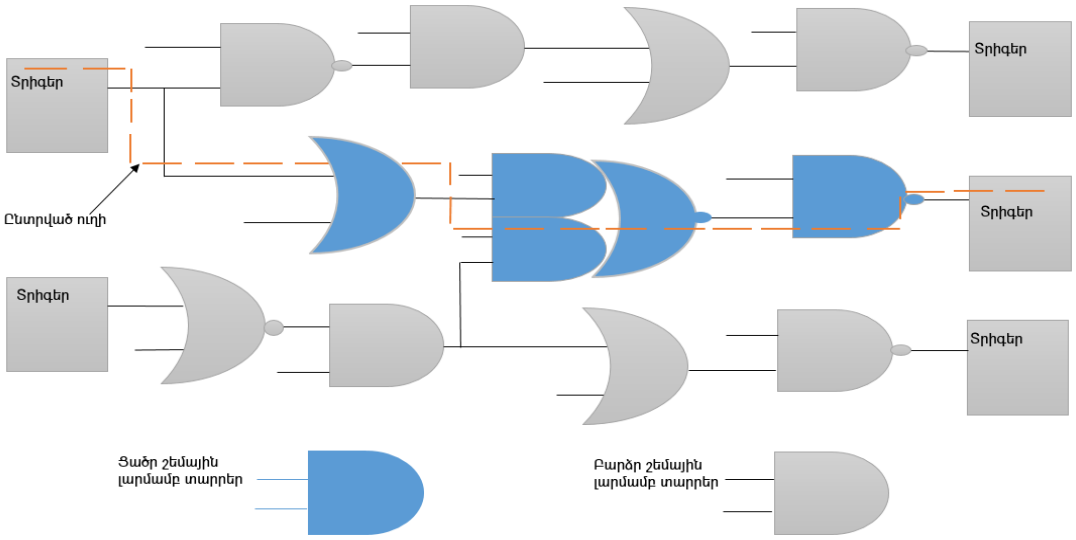
Սնման լարման իջեցման արդյունքում զգալի անկում է ունեցել ԻՍ-ի էներգասպառումը՝ հիմնականում պայմանավորված դինամիկ հզորության նվազումով: Միևնույն ժամանակ, շեմային լարման և տեխնոլոգիական գործընթացով նախագծված տրանզիստորների չափերի նվազման արդյունքում ավելի մեծ դերակատարում սկսեցին ունենալ կորստի հոսանքները [39]՝ առաջացնելով անհրաժեշտություն՝ նախագծման ժամանակ կիրառելու բազմակի շեմային լարումներով աշխատող տրանզիստորներ: Ինչպես արդեն քննարկվել է, շեմային լարումը էքսպոնենցիալ ֆունկցիա է ենթաշեմային կորստի հոսանքից, միաժամանակ, հապաղումները ավելի թույլ կախվածության մեջ են գտնվում շեմային լարումից (նկ. 1.14) [38,39]: Արդի ստանդարտ բջիջների գրադարանները հիմնականում պարունակում են երեք տիպի շեմային լարումներով տարրեր՝ ցածր, ստանդարտ և բարձր, որոնք հնարավորություն են տալիս ծրագրային միջոցին և նախագծողին միաժամանակ իրականացնել հապաղումների և հզորության օպտիմալացում:



Նկ. 1.14. Տրանզիստորի տարբեր շեմային լարումների դեպքում կորստի հզորության և հապաղման կախվածությունը

Տարբեր շեմային լարումներով տարրերի օգտագործումը հնարավորություն է տալիս ազդել կորստի հզորության վրա՝ հնարավորություն ընձեռելով կատարել փոխատեղումներ այն ուղիներում, որտեղ անհրաժեշտ է ունենալ բարձր արագագործություն, տեղադրել ցածր շեմային լարումով աշխատող հանգույցներ, որոնք արագագործ են, բայց, միևնույն ժամանակ, ունեն մեծ կորստի հզորություն [40]:

Այն ուղիներում, որտեղ անհրաժեշտ է ունենալ ցածր էներգասպառում, կարելի է օգտագործել բարձր շեմային լարումով աշխատող հանգույցներ՝ ապահովելով ցածր էներգասպառում, բայց, միաժամանակ, ունենալ ավելի մեծ հապաղումներ: Այս մեթոդի կիրառման արդյունքում հնարավոր է գտնել բալանսավորված միջինը երկու գործոնների միջև՝ հապաղման և էներգասպառման (նկ. 1.15):



Նկ. 1.15. Բազմակի շեմային լարումների կիրառմամբ սխեմատեխնիկական հանգույցը

Գոյություն ունեն բազմակի շեմային լարումներով նախագծման հնարավոր երեք եղանակներ. առաջին՝ օգտագործել միայն բարձր շեմային լարումներով աշխատող տարրեր, երկրորդ՝ օգտագործել միայն ցածր շեմային լարումներով աշխատող տարրեր, երրորդ՝ օգտագործել միաժամանակ երկուսն էլ՝ ցածր և բարձր շեմային լարումներով տարրեր՝ օգտվելով վերևում նկարագրված սկզբունքներից [41]: Ստորև բերված են արագագործ պրոցեսորային համակարգի համար կատարված փորձարարական արդյունքները (աղ. 1.3):

Աղյուսակ 1.3

Բազմակի շեմային լարումներով նախագծման երեք եղանակներով կատարված փորձարարական արդյունքների ամփոփումը

Պարամետրերը	Բարձր շեմային լարումով տարրեր	Ցածր շեմային լարումով տարրեր	Բարձր և ցածր շեմային լարումով տարրեր
Հապաղումներ/ուշացումներ *(պվ)*	-53	0	0
Դինամիկ հզորություն *(մլՎտ)*	3,2	3,3	3,2
Ստատիկ հզորություն *(նՎտ)*	914	3873	1519

Ստացված արդյունքները ցույց են տալիս, որ երեք տարբերակներում էլ հզորության դինամիկ բաղադրիչը մնում է անփոփոխ: Կիրառելով միայն բարձր շեմային լարումով տարրեր՝ իսկապես ստացվում է ստատիկ հզորության բավականին նվազեցում, բայց, միևնույն ժամանակ, առաջանում է ժամանակային սխալ: Միայն ցածր շեմային լարում կիրառելու դեպքում ժամանակային պարամետրերը բավարարում են տեխնիկական առաջադրանքին, բայց և ստացվում է ամենամեծ ստատիկ հզորության արժեքը, որը գրեթե չորս անգամ ավելի մեծ է, քան նախորդ տարբերակում ստացված արդյունքը [41]: Կիրառելով միաժամանակ և՛ բարձր, և՛ ցածր շեմային լարումներով տարրեր, ստացվել է այնպես, որ ժամանակային պարամետրերը բավարարել են տեխնիկական առաջադրանքը, իսկ ստատիկ հզորության բաղադրիչը ստացվել է ավելի քան երկու անգամ փոքր նախորդ տարբերակից և 0,7 անգամ մեծ առաջին տարբերակից:

Ամփոփելով ստացված կիրառական արդյունքները և բազմակի շեմային լարումներով նախագծման մեթոդը՝ կարելի է հանգել եզրակացության, որ տվյալ մեթոդը բավականին արդյունավետ է ստատիկ հզորության նվազարկման տեսանկյունից, իսկ փորձարարական արդյունքներն ապացուցում են, որ տվյալ մեթոդի կիրառման լավագույն տարբերակը միաժամանակ և՛ ցածր, և՛ բարձր շեմային լարումներով տարրերի համադրումն է, որը հնարավորություն է տալիս ունենալ բալանսավորում հզորության և հապաղումների միջև [42]:

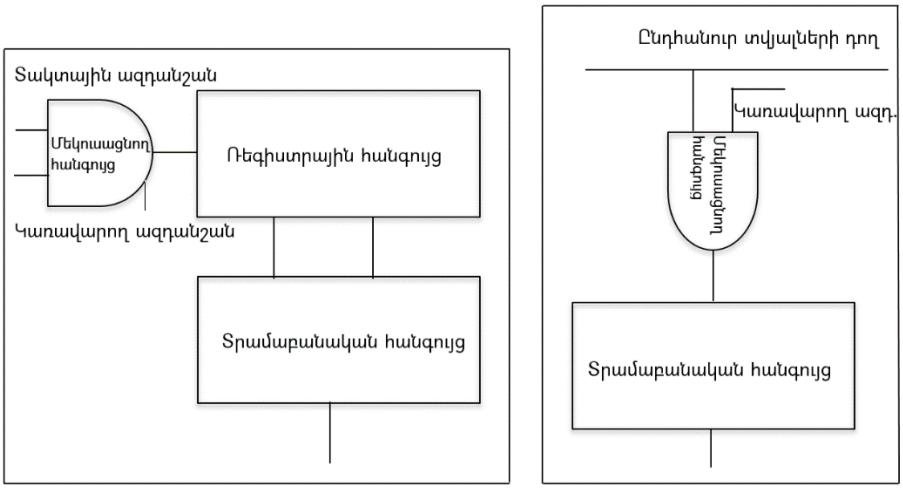
1.2.3. Սինքրոազդանշանի շրջափակմամբ նախագծման մեթոդի կիրառմամբ պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցումը

Դինամիկ հզորության մեծ մասը՝ 50 և ավելի տոկոսը, բաշխվում է տակտային ազդանշանի ցանցի և նրա մասը կազմող կրկնիչների վրա: Տակտային ազդանշանն ամենից հաճախակի է փոխանջատվում ԻՍ-ում, և նրա ուղում գտնվող բազմաթիվ կրկնիչներն ունեն բավականին մեծ ելքային տանող ուժ, որը և ավելացնում է էներգասպառումը տվյալ շղթայում [43]: Պրոցեսորային ԻՍ-ում տակտային ազդանշանն ունի առաջնային նշանակություն թե՛ աշխատունակության տեսանկյունից և թե՛ էներգասպառման, քանի որ գրեթե բոլոր համակարգերը, որոնք պրոցեսորի մաս

են կազմում, սինքրոն թվային հանգույցներ են, որոնք կառավարվում են տակտային ազդանշանի միջոցով: Ուստի արդիական է տակտային ազդանշանի ցանցի էներգասպառման նվազարկումը: Պրոցեսորային ԻՍ–երում կատարված հետազոտություններն ապացուցում են, որ, այսպես կոչված, <<սպասման>> օպերացիոն վիճակում գտնվելու ժամանակ ձևավորված հզորությունը կազմում է ընդհանուր էներգասպառման զգալի մասը [43]: Ծրագրային ապահովումները, որոնք անմիջականորեն աշխատում են պրոցեսորային ԻՍ-երի հետ, հիմնականում գործում են հետևյալ կերպ. կարճ ժամանակահատվածներում ցուցադրում են գերբարձր ակտիվություն: Դրանց հետևում են երկար ընդմիջումներով դադարները կամ շատ ցածր արտադրողականությամբ աշխատանքը, ընդհանուր առմամբ առաջացնելով պարզ տեսակետ, որ այն ժամանակահատվածներում, երբ պրոցեսորային ԻՍ–երը չեն կատարում որևէ առաջադրանք, պետք է ունենան զրո դինամիկ և գրեթե զրո ստատիկ հզորություն [44]: Հաշվի առնելով էներգասպառման կարևորությունը այնպիսի համակարգերում, ինչպիսիք են միայն դյուրակիր մարտկոցներից սնուցվող համակարգերը, որտեղ օգտագործվում են պրոցեսորային ԻՍ-երը, առաջանում է արդիական հարցադրում նախագծողների համար, այն է՝ փորձել նվազարկել էներգասպառումը պրոցեսորային ԻՍ-երում, երբ դրանք գտնվում են ոչ ակտիվ տիրույթներում կամ ռեժիմներում: Կարգավորելու համար էներգասպառման նվազարկումը՝ շատ հաճախ օգտագործվում է տակտային ազդանշանի շրջափակման մեթոդը, որի հիմնական գաղափարը պասսիվ վիճակում տակտային ազդանշանի արգելափակումն է՝ հատուկ մեկուսացնող տարրերի միջոցով [44,45]: Պասսիվ վիճակում հնարավոր կորուստներից խուսափելու համար հարկավոր է կատարել երկու հիմնական քայլ՝

- անջատել տակտային ազդանշանը տրիգերների մուտքում՝ կիրառելով տակտային ազդանշանի շրջափակման մեթոդը,
- համոզվել, որ պասսիվ վիճակում գտնվող տրամաբանական տարրերը, որոնք գտնվում նույն ուղու վրա, ինչ անջատվող տրիգերները, ունեն կայուն/անփոփոխ մուտքային ազդանշան, քանի որ մուտքային ազդանշանի փոխանջատումը, նույնիսկ առանց տակտային ազդանշանի, կարող է հանգեցնել ակտիվության և հետևաբար՝ դինամիկ հզորության մեծացման:

Մեկուսացնող տարրերը հիմնականում իրագործվում են երկու մուտքով՝ տրամաբանական <<ԵՎ>> սխեմայի միջոցով, որի մի մուտքին գալիս է տակտային ազդանշանը, իսկ մյուս մուտքին՝ կառավարող ազդանշանը, որն էլ կախված աշխատանքային ռեժիմից, ակտիվացնում կամ անջատում է մեկուսացնող տարրը [45,46]: Օրինակ՝ միկրոպրոցեսորային ԻՍ-ում հրահանգային փաթեթը, բեռնված լինելով հրամանային ռեգիստրներում, վերծանման տրամաբանության միջոցով որոշում է, թե որ տվյալների ճյուղերն են անհրաժեշտ գործողության կատարման համար՝ հետագայում կառավարող ազդանշանի (մեկուսացնող տարրերի) ակտիվացում կամ ապաակտիվացում ապահովելու նպատակով [46]: Քանի դեռ տրամաբանական տարրերի մուտքերը միացված են ռեգիստրներին, դրանք մնում են կայուն այնքան ժամանակ, որքան տակտային ազդանշանը շրջափակված է մնում (նկ. 1.16ա): Այն դեպքում, երբ մուտքային ազդանշանները միացված են ընդհանուր տվյալների դողին, անհրաժեշտ է լինում ներդնել լրացուցիչ մեկուսացնող տարրեր՝ մուտքային ազդանշանների շրջափակման համար [47] (նկ. 1.16բ):



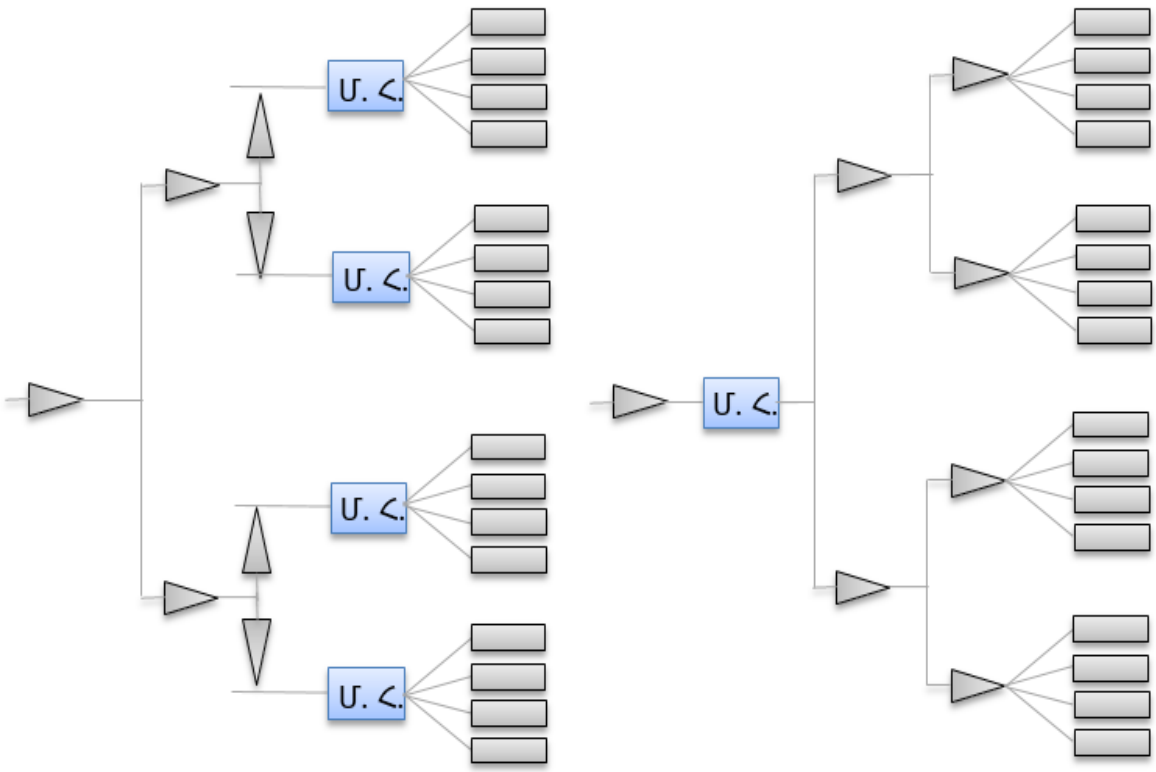
ա) բ)

Նկ. 1.16. Տակտային ազդանշանի շրջափակման ժամանակ կիրառվող մեկուսացման տարրերի տարբերակները. ա) տրամաբանական տարրերի մուտքերը միացված են ռեգիստրներին, բ) մուտքային ազդանշանները միացված են ընդհանուր տվյալների դողին

Արդի տրամաբանական սինթեզի ծրագրային միջոցները կարողանում են հասկանալ և ինտեգրել մեկուսացնող տարրերը՝ համապատասխան ծրագրային

միջոցի ալգորիթմին, տեղորոշելով այն հանգույցները, որտեղ հնարավոր է ներդնել մեկուսացնող տարրերը՝ առանց փոխելու ԻՍ-ի ֆունկցիոնալությունը [46]:

Մեկուսացնող տարրերի կիրառման արդյունքում ԻՍ-ի հապաղումները մեծանում են, և նախագծման ժամանակ անհրաժեշտություն է առաջանում հաշվի առնել ներդրված տարրերի ավելացրած ժամանակային պաշարի համատեղելիությունը նախագծի սահմանափակումների հետ [48]: Մեկուսացնող տարրերը մեծացնում են ելքային բեռը, որը բաշխվում է տակտային ազդանշանի ծառի վրա, մեծացնելով նրա հապաղումները, միաժամանակ առաջացնելով շեղումներ: Մեկուսացնող սխեմաների բաշխման հիմնական երկու տարբերակ գոյություն ունի. առաջին՝ հնարավորինս մոտ տեղաբաշխել մեկուսացնող տարրերը ռեգիստրային հանգույցներին, դրանով իսկ ապահովելով ավելի բարձր կառավարելիության մակարդակ: Առանձնացված եղանակով միացնել կամ անջատել ռեգիստրները, ունենալով ավելի մեծ նվազարկում էներգասպառման մեջ, միաժամանակ բարդեցնելով համակարգը, որի հետևանքով մեծանում են հապաղումները և մակերեսը [48,49] (նկ. 1.17ա): Երկրորդ տարբերակում մեկուսացնող տարրերը տեղաբաշխվում են ավելի հեռու ռեգիստրային հանգույցներից (տակտային ազդանշանի ծառի ավելի բարձր մակարդակներում), որն էլ իր հերթին ապահովում է բաշխման ծառի ավելի բարձր մակարդակից անջատում, ինչը նույնպես հանգեցնում է էներգասպառման նվազարկման այդ հանգույցներում: Միաժամանակ կառավարելիության մակարդակն իջնում է, ինչը նշանակում է, որ հնարավոր չի լինի

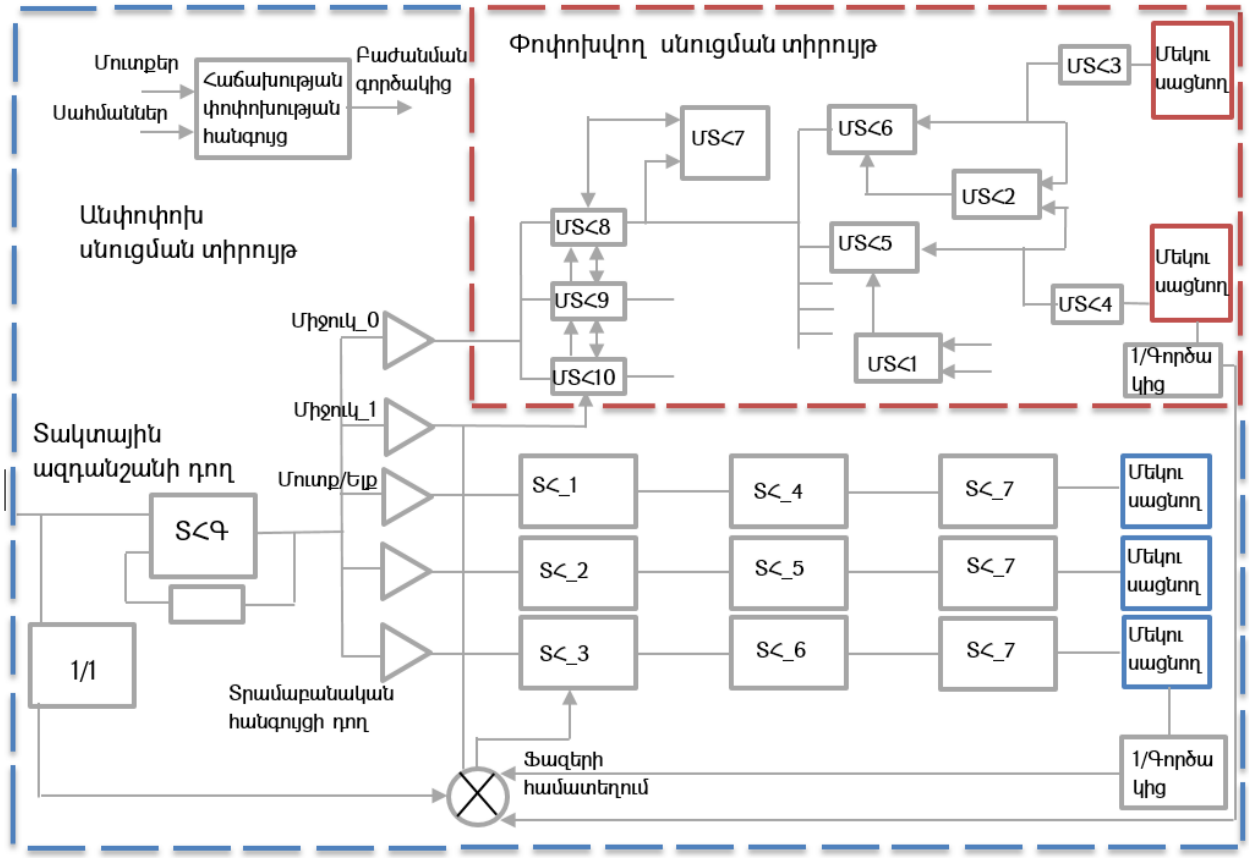


հաճախակի կատարել տակտային ազդանշանի շրջափակում [50] (նկ. 1.17բ):

ա) բ)

Նկ. 1.17. Տակտային ազդանշանի շրջափակման ժամանակ մեկուսացնող տարրերի կիրառման եղանակները. ա) ռեգիստրներին մոտ տեղաբաշխված, բ) ռեգիստրներից հեռու տեղաբաշխված

Վերը նշված մեթոդով նախագծված է Intel Itanium ճարտարապետությամբ պրոցեսորը, բերված են նաև ստացված փորձարարական արդյունքները [51] (աղ. 1.4): Տակտային ազդանշանի շրջափակումը կատարվել է բազմաբիթ ռեգիստրային հանգույցի հիման վրա, քանի որ միաբիթ ռեգիստրում կիրառման պարագայում հզորության նվազման արդյունքները չեն բավարարի տեխնիկական առաջադրանքին: Մեկուսացնող տարրերը տեղադրվել էին ռեգիստրային հանգույցից հնարավոր հեռավորության վրա՝ տակտային ազդանշանի բաշխման ծառի սկզբնամասում (նկ. 1.18): Դիտարկվել են պրոցեսորի աշխատանքային երկու ռեժիմներ՝ հաստատուն և փոփոխական տակտային ազդանշաններով [51]:



Նկ. 1.18. Intel պրոցեսորի հիման վրա մեկուսացնող տարրերի կիրառումը

Աղյուսակ 1.4

Տակտային ազդանշանի շրջափակմամբ, Intel Itanium ճարտարապետությամբ պրոցեսորի փորձարարական արդյունքների ամփոփումը

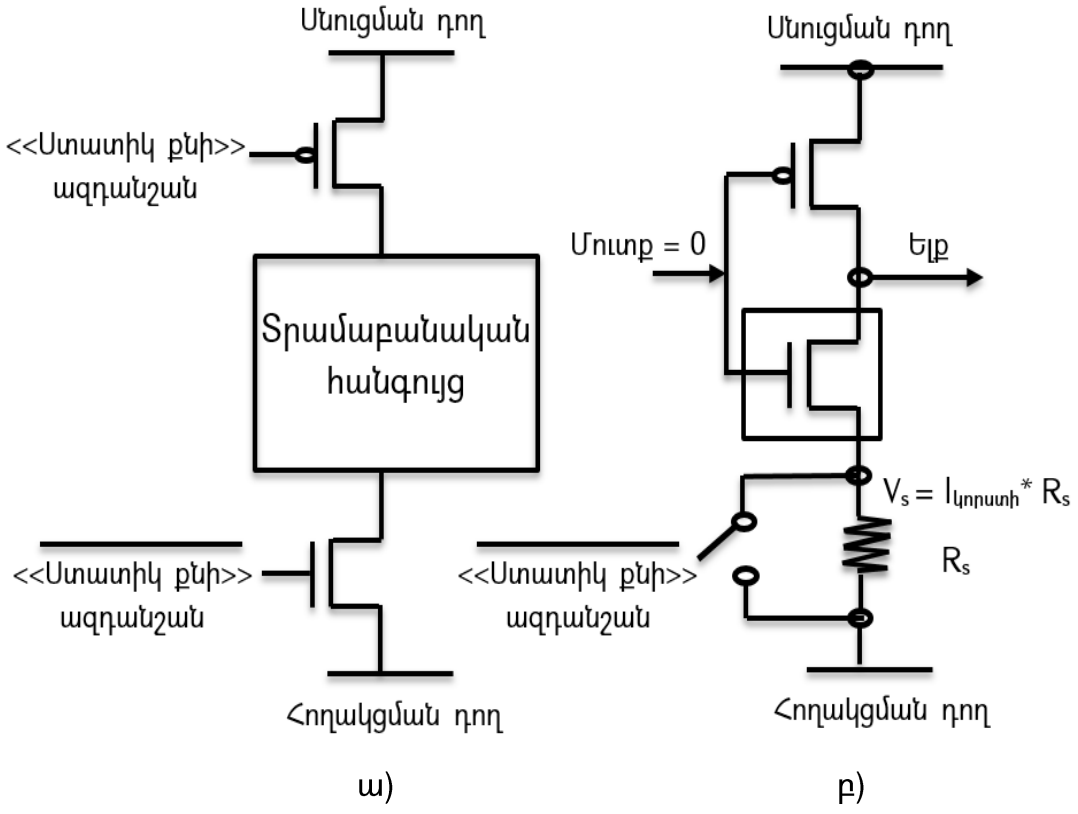
Պարամետրեր	Հաստատուն տակտային ազդանշանով	Փոփոխական տակտային ազդանշանով
Սնուցման լարում *(Վ)*	1,2	1,2
Հաճախություն *(ԳՀց)*	1,6	0,25…2,4
Ընդհանուր հզորություն *(մլՎտ)*	-34%	-38%

Այսպիսով՝ ամփոփելով նշված մեթոդում տեղ գտած բոլոր դրական և բացասական կողմերը, կարելի է եզրակացնել, որ տվյալ մեթոդը բավականին արդյունավետ է դինամիկ հզորության նվազարկման համար՝ համադրելով պասսիվ վիճակի հնարավորությունները: Չնայած զգալի էներգասպառման նվազմանը՝ որպես խնդիրներ արդիական են մնում ժամանակային հապաղումների ավելացումը տակտային ազդանշանի բաշխման ծառի երկայնքով և մակերեսի մեծացումը՝ ի հաշիվ լրացուցիչ ներդրված մեկուսացնող տարրերի:

1.2.4. Հզորության շրջափակմամբ նախագծման մեթոդի կիրառմամբ պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցումը

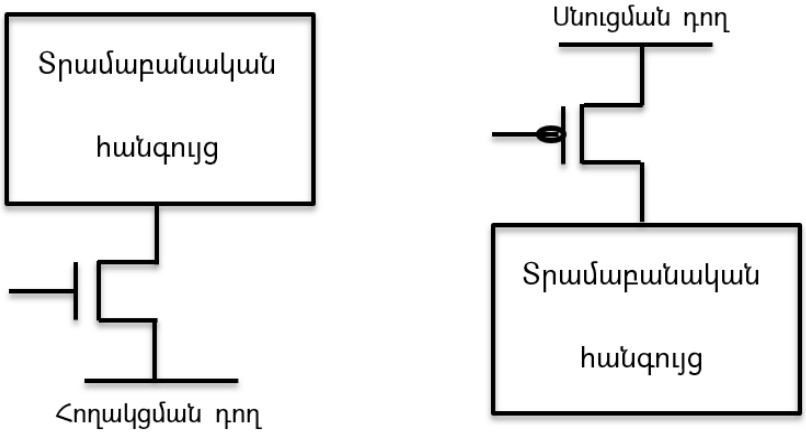
Աճող կորստի հզորության ազդեցության նվազման համար շատ հաճախ անհրաժեշտություն է առաջանում ունենալ համապատասխան մեթոդ, որը հնարավորություն կտա անջատել սնման լարումը այն հանգույցներում, որոնք չեն գտնվում ակտիվ օպերատիվ աշխատանքային ռեժիմում՝ միաժամանակ ունենալով նվազագույն ազդեցություն ԻՍ-ի ֆունկցիոնալության և արագագործության վրա, երբ այն կատարում է ակտիվ գործողություններ: Թվարկված բոլոր պարամետրերը վերաբերում են հզորության շրջափակման մեթոդին (ՀՇՄ) [52]: Այս մեթոդի հիմնական նպատակն է ունենալ աշխատանքային երկու վիճակ՝ ցածր էներգասպառման և ակտիվ աշխատանքային: Մեթոդը հնարավորություն է տալիս փոխել ԻՍ-ի վիճակը, որպեսզի ապահովի առավելագույն հզորության նվազարկում: Ի տարբերություն քննարկված այլ մեթոդների, որոնք հնարավոր էր ներդնել թե՛ համակարգային (միկրոճարտարապետության) և թե՛ սինթեզի (սխեմատեխնիկական) մակարդակներում, տվյալ մեթոդը պահանջում է փոփոխություններ համակարգային մակարդակից՝ կարգավորելու համար ԻՍ-ի ռեժիմները: Ավելին՝ ՀՇՄ-ը ունենում է շատ

ավելի մեծ ազդեցություն ԻՍ-ի ժամանակային պարամետրերի վրա, քան, օրինակ՝ տակտային ազդանշանի շրջափակման մեթոդը՝ ի հաշիվ ինտեգրված նոր տարրերի [52-53]: Կորստի հզորության նվազարկելու լավագույն տարբերակը կլիներ սնուցման ուղղակի անջատումը լարման աղբյուրից ընտրված հանգույցում, եթե առկա են իդեալական փականներ: Հաջորդ տարբերակը կլիներ փականների օգտագործումը, որոնք աշխատում են որպես մեծ դիմադրություններ՝ հանդես գալով որպես վիրտուալ սնման և հողակցման հանգույցներ: Նշված տարրերը կոչվում են փոխանջատող, որոնք միացված են իրական սնուցմանը և հողակցմանը [53]: Որպես այդպիսի տարրեր շատ հաճախ կիրառվում են տրանզիստորներ՝ համապատասխանաբար N և P տիպի: ՀՇՄ-ի տվյալ կիրառության տարբերակը շատ ավելի լավ է գործում, երբ համատեղվում է բազմակի շեմային տրանզիստորների օգտագործմանը (բարձր և ցածր), ապահովելու համար ակտիվ աշխատանքային ռեժիմում արագագործության հնարավոր բարձր արժեք: Այն դեպքում, երբ տարբեր շեմային լարումներով տարրեր են օգտագործվում, ՀՇՄ-ը կոչվում է ՚MTCMOS՚ (նկ. 1.19ա): Ավելացրած փոխանջատող տրանզիստորները մեծացնում են գումարային դիմադրությունը, կորստի հոսանքի ուղում <<ստատիկ քնի>> ռեժիմում՝ միաժամանակ բարձրացնելով տրանզիստորի շեմային լարումը: Դիմադրության և շեմային լարման մեծացումը ԻՍ-ում փոքրացնում է կորստի հոսանքը [53,54] (նկ. 1.19բ):



Նկ. 1.19. ա) Փոխանջատող տրանզիստորներով հզորության շրջափակման մեթոդի կիրառումը, բ) ԿՄՕԿ շրջիչի հիման վրա փոխանջատող տրանզիստորների կիրառումը

Արդեն իսկ քննարկված փոխանջատող տրանզիստորային հանգույցում կիրառվում են երկու տրանզիստորներ՝ վերևից և ներքևից կառավարելիություն ապահովելու համար: Գոյություն ունեն մեթոդներ, որոնք հիմնված են միայն մեկ տրանզիստորով կառավարելի դարձնելու տարբերակի վրա [54]: Դա նախատեսում է նվազեցում կորստի հզորության մեջ, բայց և հնարավորություն է ստեղծում բալանսավորելու ժամանակային հապաղումները: Միայն N-ՄՕԿ տրանզիստորով կառուցվածքն առավել գերադասելի է, քանի որ ունի ավելի փոքր դիմադրություն՝ ի համեմատ P-ՄՕԿ տրանզիստորի, նույն լայնության դեպքում (նկ. 1.20ա): Ավելին, P-ՄՕԿ տրանզիստորը կարող է մի քանի անգամ չափերով գերազանցել N-ՄՕԿ–ը (նկ. 1.20բ) [55]:

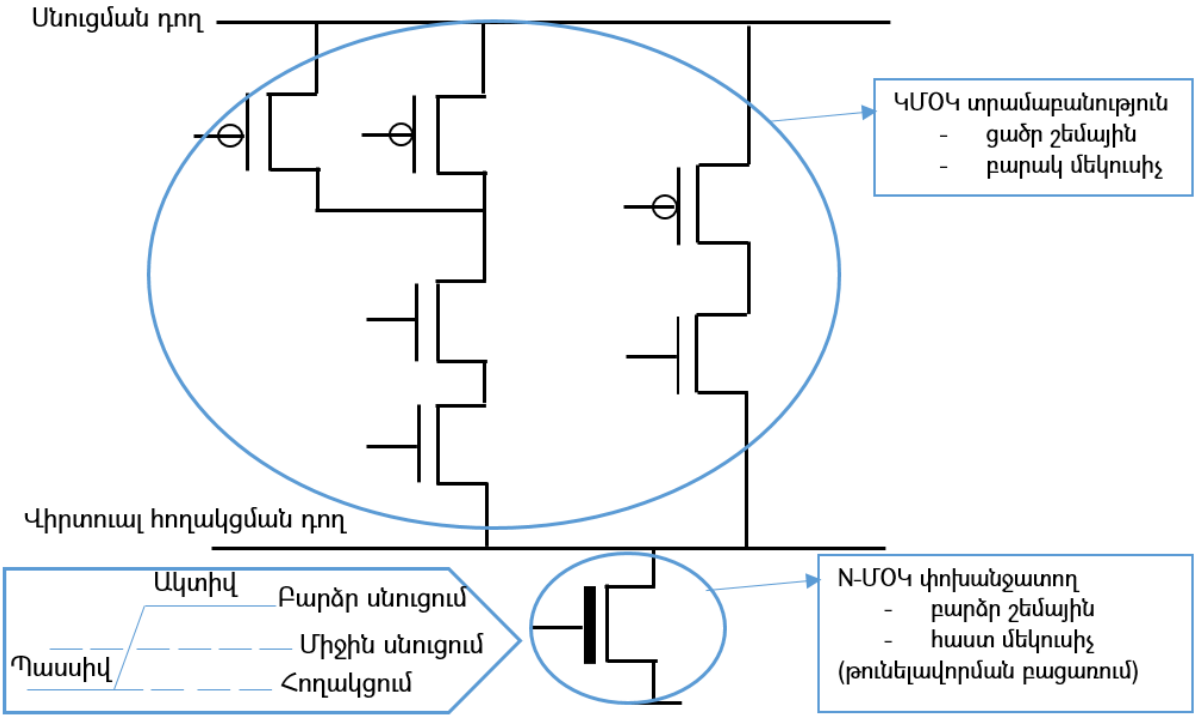


ա) բ)

Նկ. 1.20. ա) Մեկ N-ՄՕԿ տարբերակով հզորության շրջափակման մեթոդի կիրառումը, բ) մեկ P-ՄՕԿ տարբերակով հզորության շրջափակման մեթոդի կիրառումը

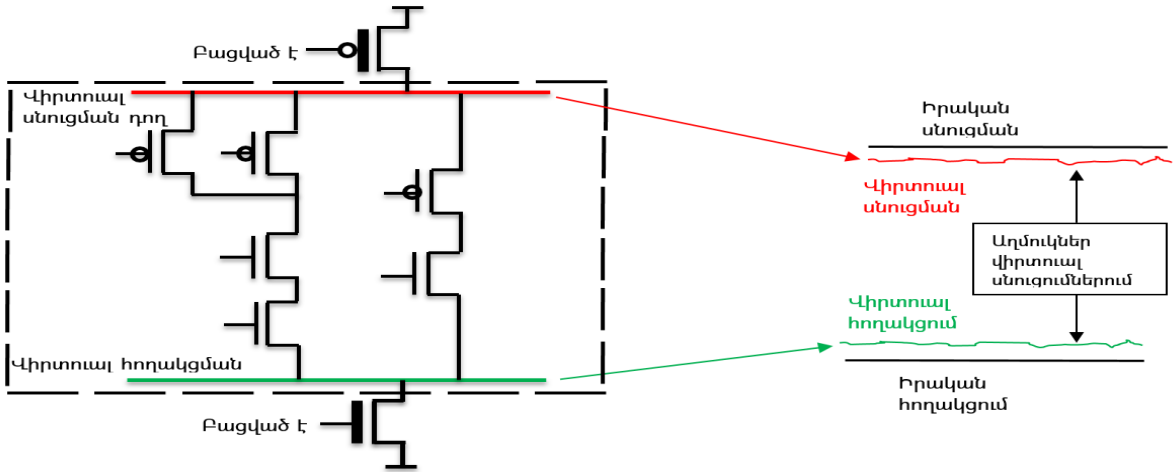
ՀՇՄ-ը ժամանակի ընթացքում ստացել է նոր զարգացումներ, որոնք էլ ավելի են կրճատում կորստի հզորությունը, միաժամանակ ունենալով ավելի փոքր ժամանակային կորուստներ: Ընդունելով որպես հիմք առաջարկվող տարբերակը՝ մեկ N-ՄՕԿ տրանզիստորի կիրառմամբ հնարավոր է փոխանջատող N-ՄՕԿ–ի փականին տալ սնման լարումից ավելի բարձր պոտենցիալ, որը կկրճատի ընդհանուր դիմադրությունը [56]: Միաժամանակ, օգտագործելով բարձր շեմային լարմամբ, հաստ

մեկուսիչով տրանզիստոր՝ ՀՇՄ-ի կիրառության այս տարբերակը կոչվում է BGMOS (նկ. 1.21):



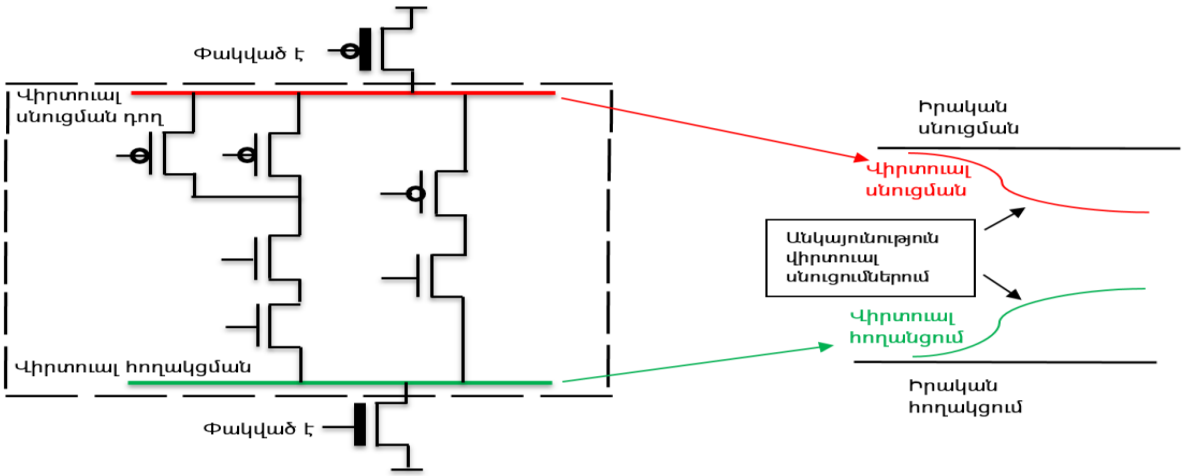
Նկ. 1.21. BGMOS տարբերակով հզորության շրջափակման մեթոդի կիրառումը

Առաջարկված մեթոդների թերություններն են արագագործության վրա ազդեցությունը և լրացուցիչ դիմադրությունների ներդրման (N-ՄՕԿ) արդյունքում սնուցման/հողակցման դողում առաջացած աղմուկները, որոնք ազդում են սնուցման մակարդակի կայունության վրա <<signal integrity>> [57]: Ստեղծվում է իրավիճակ, երբ սնուցման և հողակցման արժեքները տատանվում են, ավելին՝ անցնելով փոխանջատող տրանզիստորների միջով դեպի վիրտուալ սնուցման և հողակցման դողեր, դրանք առաջացնում են զգալի տատանումներ դողերում թե՛ ակտիվ աշխատանքի ռեժիմում և թե՛ ցածր էներգասպառման ռեժիմում (նկ. 1.22ա): Ցածր էներգասպառման ռեժիմում վիրտուալ սնուցման հանգույցներում անցողիկ երևույթներն ավելի երկար են տևում և ընդհանուր առմամբ բնորոշվում են փոխանջատող N և P ՄՕԿ տրանզիստորներով ձևավորված դիմադրությունների



բաժանիչով (նկ. 1.22բ) [58]:

ա)



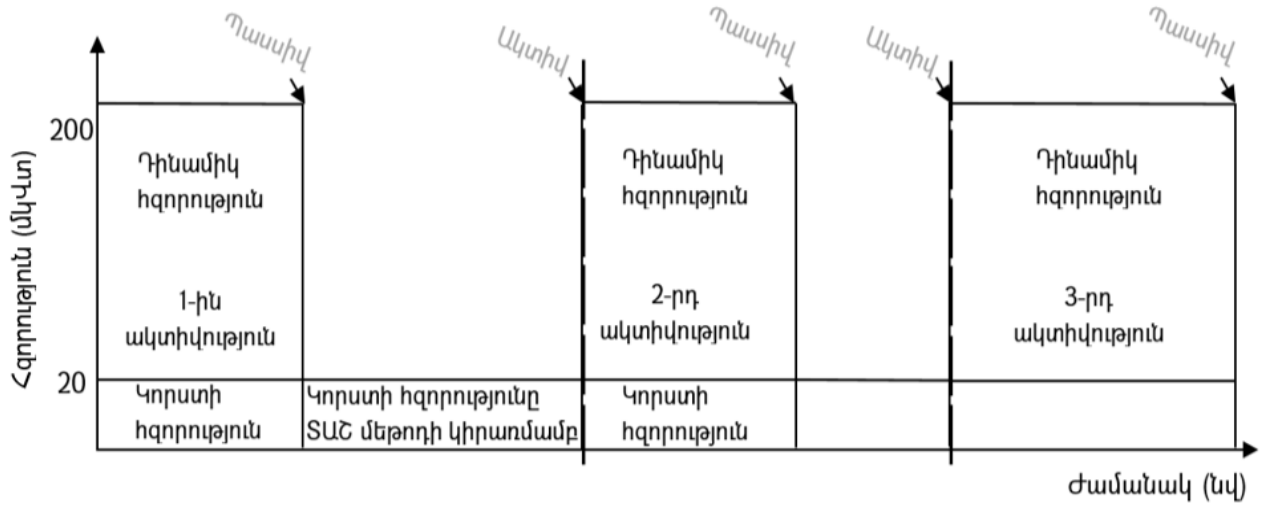
բ)

Նկ. 1.22. ա) Ակտիվ աշխատանքային ռեժիմում ՀՇՄ-ում վիրտուալ սնուցումը, բ) ցածր էներգասպառման ռեժիմում ՀՇՄ-ում վիրտուալ սնուցումը

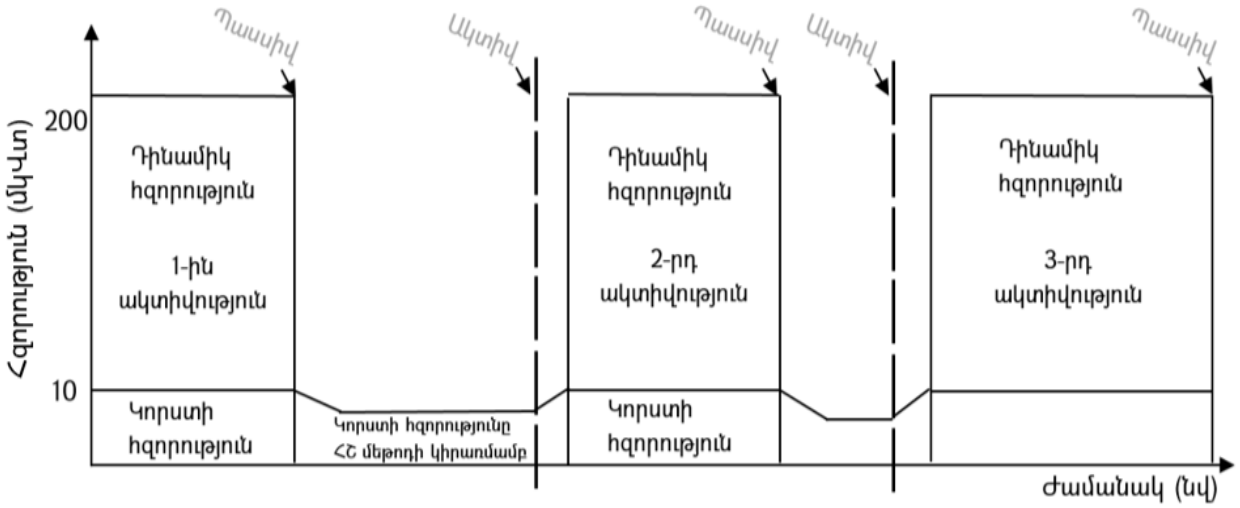
Գոյություն ունեն ՀՇՄ-ի և տակտային ազդանշանի շրջափակման (ՏԱՇ) մեթոդի համեմատական արդյունքներ, որոնք մոդելավորվել են նույն պրոցեսորային հանգույցի համար՝ օգտագործելով ակտիվ աշխատանքային ռեժիմը և ցածր էներգասպառման ռեժիմը:

Համակարգային մակարդակից ստացված արդյունքները շատ կարևոր են՝ հասկանալու համար ընդհանուր էներգասպառման նվազարկումը և ժամանակային հնարավոր հապաղումների փոփոխությունները [59]: Տակտային ազդանշանի շրջափակման մեթոդի կիրառման արդյունքում ցածր էներգասպառման ռեժիմում ստացված կորստի հզորության արժեքը 20 *մկՎտ* է, դինամիկ և կորստի հզորությունների գումարը ակտիվ աշխատանքային ռեժիմում կազմել է 200 *մկՎտ*, միաժամանակ ռեժիմների միջև անցումներում գրանցված ժամանակային հապաղումները բավարարել են սահմանված արժեքները (նկ. 1.23ա): Ռեժիմների աշխատանքը տարբերակելու համար սահմանված են երկու ազդանշան՝ ակտիվ և պասսիվ: Հզորության շրջափակման մեթոդը կիրառելիս ցածր էներգասպառման ռեժիմում ստացվել է կորստի հզորության էլ ավելի իջեցում՝ մոտ 10 *մկՎտ*, դինամիկ և կորստի հզորությունների գումարը ակտիվ աշխատանքային ռեժիմում մնացել է անփոփոխ՝ 200 *մկՎտ*, այնուամենայնիվ, դիագրամից երևում է՝ առկա են ժամանակային հապաղումներ, ռեժիմն ազդարարող ազդանշանից մինչև իրական

ռեժիմի ակտիվացում, որը պայմանավորված է հզորության շրջափակման մեթոդի թերություններով, մասնավորապես՝ ավելացրած սխեմաների և նրանցում անցողիկ երևույթների մեծ տևողությամբ (նկ. 1.23բ) [60]:



ա)



բ)

Նկ. 1.23. ա) Երկու ռեժիմներում ՏԱՇ մեթոդի հզորության և ժամանակի դիագրամը, բ) երկու ռեժիմներում ՀՇՄ-ի հզորության և ժամանակի դիագրամը

Ստորև բերված է ընդհանրացված փորձարարական արդյունքների աղյուսակը՝ երկու մեթոդների համեմատական արժեքներով (աղ. 1.5):

Աղյուսակ 1.5

ՏԱՇ և ՀՇՄ-ի համեմատական արդյունքների ամփոփումը

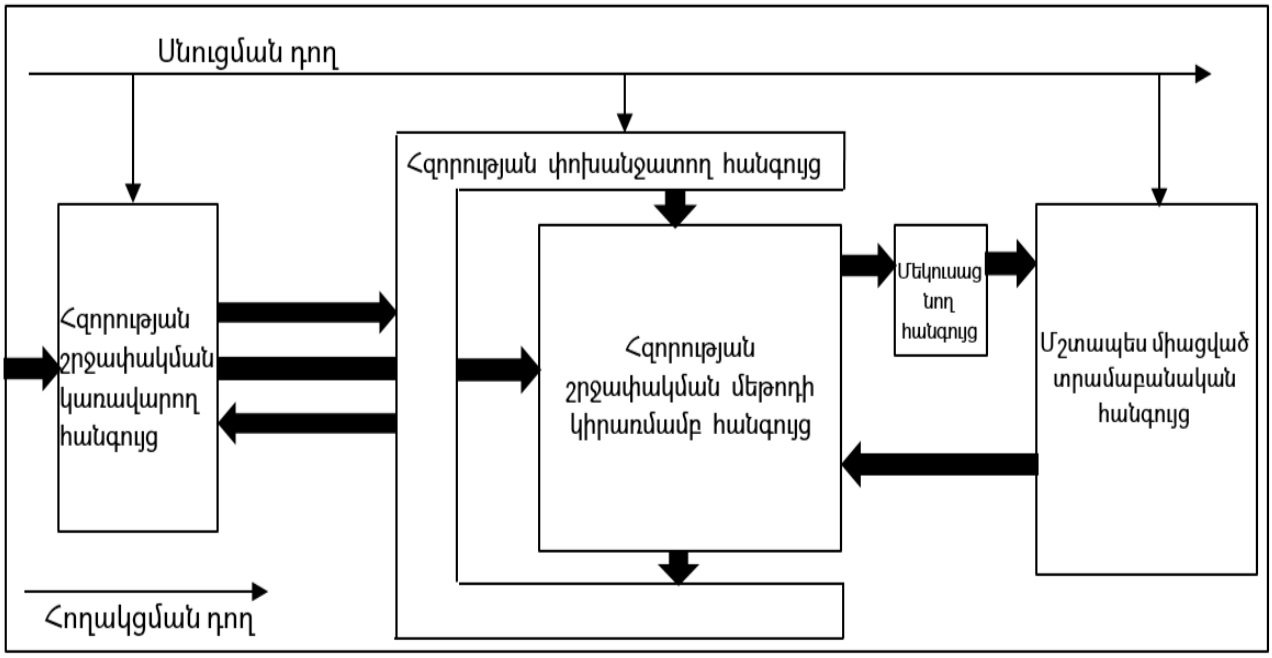
Պարամետրերը	հաստատուն տակտային	փոփոխական տակտային

	ազդանշանով	ազդանշանով
Հապաղումներ *(նվ)*	0	-0,5
Ստատիկ հզորություն *(մկՎտ)*	20	10 (-50%)
Ընդհանուր հզորություն *(մկՎտ)*	200	200

Պրոցեսորային ԻՍ-երում հզորության շրջափակման մեթոդը շատ կիրառելի է, հաշվի առնելով այն հանգամանքը, որ կենտրոնական պրոցեսորային հանգույցը շատ հաճախ չի կիրառվում, հնարավոր է այն ամբողջությամբ ենթարկել մեթոդի ներգործությանը: Այնուամենայնիվ, կան սահմանափակումներ, որոնք պետք է հաշվի առնել [61].

- Վերականգնման ժամանակի արձագանքը ակտիվ/պասսիվ հրամաններին ունի բավականին մեծ ազդեցություն պրոցեսորային ԻՍ-ում: Օրինակ, կարող է պահանջվել ավելի վաղ FIFO-ի դուրսբերումը ընտրված վիճակից, կամ կարող է պահանջվել նախապես պլանավորած ժամանակային տիրույթների սահմանում:
- Այն դեպքում, երբ կենտրոնական պրոցեսորային հանգույցը գտնվում է ցածր էներգասպառման վիճակում, նրանում առկա տեղեկությունը չի պահպանվում, հետևաբար՝ անհրաժեշտ է լինում բավականին մեծ ժամանակ և հզորություն՝ այդ արժեքները վերականգնելու համար՝ մինչ ռեժիմի փոփոխություն կատարելը:
- Հզորության շրջափակման մեթոդի արդյունավետությունը գնահատելու համար անհրաժեշտ է համեմատել մեթոդի կիրառման արդյունքում ստացված էներգախնայումը և վիճակի վերականգման, նոր ռեժիմի փոփոխության համար անհրաժեշտ էներգասպառումը:

Պրոցեսորային ԻՍ-ում ՀՇ մեթոդի կիրառումը պահանջում է ներդնել մի քանի կառավարման հանգույցներ և տարրեր [60-62] (նկ. 1.24): Մշտապես միացված տրամաբանական հանգույցը ուղիղ կապված է սնուցման դողին և միշտ գտնվում է ակտիվ ռեժիմում [62]: Ի տարբերություն այդ հանգույցի՝ ՀՇՄ-ի կիրառմամբ հանգույցը կապված է սնուցման դողին հզորության փոխանջատման հանգույցի միջոցով: Այս ենթաբլոկը հնարավորություն է տալիս կատարել փոխանջատում երկու պոտենցիալների միջև՝ սնուցման և հողակցման:



Նկ. 1.24. Պրոցեսորային ԻՍ-ի համար ՀՇՄ-ի կիրառման օրինակը

Հզորության փոխանջատման հանգույցը հիմնականում բաղկացած է բազմաթիվ ԿՄՕԿ փոխանջատող սխեմաներից, որոնք բաշխված են ՀՇՄ-ով նախագծված հանգույցի ամբողջ երկայնքով կամ էլ ինտեգրված են հենց այդ հանգույցի ներսում: ՀՇՄ-ը կառավարող հանգույցը կառավարում է ԿՄՕԿ փոխանջատող սխեմաները [63-64]: ՀՇՄ-ի նախագծման հիմնահարցերից մեկն այն է, որ ելքային ազդանշանը բավականին դանդաղ է փոխանջատվում, որի արդյունքում երկար ժամանակ է պահանջվում շեմային լարման հաղթահարման համար, որն էլ իր հերթին առաջացնում է խաչաձև հոսանքներ <<մշտապես ակտիվ>> հանգույցում: Այդ երևույթը կանխելու համար տեղադրված են <<մեկուսիչ>> սխեմաներ ՀՇՄ–ում ներդրված հանգույցի այն ելքերին, որոնք միացված են միշտ ակտիվ հանգույցին: Մեկուսիչ սխեմաները նախագծված են այնպես, որ չունեն խաչաձև հոսանք, երբ մուտքերից մեկը գտնվում է

շեմային լարման միջակայքում: ՀՇՄ-ի կառավարող հանգույցը ղեկավարում է մեկուսիչ սխեմաները [63]: Ինչպես արդեն նշվել էր, ՀՇՄ-ի կիրառումը պահանջում է պահպանել վիճակը որոշակի հանգույցներում ցածր էներգասպառման ռեժիմում, որպեսզի ապահովվի հնարավորինս արագ և էներգախնայող վերականգնում ակտիվ աշխատանքային վիճակին անցնելիս [64]: Նմանատիպ հնարավորություն ունենալու տարբերակ են վիճակը պահպանող և վերականգնող ռեգիստրները, որոնք պետք է տեղադրվեն ստանդարտ ռեգիստրների փոխարեն [65]: Այսպիսով, պրոցեսորում ՀՇՄ-ի կիրառման հիմնական բարդությունները հետևյալն են՝

- հզորության շրջափակման կառավարող հանգույցի նախագծումը,
- վիճակը պահպանող վերականգնող ռեգիստրների և մեկուսիչ տարրերի ընտրությունը,
- ՀՇՄ-ի ազդեցության նվազեցումը մակերեսի և ժամանակային պարամետրերի վրա [66]:

Այսպիսով, ամփոփելով հզորության շրջափակման մեթոդը, կարելի է կատարել հետևյալ եզրակացությունը: Մեթոդն ուղղված է կորստի հզորության նվազարկմանը, որը բավականին արդյունավետ է պրոցեսորային ԻՍ-երի պարագայում՝ հաշվի առնելով մեթոդի համակարգային հանգույցներում կիրառության լայն հնարավորությունները: Այնուամենայնիվ, գոյություն ունեն նաև խնդիրներ, որոնք հիմնականում ունենում են ազդեցություն մակերեսի և ժամանակային պարամետրերի վրա: Մասնավորապես, ավելացնելով փոխանջատող տրանզիստորները, մեկուսիչ տարրերը, վիճակը պահպանող և վերականգնող ռեգիստրները, ստացվում է զգալի մեծացում մակերեսում: Ներդնելով երկու ռեժիմներ՝ առաջանում է դրանց աշխատաժամանակների համադրության անհրաժեշտություն, մասնավորապես՝ ցածր էներգասպառման ռեժիմում կիրառվող վիճակի վերականգնման ժամանակի հետ կապված:

1.2.5. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման ժամանակ կիրառվող սխեմատեխնիկական հանգույցները

Պրոցեսորային ԻՍ-երի ցածր էներգասպառմամբ նախագծումը՝ տարբեր մեթոդների կիրառմամբ, նախատեսում է լրացուցիչ սխեմատեխնիկական հանգույցների ներգրավում՝ ապահովելու համար ինտեգրումը: Մի քանի նմանատիպ տարրեր արդեն իսկ քննարկվել են նախորդ բաժիններում, բայց և կան սխեմատեխնիկական հանգույցներ, որոնց առանձին անդրադարձ կկատարվի այս բաժնում: Քննարկման առարկա են դառնալու վիճակը պահպանող-վերականգնող ռեգիստրները և մակարդակի փոխակերպիչները [67]:

Վիճակը պահպանող վերականգնող ռեգիստրներ: Վիճակի պահպանում և վերականգնում ապահովելու համար շատ հաճախ օգտագործվում է վիճակի վերականգնման ռազմավարություն, որը կարող է ապահովել վիճակի շատ արագ պահպանում և վերականգնում՝ համակարգը բերելով ամբողջովին ֆունկցիոնալ վիճակի: Գոյություն ունեն վիճակի վերականգնման ռազմավարության գործնական կիրառման հիմնական երեք տարբերակներ [68].

- ծրագրային ապահովման վրա հիմնված՝ կապված ռեգիստրների գրել/կարդալու ֆունկցիոնալ հնարավորությունների հետ,
- թեստավորման ուղիների վրա հիմնված՝ հիմնական նպատակ ունենալով օգտագործելու թեստավորման համար նախատեսված ուղիները վիճակի պահպանման և վերականգնման համար,
- ռեգիստրների վրա հիմնված՝ օգտագործելով վիճակը պահպանող և վերականգնող (ՎՊՎ) տրիգերներ [69]:

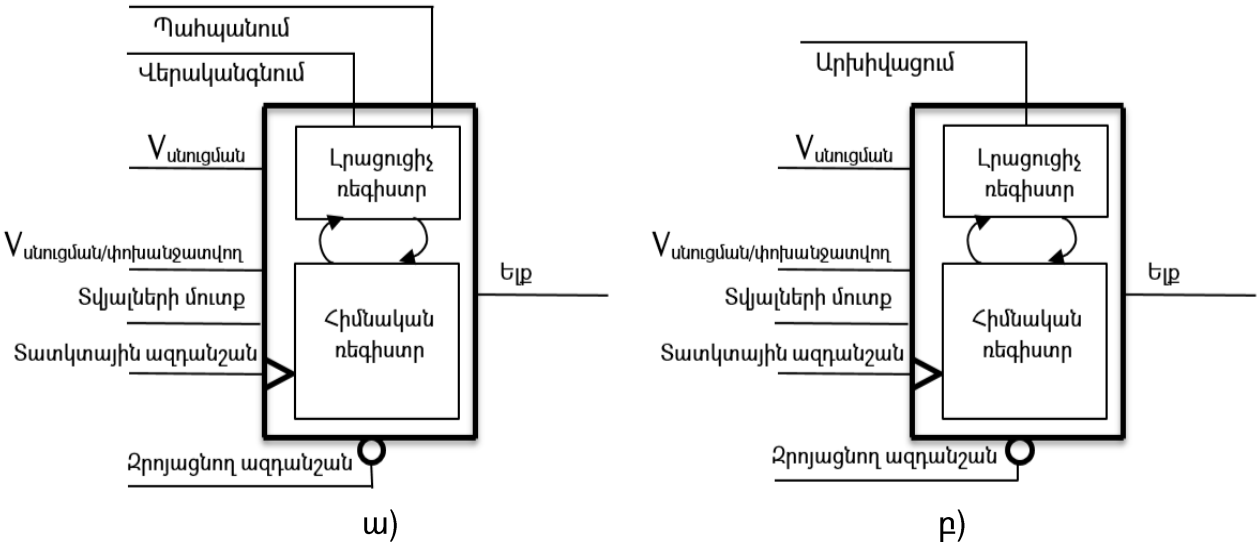
Ծրագրային ներդրման հիմնական գաղափարն այն է, որ սնուցումից անջատված հանգույցի պահպանման ենթակա տվյալները գրանցվում են պրոցեսորի հիշողության ենթաբլոկում: Տվյալ տարբերակն առաջացնում է բազմաթիվ խնդիրներ թե՛ նախագծման և թե՛ կիրառման տեսանկյունից: Առաջանում են դժվարություններ ծրագրի բարդության հետ կապված, ժամանակային կառավարման տեսանկյունից, երբ կատարվում է սնուցման անջատում, ազդանշանների գրանցում, ընթերցում և այլն: Այդ իսկ պատճառներով այս տարբերակը առավել քիչ օգտագործվող տարբերակ է [68]:

Թեստավորման ուղիների վրա հիմնված տարբերակում նախապես ընտրված ուղիները աշխատում են այնպես, ինչպես թեստավորման ռեժիմում, բայց նրանց ելքերը, ի տարբերություն ստանդարտ թեստավորման, միացված են հիշողության

սարքին, որը կարող է լինել թե՛ ներքին և թե՛ արտաքին: Ներքինի պարագայում հիշողության հանգույցը պետք է միշտ միացված լինի սնուցմանը [68,70]: Ապահովելու համար տվյալների հնարավոր արագ գրանցում հիշողության սարքում, անհրաժեշտ է օգտագործել նրա կարգային ամբողջ լայնությունը, բայց, որպես կանոն, թեստավորման ուղիները գրեթե միշտ ավելի պակաս կարգայնություն ունեն, քան հիշողության կարգայնությունն է, ինչի պատճառով առաջանում են ժամանակային հապաղումներ: Միաժամանակ, թեստավորման ուղիներում գտնվող ռեգիստրներով անցում ապահովելու համար կարող է պահանջվել շատ մեծ դինամիկ հզորություն, որը պայմանավորված է այդ ուղում գտնվող ռեգիստրների հաճախակի փոխանջատումներով: Արտաքին հիշողության պարագայում նույնպես ստացվում է մեծ էներգասպառում՝ ի հաշիվ մուտք/ելք հանգույցների [70]: Այս մեթոդի կիրառումը կարող է արդյունավետ լինել այն պարագայում, երբ գոյություն ունի շատ երկար ժամանակով սնուցման անջատման հնարավորություն:

Ռեգիստրների վրա հիմնված տարբերակի դեպքում հիմնական ռեգիստրները փոխարինվում են վիճակը պահպանող և վերականգնող ռեգիստրներով, որոնք ունեն լրացուցիչ հանգույց, որի միջոցով կարողանում են պահպանել ռեգիստրի վիճակը, երբ հանգույցում սնուցման լարումն անջատված է [69]: Հաջորդիվ՝ կարողանում են վերականգնել պահպանված վիճակը՝ սնուցման լարումը միացնելուն պես: Լրացուցիչ հանգույցը ռեգիստրում միշտ միացված է սնուցման լարմանը: Այդ հիշող սարքի կառուցվածքը հետևյալն է. երկու հիմնական ռեգիստրային հանգույցները սնուցվում են $V_{\text{սնուցման/փոխանջատվող}}$ դողից, որը, կախված ռեժիմից, փոխանջատվում է: Երրորդ լրացուցիչ ռեգիստրը սնուցվում է $V_{\text{սնուցման}}$ դողից, որը միշտ միացված է սնուցման լարմանը: <<Պահպանում>> ազդանշանի ակտիվացվելուն պես՝ հիմնական ռեգիստրի արժեքը գրանցվում է լրացուցիչ ռեգիստրում: <<Վերականգնում>> ազդանշանի ակտիվացվելուն պես՝ լրացուցիչ ռեգիստրի արժեքը հետ է գրանցվում հիմնական ռեգիստրում:

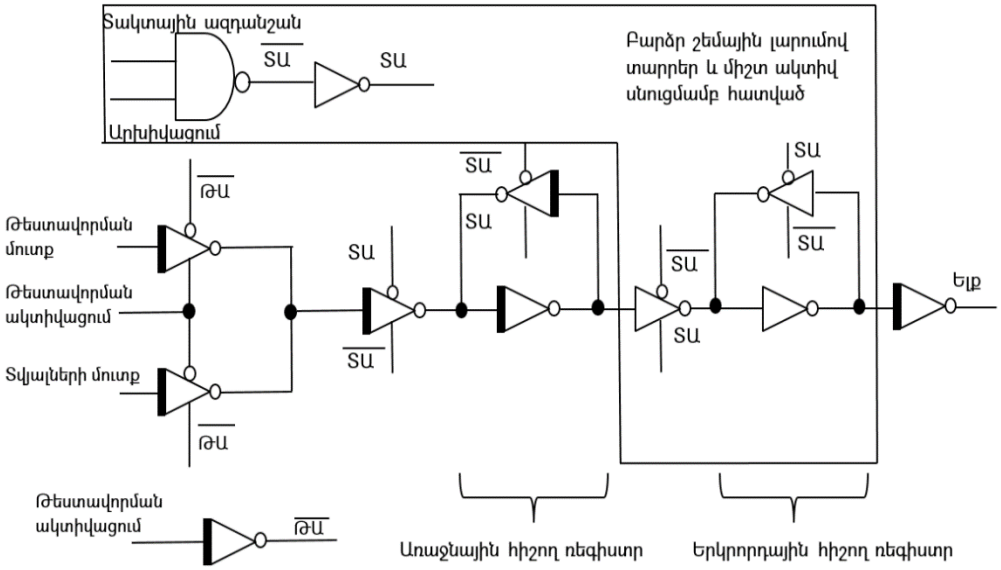
Այս տարբերակում պահպանում/վերականգնում ազդանշանների կառավարումը կատարվում է ազդանշանի մակարդակով (նկ. 1.25ա) [60,71]: Մյուս տարբերակում, <<արխիվացում>> ազդանշանի ակտիվացնելուն պես, հիմնական ռեգիստրի պարունակությունը գրանցվում է լրացուցիչ ռեգիստրում, ապաակտիվացնելուն պես՝ տվյալները հետ են գրանցվում հիմնական ռեգիստրում: Արխիվացում ազդանշանի կառավարումը կատարվում է ազդանշանի ճակատով (նկ. 1.25բ):



Նկ. 1.25. ա) Տարբերակ 1. վիճակը պահպանող-վերականգնող ռեգիստրը,
բ) տարբերակ 2. վիճակը պահպանող-վերականգնող ռեգիստրը

ՎՊՎ ռեգիստրներն իրենց մակերեսով գերազանցում են դասական ռեգիստրներին 20 կամ ավելի տոկոսով: Ռեգիստրների մեծ քանակության պարագայում մակերեսի մեծացումը բավականին զգալի է լինում: ՎՊՎ ռեգիստրի՝ մեկ արխիվացնող մուտքով տարբերակի պարագայում կատարվում են հնարավոր նվազագույն փոփոխությունները սխեմատեխնիկական մակարդակում, որը և ապահովում է զգալի փոքր կորստի հզորություն և մակերես (նկ. 1.26) [69-71]: Այդ ՎՊՎ ռեգիստրի մուտքին տեղադրված է մուլտիպլեքսոր, որը ընտրում է համապատասխանաբար, թե որ ռեժիմում է սխեման գտնվում՝ թեստավորման, թե՛ ակտիվ տվյալների փոխանցման: Առաջնային հիշող ռեգիստրը աշխատում է ՏԱ–ի ցածր մակարդակով, միաժամանակ պարունակում է ցածր շեմային լարմամբ տարրեր՝ բարձր արագագործություն ստանալու նպատակով [71]: Երկրորդային ռեգիստրը աշխատում է ՏԱ–ի բարձր մակարդակով, ելքում տեղակայված ցածր շեմային լարմամբ

շրջիչը նախատեսված է տարբեր ելքային տանող ուժերի պարագայում համակարգի արագագործությունն ապահովելու համար:



Նկ. 1.26. Վիճակը պահպանող վերականգնող ռեգիստրի սխեմատեխնիկական կառուցվածքը

Ներկայացված ՎՊՎ ռեգիստրի աշխատանքի սկզբունքը հետևյալն է.

- Տրամաբանական ԵՎ-ՈՉ տարրերը ներդրվել են՝ ապահովելու համար տակտային ազդանշանի կախվածությունը արխիվացնող ազդանշանից:
- Ցածր էներգասպառման ռեժիմում հնարավոր է իջեցնել սնուցման լարումը այս հանգույցում՝ ապիովելու համար հնարավորինս քիչ կորստի հոսանքներ:

Տվյալ ռեգիստրի առավելություններն են [72].

- պահանջում է նվազագույն մակերեսի ավելացում,
- կառավարվում է միակ մեկ լրացուցիչ ազդանշանի միջոցով:

Տվյալ ռեգիստրի թերություններն են [72].

- առաջանում են ժամանակային հապաղումներ, որոնք պայմանավորված են այն հանգամանքով, որ երկրորդային հիշող ռեգիստրը կազմված է ամբողջովին բարձր շեմային լարումներով աշխատող տարրերից, որոնք նախատեսված են հնարավորինս փոքր կորստի հզորություն ունենալու պահանջով, ցածր էներգասպառման ռեժիմում,
- բարձր շեմային լարումներով աշխատող տարրերը տակտային ազդանշանի ուղում բարձրացնում են տեղակայման ժամանակի սահմանափակումները:

Ստորև բերված են երկու կառուցվածքներով (մեկ կառավառող մուտքով և երկու կառավառող մուտքերով) ՎՊՎ ռեգիստրների համեմատական արդյունքները [72] (աղ. 1.6):

Ամփոփելով կարելի է եզրակացնել, որ մեկ արխիվացնող մուտքով ՎՊՎ ռեգիստրը ավելի էներգախնայող է թե՛ ընդհանուր հզորության (-6%) և թե՛ կորստի հզորության տեսանկյունից ակտիվ (-5%) և ցածր էներգասպառման (-10%) ռեժիմներում: 40% -ով ավելի քիչ մակերես է զբաղեցնում, միևնույն ժամանակ, ավելի դանդաղագործ է մոտ 3% -ով:

Աղյուսակ 1.6

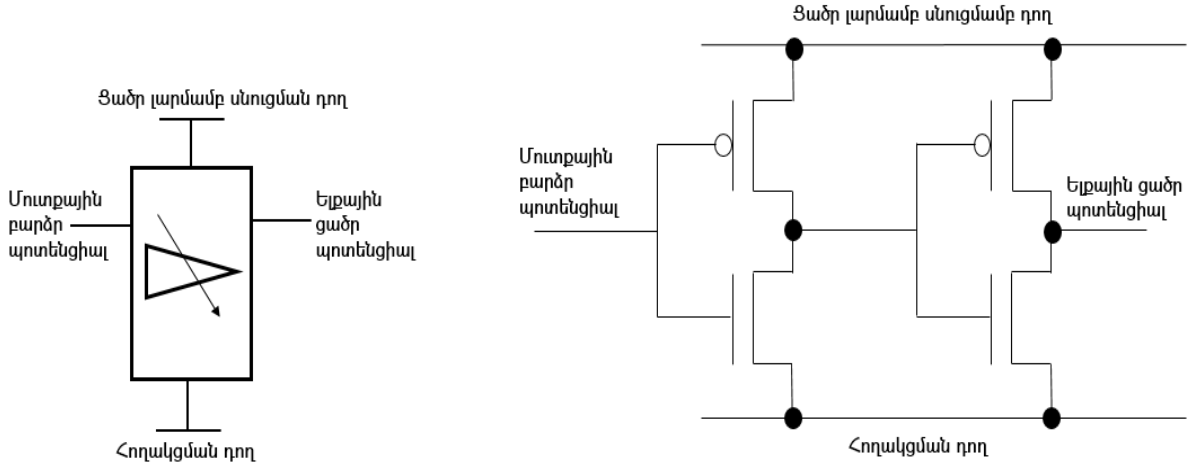
ՎՊՎ ռեգիստրների տարբեր կառուցվածքների համեմատական արդյունքները

Պարամետրեր	ՎՊՎ ռեգիստրի՝ մեկ արխիվացնող մուտքով տարբերակ	ՎՊՎ ռեգիստրի՝ երկու Պահպանում/Վերականգնում մուտքերով տարբերակ
Հապաղումներ (ClkToQ) *(պվ)*	39,54	38,17
Ընդհանուր հզորություն *(մկՎտ)*	4,16	4,37
Կորստի հզորությունը ցածր էներգասպառման ռեժիմում *(նՎտ)*	846,67	927,61
Կորստի հզորությունը ակտիվ աշխատանքային ռեժիմում *(նՎտ)*	50,39	52,46
Մակերես *(մկմ²)*	59	98

Լարման մակարդակի փոխակերպիչներ (ԼՄՓ): Ազդանշանը տարբեր հզորության կզոյակների միջև փոխանցելու ժամանակ անհրաժեշտ են ԼՄՓ-ներ: ԼՄՓ-երն ուղղված են մի ուղղությամբ՝ կամ բարձր պոտենցիալից դեպի ցածր, կամ հակառակը՝ ցածրից դեպի բարձր: Այն դեպքում, երբ բազմասնուցում նախագծման ժամանակ լարման արժեքները մնում են անփոփոխ, ԼՄՓ-ների միաուղղված լինելը մեծ բարդություններ չի առաջացնում [73]:

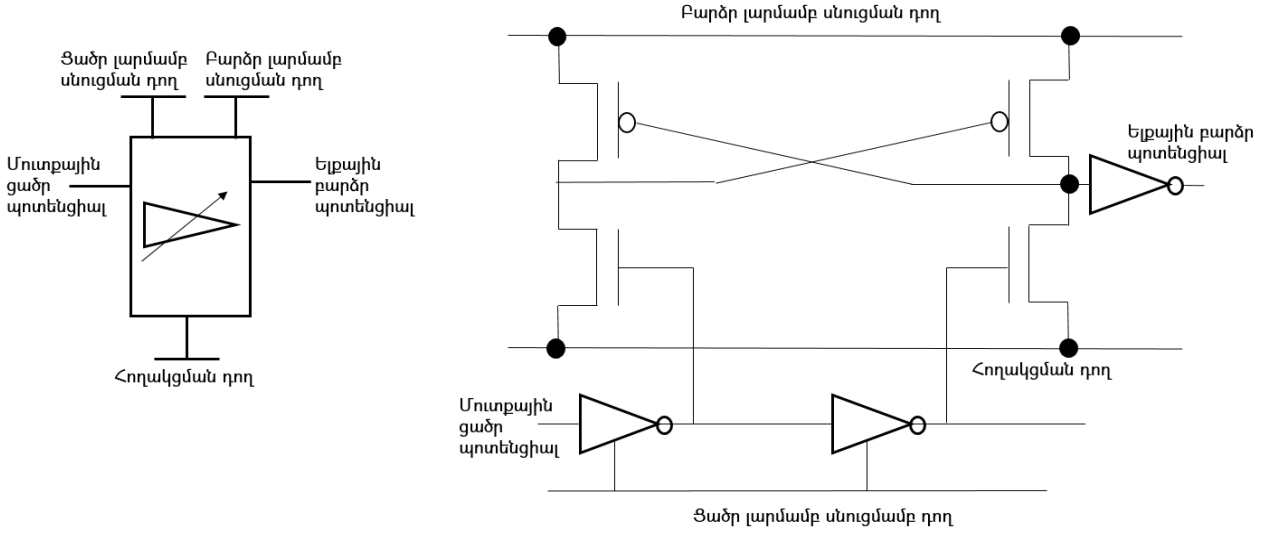
Բարձրից ցածր ԼՄՓ-ներն ունեն պարզ կառուցվածք և հիմնականում զերծ են բացասական երևույթներից: Պարզագույն բարձրից ցածր ԼՄՓ–ները կազմված են երկու իրար հաջորդաբար միացված շրջիչներից: Ժամանակային պարամետրերի տեսանկյունից նույնպես չեն առաջանում մեծ խնդիրներ, քանի որ տվյալ ԼՄՓ-ները դիտարկվում են որպես տրամաբանական կրկնիչներ (նկ. 1.27) [73]: Ցածր լարման

տիրույթից ազդանշանի անցումը դեպի բարձր լարման տիրույթ ավելի բարդ հարց է: Ցածր պոտենցիալով մուտքային ազդանշանը մեծացնում է աճի և նվազման ժամանակները ԼՄՓ-ի մուտքերում:



Նկ. 1.27. Բարձրից ցածր լարման մակարդակի փոխակերպիչը

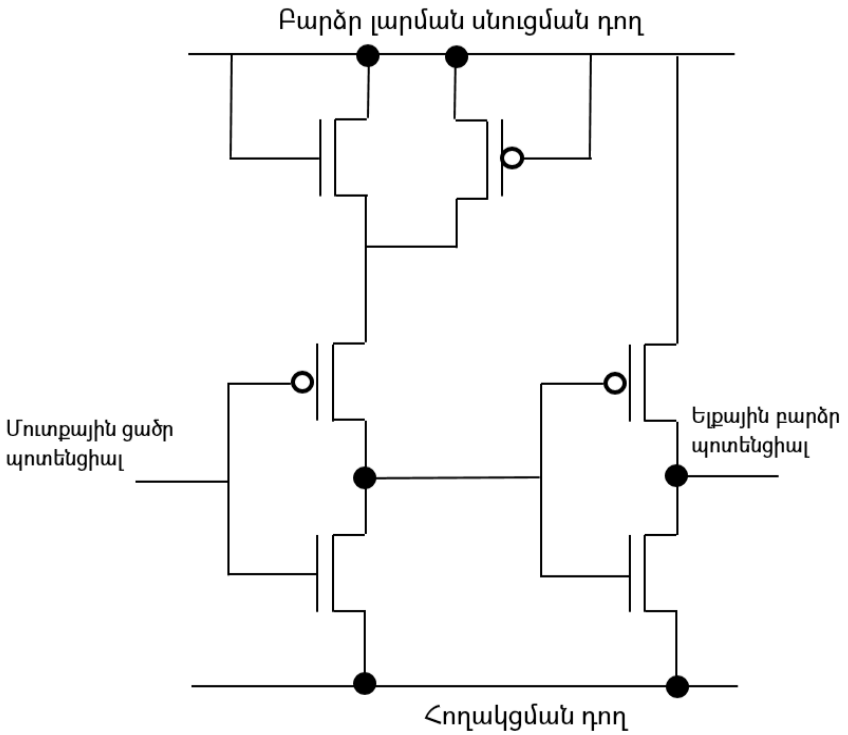
Այդ երևույթի արդյունքում մեծանում է փոխանջատման հոսանքը, և փոքրանում է աղմկակայունությունը [74] (նկ. 1.28): Նմանատիպ ԼՄՓ–ներն ունեն երկու սնման լարումներ և մեկ հողակցման դող: Այս կառուցվածքի հիմնական բաղադրիչը խաչվող միացմամբ շրջիչների զույգն է, որն ապահովում է լարման մակարդակի փոփոխություն: Բարձրից ցածր այդ ԼՄՓ–ներն ունեն բավականին մեծ հապաղում՝ ի տարբերություն ցածրից բարձր ԼՄՓ-ի:



Նկ. 1.28. Ցածրից բարձր լարման մակարդակի դասական փոխակերպիչը

Առավել արդիական են դառնում տվյալ հապաղումները համակարգային մակարդակում, օրինակ՝ պրոցեսորային ԻՍ-երում, երբ կատարվում է անցում կենտրոնական մշակման հանգույցից դեպի օպերատիվ հիշողության հանգույց, որոնք գտնվում են տարբեր լարման տիրույթներում [74]:

Գոյություն ունեն ԼՄՓ-ներ, որոնք կատարում են ցածրից բարձր լարման մակարդակի փոփոխություն՝ օգտագործելով միայն մեկ սնման լարում՝ ընձեռելով ավելի լայն հնարավորություններ ֆիզիկական նախագծման ժամանակ: Տվյալ կառուցվածքում բարձր սնման լարմանը միանում են N-ՄՕԿ և P-ՄՕԿ տրանզիստորային հանգույցը, որի արդյունքում առաջնային շրջիչի վրա ձևավորվում է ցածր սնման լարում [75]: Մուտքային լարման պայմանականորեն մեկ (ձևավորված ցածր պոտենցիալով $V_{\text{ցածր/սնուցում}}$) արժեքի պարագայում առաջնային շրջիչի վրա ձևավորվում է $V_{\text{բարձր/սնուցում}} - V_{\text{շեմային}}$ սնուցման լարման անկում, որն էլ փակում է առաջնային շրջիչի P-ՄՕԿ տրանզիստորը՝ ելքում ձևավորելով տրամաբանական զրո մակարդակ: Այնուհետև զրո մուտքային ազդանշանը բացում է երկրորդ շրջիչի P-ՄՕԿ տրանզիստորը՝ ապահովելով բարձր լարման արժեքը ելքում: Տվյալ սխեմայի հիմնական առավելությունը հատկապես նկատելի է ֆիզիկական նախագծման դեպքում՝ ավելի պարզ տեղաբաշխման հնարավորությամբ [75,76]: Այնուամենայնիվ, մեկ սնուցմամբ սխեման ունի նաև թերություններ, որոնք հիմնականում պայմանավորված են ժամանակային հապաղումներով, որի պատճառներն են սնուցման դողում տրանզիստորների դիոդային միացման կիրառումը և մուտքային լարման սահմանափակ տարբերությունը սնուցման լարման նկատմամբ, մասնավորապես՝ դրանց տարբերությունը չի կարող գերազանցել կիրառված N-ՄՕԿ տրանզիստորի շեմային լարումը (նկ. 1.29):



Նկ. 1.29. Մեկ սնուցմամբ ցածրից բարձր լարման մակարդակի փոխակերպիչը

Ստորև բերված է փորձարարական արդյունքների համեմատությունը երկու սնուցմամբ և մեկ սնուցմամբ նախագծված ԼՄՓ-ների միջև (աղ. 1.7): Կատարվել է լարման փոխարկում 0,7 *Վ*–ից *1* Վ-ի, մուտքային ազդանշանի հաճախությունը՝ 500 *ՄՀց*:

Աղյուսակ 1.7

ԼՄՓ-ների տարբեր կառուցվածքների համեմատական արդյունքները

Պարամետրեր	Երկու սնուցմամբ ցածրից բարձր ԼՄՓ (նկ. 1.28)	Մեկ սնուցմամբ ցածրից բարձր ԼՄՓ (նկ. 1.29)
Հապաղումներ *(պվ)*	3,65	4,745
Ընդհանուր հզորություն *(մկՎտ)*	16,9	12,98
Մակերես *(մկմ²)*	9,18	10,3
Մոդելավորման ժամանակ *(նվ)*	200	200

Մեկ սնուցմամբ ԼՄՓ-ի պարագայում ստացվել է ընդհանուր հզորության նվազեցում մոտ 20%, միևնույն ժամանակ հապաղումները մեծացել են մոտ 30%-ով, իսկ ընդհանուր մակերեսը մեծացել է մոտ 12%-ով [76]: Հաշվի առնելով այն փաստը, որ ընդհանուր հզորության նվազեցումը կազմել է ընդամենը 20%, կարելի է եզրակացնել,

որ հզորության նվազարկման տեսանկյունից կա արդյունավետ տարբերակ՝ ոչ մեծ նվազարկման հնարավորությամբ, միաժամանակ ընդունելով, որ մակերեսը և ժամանակային հապաղումները ենթարկվել են մեծ փոփոխությունների:

Այսպիսով, ամփոփելով տվյալ բաժնում տեղ գտած հիմնավորումները, կարելի է եզրակացնել, որ ԼՄՓ-ներում քննարկված սխեմատեխնիկական կառուցվածքները ամբողջությամբ չեն բավարարում հզորության և արագագործության արդի պահանջները, և անհրաժեշտություն կա՝ լավարկելու այդ կառուցվածքները: Քննարկված ՎՊՎ ռեգիստրներում հիմնական հարցը վիճակի արագ վերականգնումն էր ռեժիմի փոփոխությանը զուգահեռ: Բավականին մեծ խոչընդոտ են նաև մակերեսի չափազանց մեծ արժեքները, առավել ևս պրոցեսորային համակարգերում կիրառման դեպքում, քանի որ դրանցում ընդգրկվում են հազարավոր անգամ տասնյակ հազարավոր ռեգիստրներ:

1.3. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման առաջարկվող սկզբունքները

Նախորդող շարադրանքից հետևում է, որ առկա մեթոդները, որոնք նպատակաուղղված են պրոցեսորային ԻՍ-երի էներգասպառման նվազարկմանը, ունեն թերություններ՝ ԻՍ-ի ժամանակային հապաղումների և մակերեսի մեծացման տեսանկյունից: Առաջանում է նաև համակարգի կայունության հարց պրոցեսորային ԻՍ-ում ռեժիմների փոփոխության ժամանակ: Մասնավորապես.

1. Ցածր էներգասպառման մեթոդների ներդրման արդյունքում պրոցեսորային ԻՍ-ն բավականին բարդանում է, ինչը հանգեցնում է սխալների թե՛ ֆիզիկական և թե՛ տրամաբանական նախագծման փուլերում:
2. Տարբեր մեթոդների կիրառման արդյունքում ավելացված լրացուցիչ սխեմաներն ունենում են ուղիղ ազդեցություն պրոցեսորային ԻՍ-ի ժամանակային հապաղումների, մակերեսի և հզորության վրա, մասնավորապես՝ մակարդակի փոխակերպիչների միջոցով:
3. Հզորության շրջափակման մեթոդի կիրառմամբ նախագծման պարագայում պրոցեսորային ԻՍ-երում առաջանում են խնդիրներ տարբեր

աշխատանքային ռեժիմների միջև փոփոխություններ կատարելու ընթացքում, մասնավորապես՝ ընտրված հանգույցներում սնուցման անջատումից հետո վիճակի պահպանման և վերականգման առումով, որոնք սահմանափակում են մեթոդի ամբողջական արդյունավետ կիրառումը:

4. Բազմասնուցում նախագծման՝ լարման և հաճախության դինամիկ մասշտաբավորման ժամանակ հաշվի չեն առնվում բոլոր պարամետրերը, որոնք կարող են ազդեցություն ունենալ հզորության նվազարկման վրա:
5. Դիտարկված բոլոր մեթոդները լուծում են մեկ խնդիր՝ հզորության կամ դինամիկ, կամ ստատիկ բաղադրիչների նվազարկումը, բայց, միաժամանակ, պրոցեսորային ԻՍ-երում մեծ է թե՛ ստատիկ և թե՛ դինամիկ բաղադրիչների ազդեցությունը ընդհանուր հզորության ձևավորման գործում:

Առաջարկվում են պրոցեսորային ԻՍ-երի էներգասպառման նվազարկման նոր սկզբունքներ, որոնց հիմնական նպատակն է ունենալ էներգասպառման զգալի նվազարկում՝ միաժամանակ պահպանելով հնարավորինս նվազագույն ազդեցություն կիսահաղորդչային բյուրեղում ԻՍ-ի զբաղեցրած մակերեսի և ժամանակային հապաղումների վրա:

Առաջին՝ պրոցեսորային ԻՍ-երում էներգասպառման նվազարկման մեթոդի առաջարկված տարբերակը հնարավորություն է տալիս ապահովել բավարար չափով հզորության նվազարկում, մասնավորապես՝ ստատիկ և դինամիկ բաղադրիչների միաժամանակյա ներգրավմամբ: Մեթոդի հիմքում ընկած է էներգասպառման նվազարկման տարբեր մեթոդների համադրությունը՝ հատուկ մշակված կառավարման հանգույցի հետ միասին:

Երկրորդ՝ առաջարկվել է ՀՇՄ-ի նոր մոտեցում: Մեթոդի հիմքում ընկած են վիճակի պահպանման վերականգնման գործառույթը և դրա իրականացումն ապահովող միջոցների տեղափոխման գաղափարը՝ հանգույցի ներսից դեպի դրա եզրագծեր: Մեթոդի կիրառումը ամբողջական և արդյունավետ դարձնելու համար առաջարկվել են նոր կառուցվածքով լարման մակարդակի փոխակերպիչներ, որոնք ավելի էներգախնայող են և ունեն վիճակի պահպանման հնարավորություն, միաժամանակ՝ ապահովում են պահանջված արագագործությունը:

Երրորդ՝ առաջարկվել է դինամիկ լարման և հզորության մասշտաբավորման նոր եղանակ՝ հիմնված ջերմաստիճանային կախվածության ներդրման վրա: Առաջարկված մեթոդի իրականացման համար մշակվել է հատուկ ջերմաստիճանի մասշտաբավորման հանգույց:

Եզրակացություններ

1. Ժամանակակից պրոցեսորային ինտեգրալ սխեմաների էներգասպառումը հասել է անթույլատրելի արժեքների, և դրա նվազարկումը ներկայումս ծայրահեղ անհրաժեշտություն է:
2. Չնայած ԿՄՕԿ պրոցեսորային ինտեգրալ սխեմաների էներգասպառման առկա նվազարկման միջոցները ինչ-որ չափով փոքրացնում են սնման աղբյուրից ծախսվող հոսանքները, այնուամենայնիվ, դրանք չեն բավարարում ժամանակակից նախագծման գործնական պահանջները, ինչը թելադրում է նոր եղանակների և մեթոդների մշակման անհրաժեշտությունը:
3. Առաջարկվել են պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկման սկզբունքներ, որոնց հիման վրա նախագծումն ապահովում է սնման աղբյուրից ծախսվող հոսանքների բավարար չափով փոքրացում:

ԳԼՈՒԽ 2. ՊՐՈՑԵՍՈՐԱՅԻՆ ԻՆՏԵԳՐԱԼ ՍԽԵՄԱՆԵՐԻ ԷՆԵՐԳԱՍՊԱՌՄԱՆ ՆՎԱԶԵՑՄԱՆ ԱՌԱՋԱՐԿՎՈՂ ՄԻՋՈՑՆԵՐԸ

2.1. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման խնդրի դրվածքը

Աշխատանքի նպատակն է հետազոտել և մշակել պրոցեսորային ԻՍ-երի էներգասպառման նվազեցման նոր մեթոդների և միջոցների համախումբ, որոնց կիրառման արդյունքում հնարավոր կլինի զգալիորեն նվազարկել պրոցեսորային ԻՍ-ի էներգասպառումը՝ միաժամանակ ապահովելով մակերեսի և արագագործության վրա ոչ էական ազդեցություն: Ներդրված մեթոդները և միջոցները պետք է անկախ լինեն ԿՄՕԿ տեխնոլոգիական գործընթացից, միաժամանակ համադրելի լինեն դրանց հետ և հնարավորինս համապիտանի լինեն տարբեր պրոցեսորային ճարտարապետությունների տեսանկյունից:

Հաշվի առնելով արդեն իսկ կիրառվող պրոցեսորային ԻՍ-երի էներգասպառման նվազեցման գոյություն ունեցող մեթոդների թերությունները՝ մշակվել է դրված խնդրի լուծման նոր մեթոդների և միջոցների համախումբ, մասնավորապես՝

- պրոցեսորային ԻՍ-երում միաժամանակյա ստատիկ և դինամիկ հզորությունների նվազարկման նոր մեթոդի սկզբունքներ, որոնք հիմնված են տարբեր մեթոդների համադրման և համատեղման լուծումների վրա,
- ՀՇՄ-ի նոր մոտեցում՝ ցածր էներգասպառման ռեժիմում ընտրված հանգույցում լարման ամբողջովին անջատման սկզբունքի հիման վրա,
- նոր մոտեցմամբ՝ ցածրից բարձր արագագործ լարման մակարդակի փոխակերպիչի սխեմատեխնիկական կառուցվածք,
- նոր մոտեցմամբ՝ վիճակը պահպանող և վերականգնող լարման մակարդակի փոխակերպիչի սխեմատեխնիկական կառուցվածք,
- լարման և հաճախության դինամիկ մասշտաբավորման մեթոդի նոր եղանակ, որի դեպքում ջերմաստիճանի կախվածությունը հաշվի առնելով՝ ստացվել է էներգասպառման նվազեցում:

Առաջարկված մեթոդները և միջոցները ներդրվել են տարբեր կառուցվածքով պրոցեսորային ԻՍ-երում: Ստացված փորձարարական արդյունքները համեմատվել են պրոցեսորային ԻՍ-երի արդեն իսկ գոյություն ունեցող էներգասպառման նվազարկման մեթոդների և ստանդարտ նախագծերի հետ: Գնահատելու համար ներդրված մեթոդների ամբողջական արդյունավետությունը՝ կատարվել է անհրաժեշտ պարամետրերի համեմատական համադրում:

2.2. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման հնարավոր նվազեցման հետազոտումը

Պրոցեսորային ԻՍ-երի էներգասպառման նվազարկման համար սովորաբար թվային միջուկը բաժանում են տարբեր տիրույթների՝ փորձելով բալանսավորել առանձնացված տիրույթները՝ ըստ դրանց արագագործության, օպերատիվ ծանրաբեռնվածության և կայունության: Առանձնացված տիրույթները միմյանցից կարող են տարբերվել ինչպես սնուցման լարման, այնպես էլ տակտային ազդանշանի

հաճախությամբ: Տվյալ պարագայում հնարավոր է լինում կատարել էներգասպառման օպտիմալ բաժանում պրոցեսորային ԻՍ-ի տարբեր ենթահանգույցների միջև: Միաժամանակ, պրոցեսորային ԻՍ-երում նմանատիպ բաժանումները շատ հաճախ առաջացնում են ժամանակային մեծ հապաղումներ, մակերեսի մեծացում և այլն: Նկատի ունենալով արդի ԿՄՕԿ տեխնոլոգիական գործընթացներով նախագծված տրանզիստորների չափերի նվազեցումը և դրանց աճող զգայնությունը՝ կորստի և դինամիկ հզորությունների հանդեպ, առավել հաճախ է առաջանում դրանց միասնական նվազեցման հնարավոր եղանակների հայտնաբերման անհրաժեշտություն: Ժամանակակից պրոցեսորային ԻՍ-երը, որոնք օգտագործվում են նորագույն սարքավորումներում և աշխատում են միայն դյուրակիր մարտկոցների հիման վրա, նախագծվում և արտադրվում են առաջադեմ ԿՄՕԿ տեխնոլոգիական գործընթացներով, ինչպիսիք են 16*նմ*/10*նմ*/7*նմ*–ը: Այսպիսով, ԿՄՕԿ տեխնոլոգիական գործընթացը և դրա շեղումները, տակտային գերբարձր հաճախությունը, ինտեգրված տարրերի քանակը և խտության աստիճանը պրոցեսորային ԻՍ-ը դարձնում են դինամիկ և ստատիկ հզորությունների միաժամանակյա նվազեցման կարևորագույն հանգույց:

2.3. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման առաջարկվող մեթոդների համադրման սկզբունքը

1.2–ում առաջարկված մեթոդները հիմնականում անդրադառնում են առանձին-առանձին միայն ստատիկ կամ միայն դինամիկ հզորության նվազարկմանը՝ միաժամանակ ունենալով մեծ ազդեցություն ժամանակային պարամետրերի և ԻՍ-ի մակերեսի վրա: Տվյալ խնդիրը լուծելու նպատակով առաջարկվել է նոր մեթոդ, որն ապահովում է հզորության երկու բաղադրիչների իջեցում՝ միաժամանակ առաջացնելով նվազագույն փոփոխություններ մակերեսի և ժամանակային պարամետրերի վրա: Մեթոդների սահուն ներդրումն ապահովելու համար նախագծվել են անհրաժեշտ սխեմատեխնիկական հանգույցներ, որոնք պարունակում են նոր մոտեցումներ առկա խնդիրների լուծման համար:

Մեթոդի ուսումնասիրումը կատարվել է տարբեր համակարգերի նկատմամբ դրա կիրառումը և արդյունավետությունը գնահատելու համար, մասնավորապես՝ թվային

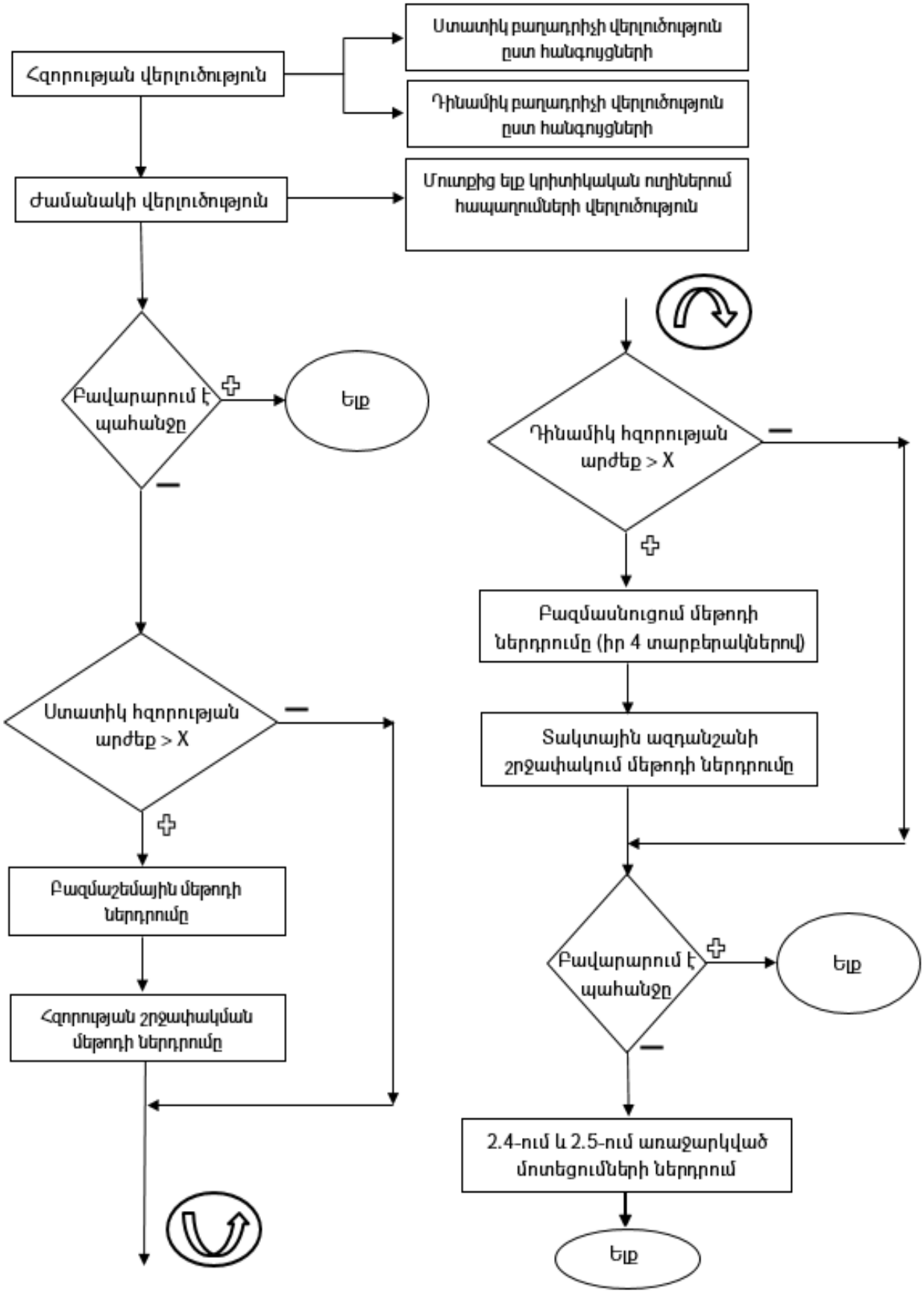
սինքրոն համակարգերում, ինչպիսիք են RISC ճարտարապետության պրոցեսորային ԻՍ-երը, այնպես էլ համապիտանի հաջորդական դողի (ՀՀԴ) թվային միջուկը: Առաջարկված մոտեցման հիմնական նպատակն է ստատիկ և դինամիկ հզորությունների միաժամանակյա նվազարկումը, որը հիմնված է ցածր էներգասպառման տարբեր մեթոդների համադրության և ներդրման վրա: Միասնական կիրառման հիմնական առավելությունն այն է, որ հնարավոր է տարբեր ենթաբլոկերում կիրառել հզորության նվազարկման տարբեր մոտեցումներ՝ կախված տվյալ հանգույցի պարամետրերից: Տվյալ մոտեցման հիմնական հիմնահարցերից են կիրառելի մեթոդների որոշումը և դրանց միասնական ինտեգրման ապահովումը: Ատենախոսությունում, որպես պրոցեսորային ԻՍ-եր, հիմք են վերցվել միայն RISC ճարտարապետության տարբեր պրոցեսորային համակարգերը:

Առաջարկվող մոտեցման բլոկ-դիագրամն ունի հետևյալ կառուցվածքը. համակարգում կատարելով էներգասպառման վերլուծություն՝ հնարավոր է բացահայտել դրա կազմում գտնվող առանձին ստատիկ/դինամիկ բաղադրիչների արժեքները և դրանց կշիռը ընդհանուր հզորության մեջ: Համակարգում կատարելով ժամանակային վերլուծություն՝ հնարավոր է պարզել մուտքից ելք կրիտիկական ուղիները և բաշխել դրանք ըստ կարևորության աստիճանի:

Հաջորդիվ կատարվում է ընդհանուր էներգասպառման արդյունավետության գնահատում և համեմատում՝ ստուգելու համար համապատասխանությունը տրված առաջադրանքին: Հաջորդ քայլով կատարվում է արդեն իսկ առանձնացված ստատիկ հզորության ստացված տվյալների համեմատում: Այնուհետև, հաշվի առնելով ժամանակի և հզորության վերլուծության արդյունքները՝ ըստ հանգույցների, կատարվում է նախապես ընտրված ստատիկ հզորության նվազարկման՝ բազմաշեմային և հզորության շրջափակում մեթոդների ներդրումը: Հաջորդ քայլով կատարվում է դինամիկ հզորության վերլուծություն և ստացված արդյունքների համեմատում՝ ստուգելու համար տեխնիկական առաջադրանքին դրանց բավարար լինելը: Դրական արդյունքի պարագայում համակարգը դուրս է գալիս ակտիվ գործողություն կատարելու վիճակից և համապատասխան ազդանշան է ուղարկում պրոցեսորի կենտրոնական մշակման հանգույցին: Հակառակ դեպքում՝ կիրառվում է նախապես ընտրված դինամիկ հզորության նվազարկման մեթոդների ներդրում՝ բազմասնուցում և տակտային ազդանշանի շրջափակում: Այնուհետև կատարվում է

ստացված ընդհանուր էներգասպառման համեմատական արդյունքների ամփոփում: Արդյունքները չբավարարելու դեպքում ներդրվում են 2.4-ում և 2.5-ում առաջարկված մոտեցումները: Հակառակ պարագայում՝ մշակված համակարգը վերջացնում է ակտիվ գործողությունները և համապատասխան ազդանշան ուղարկում պրոցեսորի կենտրոնական մշակման հանգույցին: Համակարգային բլոկ-սխեման իր հանգույցներով և հետադարձ կապերով ամբողջովին բնութագրում է առաջարկված մեթոդի աշխատանքի սկզբունքը (նկ. 2.1):

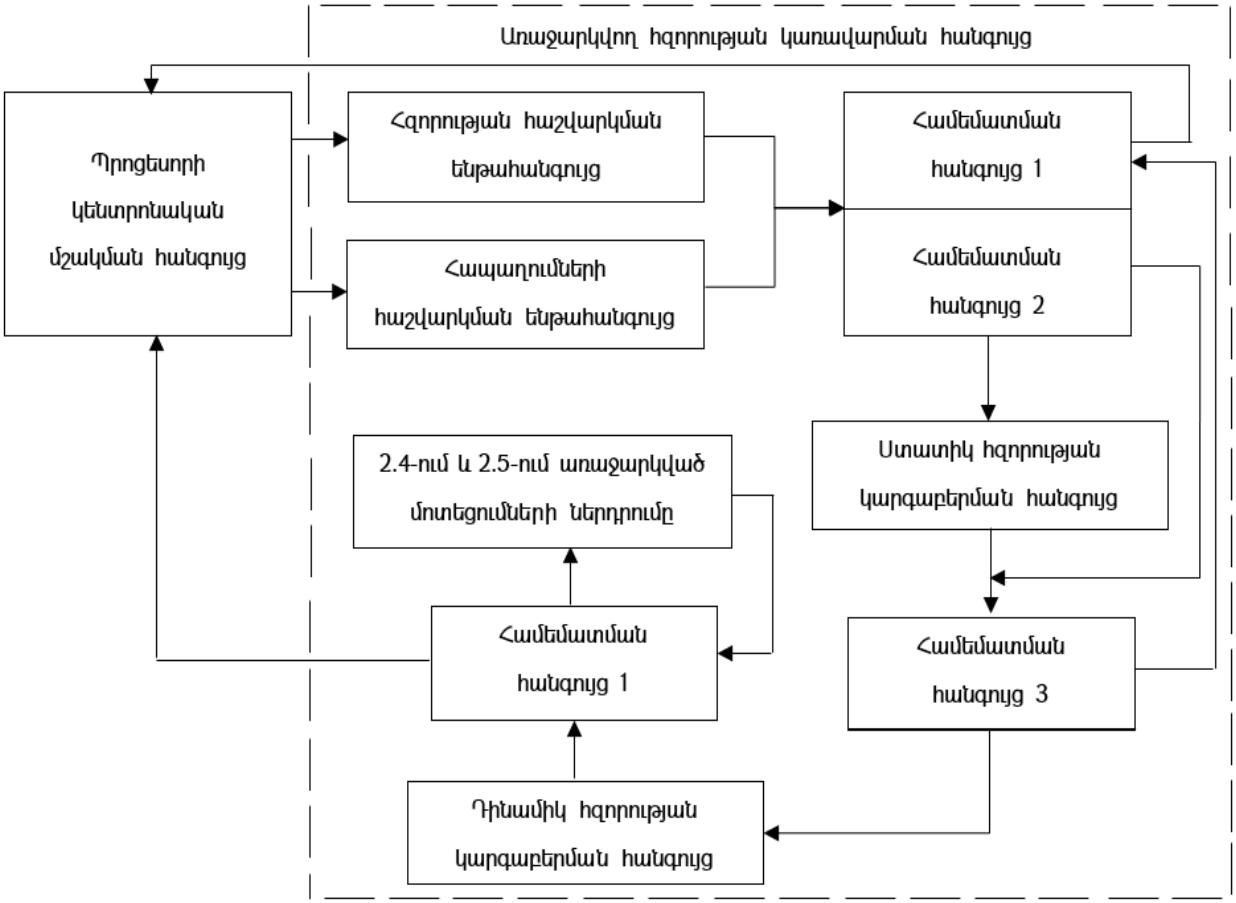
Առաջարկված մեթոդի համար մշակվել են թե՛ սխեմատեխնիկական լուծումներ և թե՛ ծրագրային միջոցներ՝ դրանց ներդրումը համակարգային մակարդակով ապահովելու համար: Ներդրված հանգույցը կառավարվում է արտաքին աղբյուրներից մուտքային ազդանշանների միջոցով՝ հիմք ընդունելով պրոցեսորից ստացված վերլուծության տվյալները: Բլոկ-դիագրամի ցանկացած մակարդակում կա զրոյացման գործողության և դադարի հրամանի ակտիվացման հնարավորություն: Ստատիկ և դինամիկ հզորությունների վերլուծության գործողությունները կատարվում են ծրագրային մակարդակում՝ հատուկ մշակված ծրագրով, որի բնութագրերին անդրադարձ կկատարվի երրորդ գլխում:



Նկ. 2.1. Առաջարկվող էներգասպառման նվազարկման տարբեր մեթոդների համադրման բլոկ-դիագրամը

Այս մեթոդի հիման վրա մշակվել է հզորության կառավարման հանգույցը, որը կազմված է հետևյալ ենթաբլոկերից (նկ. 2.2)՝

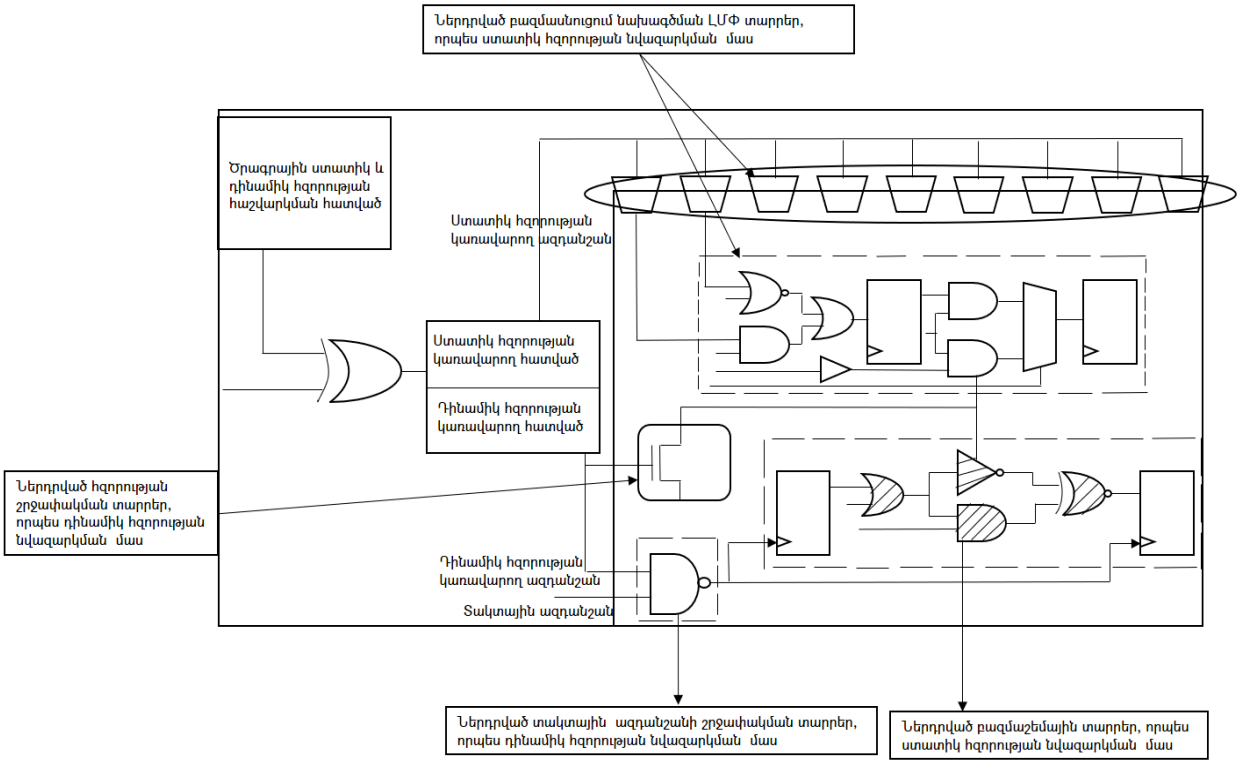
- հզորության հաշվարկման, բաղադրիչների վերլուծության ենթաբլոկ,
- հապաղումների հաշվարկման, ուղիների վերլուծության ենթաբլոկ,
- համեմատման ենթաբլոկ (կազմված 3 մասերից),
- ստատիկ հզորության կարգաբերման ենթաբլոկ,
- դինամիկ հզորության կարգաբերման ենթաբլոկ,
- առաջարկվող մոտեցումների ներդրման ենթաբլոկ:



Նկ. 2.2. Առաջարկված էներգասպառման նվազարկման մոտեցումն իրականացնող համակարգի ընդհանրացված կառուցվածքը

Առաջարկված համակարգի <<հզորության հաշվարկման ենթահանգույցը>> և <<հապաղումների հաշվարկման ենթահանգույցը>> իրականացվել են ծրագրային եղանակով: Որպես համեմատման հանգույցներ օգտագործվել են բացառող ԿԱՄ-ի սխեմաները, որի մի մուտքին գալիս են ծրագրային եղանակով հաշվարկված

հզորության և ժամանակային հապաղումների արժեքները, իսկ մյուս մուտքին՝ արտաքին միջավայրից ստացված համապատասխան սահմանափակումների արժեքները, երկուսն էլ՝ երկուական կոդով: Թե՛ դինամիկ և թե՛ ստատիկ հզորությունների նվազարկման մեթոդների սխեմատեխնիկական իրականացումները ի սկզբանե ինտեգրվում են ընտրված հանգույցում և ակտիվանում հզորության կառավարման հանգույցի համապատասխան ազդանշանի միջոցով: Բազմաշեմային տարրերն ինտեգրվում են ըստ վերլուծության և աշխատանքի ընթացքում փոփոխության չեն ենթարկվում: Հզորության շրջափակման նախագծման ժամանակ օգտագործվել են փոխանջատող N-ՄՕԿ տրանզիստորներ (նկ. 2.3): Երկու դեպքերում էլ դրանք կառավարվում են <<ստատիկ հզորության կառավարող ազդանշան>>-ի միջոցով: Տակտային ազդանշանի շրջափակում և բազմասնուցում մեթոդների ներդրումը կատարվել է սխեմատեխնիկական լուծումների միջոցով, օգտագործվել են ԼՄՓ-ներ և տրամաբանական <<ԵՎ>> տարրեր: Երկուսի կառավարման համար օգտագործվում է <<դինամիկ հզորության կառավարող ազդանշան>>-ը: Հզորության կառավարման ենթահանգույցը բաժանված է ստատիկ և դինամիկ կառավարման առանձնացված հատվածների:



Նկ. 2.3. Առաջարկվող հզորության նվազեցման մոտեցումն իրականացնող համակարգի տրամաբանական մակարդակի կառուցվածքը

Առաջարկված մոտեցման և մշակված հանգույցի օպտիմալ համադրումը, ինչպես նաև առանձնացված մեթոդների ազդեցությունը RISC ճարտարապետության պրոցեսորային ԻՍ-երի վրա պարզելու համար կատարվել են տարբեր հետազոտություններ, մասնավորապես՝ ORCA, OPEN RISC(ARC) [77], simple RISC [78] պրոցեսորային ԻՍ-երում փորձարկվել են հետևյալ մոտեցումների ներդրման ազդեցությունները՝

- ORCA պրոցեսորային ԻՍ-ում ներդրվել է համատեղված տակտային ազդանշանի շրջափակմամբ նախագծումը և բազմաշեմային նախագծումը [79],
- OPEN RISC պրոցեսորային ԻՍ-ում ներդրվել է համատեղված բազմասնուցում, հզորության շրջափակում, բազմաշեմային նախագծում մեթոդները [80]:

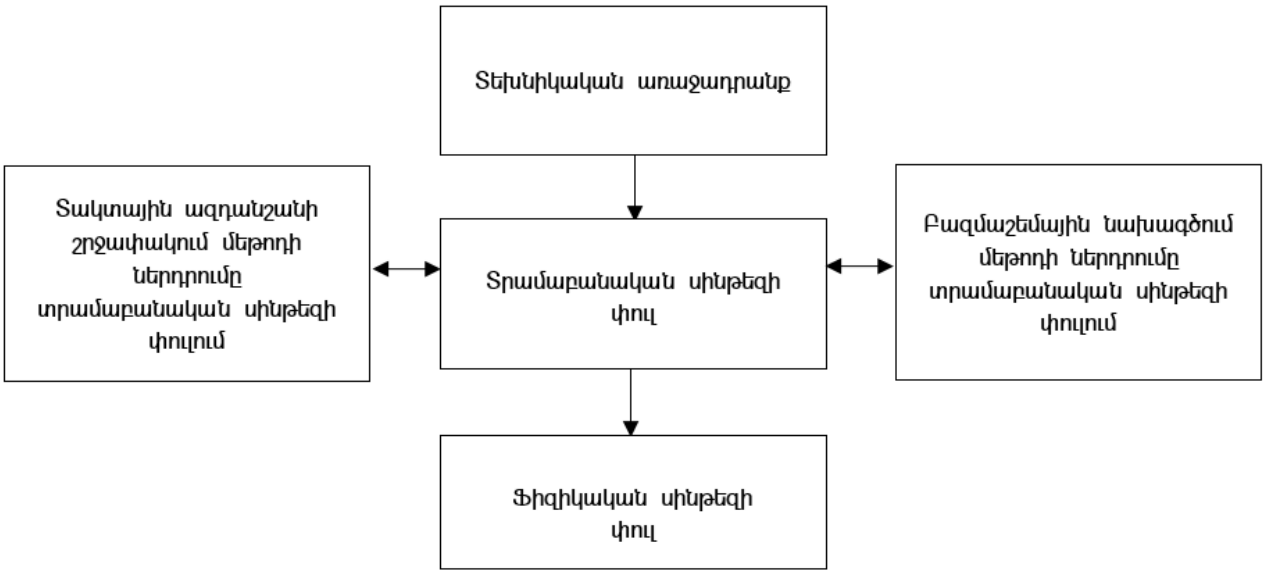
Կատարվել են նաև առաջարկված վերջնական մոտեցման կիրառական արդյունքերի հետազոտում և վերլուծություն՝

- RISC պրոցեսորային ԻՍ-ում ներդրվել է ստատիկ և դինամիկ հզորությունների միաժամանակյա նվազարկման համար նախատեսված միասնական չորս մեթոդների համադրումը՝ բազմասնուցում, հզորության շրջափակում, տակտային ազդանշանի շրջափակում և բազմաշեմային նախագծումը՝ հզորության կառավարման հանգույցի հետ միասին [81]:

Հաջորդիվ հանգամանալից անդրադարձ կկատարվի առաջարկվող մոտեցման ներդրման վերը թվարկված բոլոր եղանակներին առանձնացված կարգով՝ պարզելու համար դրանց արդյունավետությունը:

ORCA պրոցեսորային ԻՍ-ում ներդրվել է տակտային ազդանշանի շրջափակմամբ և բազմաշեմային նախագծումը [79]: Առաջարկվել է RISC ճարտարապետության ORCA պրոցեսորային ԻՍ-ում երկու տարբեր մեթոդների համադրություն՝ բազմաշեմային և տակտային ազդանշանի շրջափակում: Այս մեթոդների համադրումը միաժամանակ կրճատում է թե՛ դինամիկ բաղադրիչը տակտային ազդանշանի շրջափակմամբ և թե՛ ստատիկը բազմաշեմային տարբերակի կիրառմամբ: Կատարվել են երկու տեսակ նախագծեր՝ դասական և առաջարկված տարբերակներով՝ օգտագործելով նույն մուտքային սահմանափակումները, որի նպատակն է ստեղծել հավասար պայմաններ՝ ստացված արդյունքների

համեմատական վերլուծության համար: Նախագծման ընթացքում օգտագործվել են երեք տեսակ ստանդարտ բջիջների գրադարաններ, որոնք պարունակում են ցածր, միջին և բարձր շեմային լարմամբ տարրեր՝ ապահովելու համար բոլոր երեքի միասնական կիրառումը: Համադրված մեթոդների ներդրումը ամբողջովին ինտեգրվել է թվային ավտոմատ նախագծման գործընթացի մեջ և կիրառվել է տրամաբանական սինթեզի փուլում (նկ. 2.4):



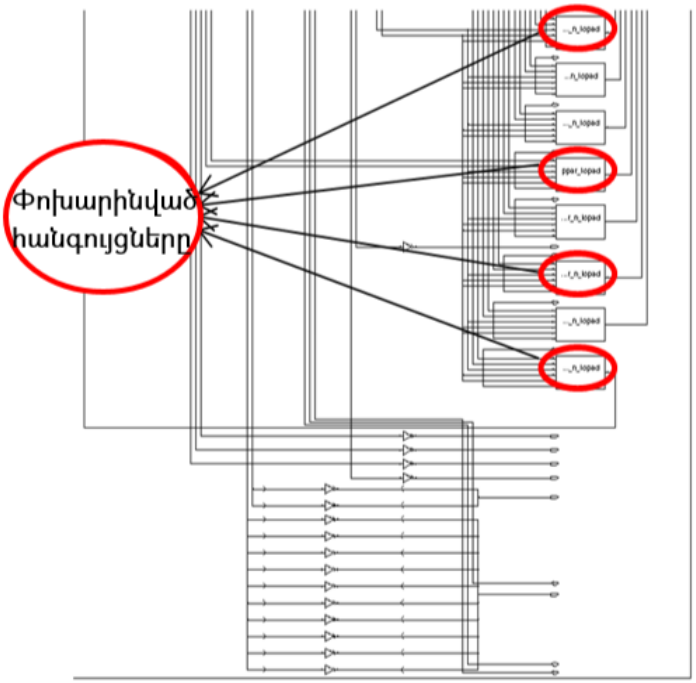
Նկ. 2.4. ORCA պրոցեսորային ԻՍ-ում երկու համադրված մեթոդներով առաջարկվող նախագծման գործընթացը

Կատարվել է ORCA պրոցեսորային ԻՍ-ում կրիտիկական ուղիների վերլուծությունը՝ հաշվի առնելով հզորության և ժամանակային հապաղումների արժեքները: Ընտրվել և առանձնացվել են այն ուղիները, որոնք ունեն մեծ ազդեցություն ընդհանուր հզորության և ժամանակային հապաղումների ձևավորման գործընթացում: Այն ուղիներում, որտեղ ժամանակային հապաղումները մեծ են, և կարևորագույն խնդիրը դրանց կարգավորումն է, օգտագործվել են ցածր շեմային լարմամբ տարրեր՝ ապահովելով հզորության մեծացում: Այն ուղիներում, որտեղ մեծ է հզորության արժեքը, օգտագործվել են առավելագույն քանակով բարձր շեմային լարմամբ տարրեր: Այս տարբերակի կիրառումը սահմանափակ է, քանի որ այն ուղղակի ազդեցություն է ունենում ժամանակային հապաղումների վրա և կարող է

ընդունելի լինել այնքան ժամանակ, որքան հնարավոր կլինի խուսափել ֆունկցիոնալ սխալներից:

Տակտային ազդանշանի շրջափակումը ներդնելու համար վերլուծվել են տակտային ազդանշանի ծառը և դրա մաս կազմող տրամաբանական կրկնիչները: Ընտրվել են այն հանգույցները, որտեղ հնարավոր է կատարել ազդանշանի շրջափակում՝ տրամաբանական սինթեզի փուլում:

Մեթոդների (բազմաշեմային և տակտային ազդանշանի շրջափակում) ներդրման արդյունքում փոփոխությունների ենթարկված հանգույցները առանձնացված են կարմիր շրջանակով (նկ. 2.5):



Նկ. 2.5. ORCA պրոցեսորում բազմաշեմային և տակտային ազդանշանի շրջափակում մեթոդների ներդրման արդյունքում փոփոխությունների ենթարկված հանգույցները

Երկու մեթոդների համադրումը տեղի է ունեցել ըստ հզորության և ժամանակի վերլուծությունների այն հանգույցներում, որոնցում արագագործությունը կրիտիկական է, և հնարավոր չէ կիրառել ամբողջությամբ բազմաշեմային մեթոդը, ուստի ներդրվել է տակտային ազդանշանի շրջափակումը: Հակառակ դեպքում, երբ հնարավոր չէ կիրառել տակտային ազդանշանի շրջափակումը, ներդրվել են բարձր շեմային լարմամբ տարրեր՝ փորձելով օգտագործել փոխլրացնելու սկզբունքը, առավելագույն արդյունավետություն ստանալու համար: Ստորև բերված են երկու մեթոդների

միասնական ներդրման փորձարարական արդյունքները՝ համեմատած դասական նախագծի հետ (աղ. 2.1):

Աղյուսակ 2.1

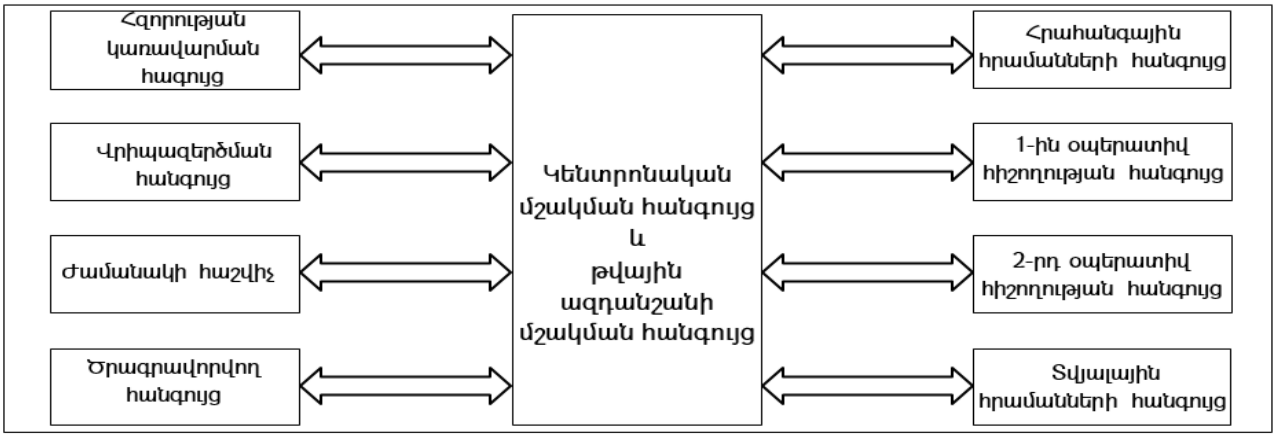
ORCA պրոցեսորի ստացված համեմատական արդյունքները՝ բազմաշեմային և տակտային ազդանշանի շրջափակում մեթոդների միասնական ներդրման դեպքում

Պարամետրերը	Միասնական ներդրված բազմաշեմային և տակտային ազդանշանի շրջափակում մեթոդներով նախագիծ	Ստանդարտ նախագիծ
Տակտային հաճախությունը (*ՄՀց*)	200	200
Մուտքային ազդանշանների անհրաժեշտ ժամանակային պաշարը (*նվ*)	25,69	20,21
Մուտքային ազդանշանների իրական ժամանակային պաշարը (*նվ*)	-22,58	-20,05
Անհրաժեշտ և իրական ժամանակային պաշարների տարբերությունը (*նվ*)	3,11	0,15
Ընդհանուր հզորությունը (*մՎտ*)	78,7828 ~(-11%)	88,52
Մտավոր սեփականության հանգույցի մակերեսը (*մկմ²*)	16340,8	16340,8
Ընդհանուր օգտագործված ստանդարտ բջիջների մակերեսը (*մկմ²*)	689484,51 (+17%)	589303,444805
Ընդհանուր մակերեսը (*մկմ²*)	705825,31 ~(+16,5%)	605644,241192

Այսպիսով, ի համեմատ դասական տարբերակով նախագծի, առաջարկվողի պարագայում ստացվել է 11%-ով ընդհանուր հզորության նվազարկում, որը հիմնականում պայմանավորված է երկու գործոններով՝ դինամիկ և ստատիկ հզորությունների միաժամանակյա նվազեցմամբ: Ընդհանուր մակերեսի մեծացումը կազմել է մոտ 16,5%, որը հիմնականում պայմանավորված է տակտային ազդանշանի շրջափակման մեթոդի արդյունքում լրացուցիչ հանգույցների ավելացմամբ: Արդյունքներից պարզ է դառնում, որ երկու դեպքում էլ հնարավոր է եղել ապահովել կայուն, առանց ֆունկցիոնալ սխալների աշխատանք 200 *ՄՀց* հաճախության պարագայում: Միաժամանակ, անհրաժեշտ և իրական ժամանակային պաշարների տարբերությունը ավելի մեծ է ստացվել առաջարկված տարբերակում, ինչը նշանակում է, որ առկա է կայունության ավելի մեծ աստիճան:

Ամփոփելով տվյալ նոր մոտեցման՝ միավորված բազմաշեմային և տակտային ազդանշանի շրջափակում մեթոդների արդյունքները RISC ճարտարապետության ORCA պրոցեսորային ԻՍ-ի դեպքում, կարելի է եզրակացնել, որ այն բավականին արդյունավետ է հզորության նվազարկման համար այն պարագայում, երբ հարկավոր է ունենալ հնարավորինս փոքր ազդեցություն արագագործության վրա:

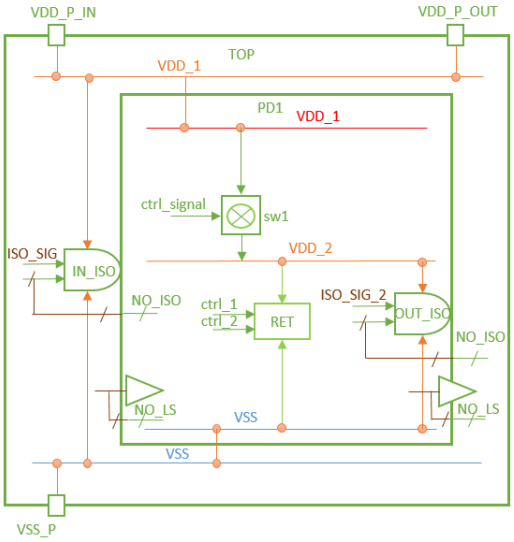
OPEN RISC պրոցեսորային ԻՍ-ում ներդրվել են բազմասնուցում, հզորության շրջափակում և բազմաշեմային նախագծում մեթոդները [80]: Առաջարկված մոտեցման հաջորդ փորձարկումն իրականացվել է OPEN RISC պրոցեսորում, որը իր հիմքում պարունակում է երեք տարբեր մեթոդների համադրումը: OPEN RISC 1200-ը 32 բիթ պրոցեսոր է, որն ունի Harvard [82] ճարտարապետություն, վիրտուալ հիշողության, թվային ազդանշանի մշակման, հզորության կառավարման, վրիպազերծման, կենտրոնական մշակման և հիշողության կառավարման հանգույցներ (նկ. 2.6):



Նկ. 2.6. OPEN RISC 1200 պրոցեսորային ԻՍ-ի կառուցվածքը

OPEN RISC 1200 պրոցեսորային ԻՍ-ը նախատեսված է՝ օգտագործելու ներդրված և ցանցային համակարգերում: Սովորաբար այս ճարտարապետության պրոցեսորները նախագծվում են՝ հատուկ ուշադրություն դարձնելով արագագործությանը, պարզությանը, մասշտաբավորմանը և համապիտանելիությանը: Նախագծման գործընթացում կատարվել են թե՛ տրամաբանական և թե՛ ֆիզիկական սինթեզներ, մշակվել է ՄՀՁ-ը՝ երեք մեթոդների համադրումն ապահովելու համար: Արդյունքում ստացվել է ՄՀՁ դիագրամը, որը բնութագրում է լարման առանձնացված տիրույթները և հատուկ բջիջները, մասնավորապես՝ մեկուսիչ հանգույցները, որոնք բաշխվել են լարման կղզյակի սահմանագծում: Ցածր լարման տիրույթներ են ընտրվել երկուսը՝ վրիպազերծման հանգույցը և հզորության կառավարման հանգույցը, որոնք

համեմատաբար ավելի դանդաղագործ են: Բարձր լարման բոլոր տիրույթներում սնուցման լարման արժեքն ընտրվել է 0,95 Վ, իսկ ցածր լարմանը՝ 0,7 Վ (նկ. 2.7):



Նկ. 2.7. OPEN RISC 1200 պրոցեսորային ԻՍ-ի ՄՀՁ դիագրամը

Այսպիսով, երեք մեթոդների՝ բազմասնուցման, հզորության շրջափակման և բազմաշեմային նախագծման համադրման արդյունքում ստացվել է էներգասպառման նվազարկում մոտ 17%-ով՝ ի համեմատ ստանդարտ նախագծի (աղ. 2.2), միաժամանակ ունենալով մակերեսի մեծացում մոտ 15%-ով:

Աղյուսակ 2.2

OPEN RISC 1200 պրոցեսորային ԻՍ-ի ստանդարտ և առաջարկվող մոտեցմամբ նախագծերի ստացված համեմատական արդյունքները

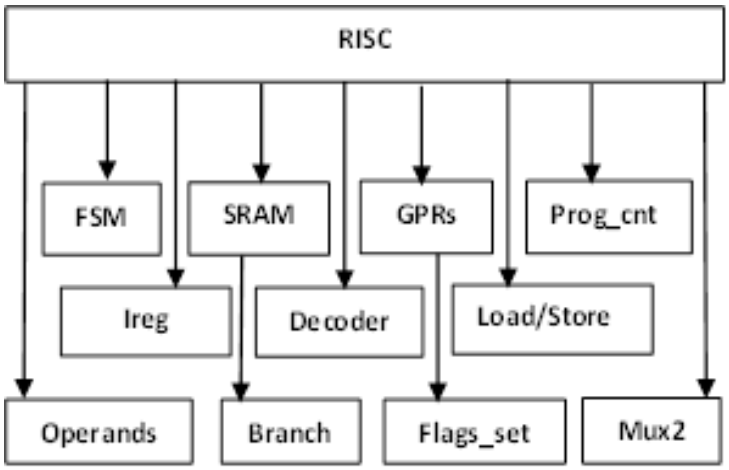
Պարամետրերը	Ստանդարտ նախագծում	Բազմաշեմային, հզորության շրջափակմամբ և բազմասնուցմամբ համադրված մեթոդներով նախագծում
Տակտային հաճախությունը (*ՄՀց*)	200	200
Մուտքային ազդանշանների անհրաժեշտ ժամանակային պաշարը (*նվ*)	20,12	20,12
Մուտքային ազդանշանների իրական ժամանակային պաշարը (*նվ*)	-20,1	-20,03
Անհրաժեշտ և իրական ժամանակային պաշարների տարբերությունը (*նվ*)	0,02	0,09
Դինամիկ հզորությունը (*մՎտ*)	9,701	8,21

Կորստի հզորությունը (*մՎտ*)	5,337	4,271
Ընդհանուր հզորությունը (*մՎտ*)	15,038	12,481 (-17%)
Ընդհանուր մակերեսը (*մկմ²*)	33000,349325	37950,40172 (+15%)

Ժամանակային պարամետրերի տեսանկյունից հնարավոր է եղել ապահովել 200 *ՄՀց* հաճախությունը, այնուամենայնիվ, ունենալով 0,7 *նվ* կորուստ մուտքից ելք հապաղման մեջ:

RISC պրոցեսորային ԻՍ-ում ներդրվել է ստատիկ և դինամիկ հզորությունների միաժամանկյա նվազեցման համար նախատեսված միասնական չորս մեթոդների համադրումը՝ բազմասնուցում, հզորության շրջափակում, տակտային ազդանշանի շրջափակում և բազմաշեմային նախագծումը՝ հզորության կառավարման հանգույցի հետ միասին [81]:

Նոր մոտեցման իրականացման համար մշակվել է simple RISC պրոցեսորային ԻՍ-ն՝ հատուկ միկրոճարտարապետության լուծումներով, որոնք միտված են էներգաարդյունավետությանը և բարձր արագագործությանը: RISC պրոցեսորը մշակվել է՝ հիմքում ունենալով ավանդական <<Վոն Նուեման>> ճարտարապետությունը՝ գրանցում/պահպանում հիշողության հնարավորությամբ (նկ. 2.8ա): Մշակված RISC պրոցեսորում նվազարկված են հրահանգային հրամանների հանգույցն ապահովելու համար բարձր արագագործություն, ցածր էներգասպառում: RISC պրոցեսորը բաղկացած է ութ գլխավոր նշանակության ռեգիստրներից, 16 բիթ լայնությամբ տվյալների և հասցեների հանգույցներից, որոնք կատարում են 16 միաժամանակյա գործողություններ (նկ. 2.8բ):



Գործարկման պայմաններ	Հրամաններ
Z1Z2Z3Z4	
0 0 0 0	NOP
0 0 0 1	MOV _R1,R2,TYPE
0 0 1 0	LOAD
0 0 1 1	STORE
0 1 0 0	ADD_R1,R2,R3
0 1 0 1	SUB_R1,R2,R3
0 1 1 0	SHR_R1, IMM
0 1 1 1	SHL_R1,IMM
1 0 0 0	HALT
1 0 0 1	BRANCH_ INSTRUCTION
1 0 1 0	MUL_R1,R2,R3
1 0 1 1	SQ_RT
1 1 0 0	NOT_R1
1 1 0 1	AND _R1,R2,R3
1 1 1 0	OR_R1,R2,R3
1 1 1 1	XOR_R1,R2,R3

ա) բ)

Նկ. 2.8. ա) Simple RISC պրոցեսորային ԻՍ-ի կառուցվածքային պատկերը,

բ) Simple RISC պրոցեսորի հրամանային հրահանգների կառուցվածքը

Տվյալները պահպանվում են ինտեգրված 256x16 և 64x8 չափայնությամբ SRAM հանգույցներում՝ կարդալու (mem_out) և գրելու (mem_in) առանձնացված մուտքային դողերով: <<Գրել>> գործողությունը սինքրոն է, իսկ <<կարդալ>> գործողությունը՝ ասինքրոն: SRAM–ի մատրիցը կառուցվել է՝ օգտագործելով յոթ տրանզիստորանոց միահանգույց կարգային դողով տարրական բջիջը, իսկ ղեկավարման բլոկում կիրառվել է հետադարձ գրանցման մեթոդը (write-back technique) [83]:

Նախագիծը կատարվել է երկու տարբերակով՝ ստանդարտ և առաջարկվող մոտեցմամբ, որը ներառում է հատուկ նախագծված հզորության կառավարման հանգույց՝ չորս ցածր էներգասպառման մեթոդների ներգրավումն ապահովելու համար: Հզորության և արագագործության վերլուծության արդյունքում հզորության կառավարման հանգույցը պրոցեսորային ԻՍ-ում տարբեր մեթոդները բաշխել է հետևյալ կերպ (աղ. 2.3):

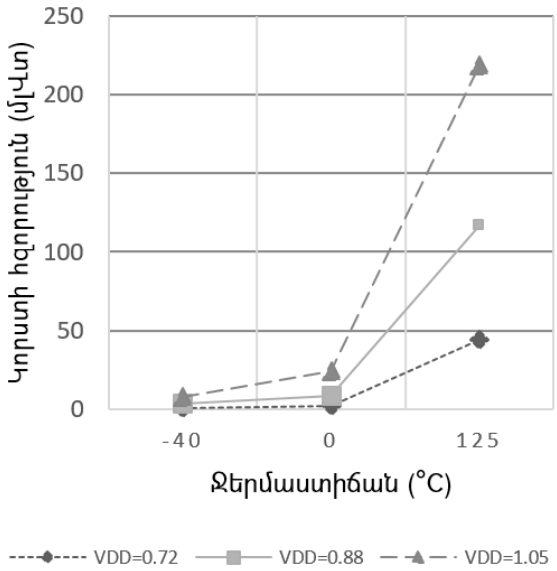
Աղյուսակ 2.3

Կառավարման հանգույցի միջոցով simple RISC պրոցեսորային ԻՍ-ի ցածր էներգասպառմամբ նախագծման եղանակների բաշխումը

Հզորության տիրույթներ	Ցածր էներգասպառման մեթոդներ
GPRs, ALU	Բազմասնուցում նախագծում (Multi voltage)
GPRs, ALU, Program, Mux2	Բազմաշեմային նախագծում (Multi threshold)
GPRs, FSM, Decoder	Տակտային ազդանշանի շրջափակմամբ նախագծում (Clock gating)
RISC Core	Հզորության շրջափակմամբ նախագծում (Power gating)
SRAM	Ցածր էներգասպառման ռեժիմ (Low power modes)

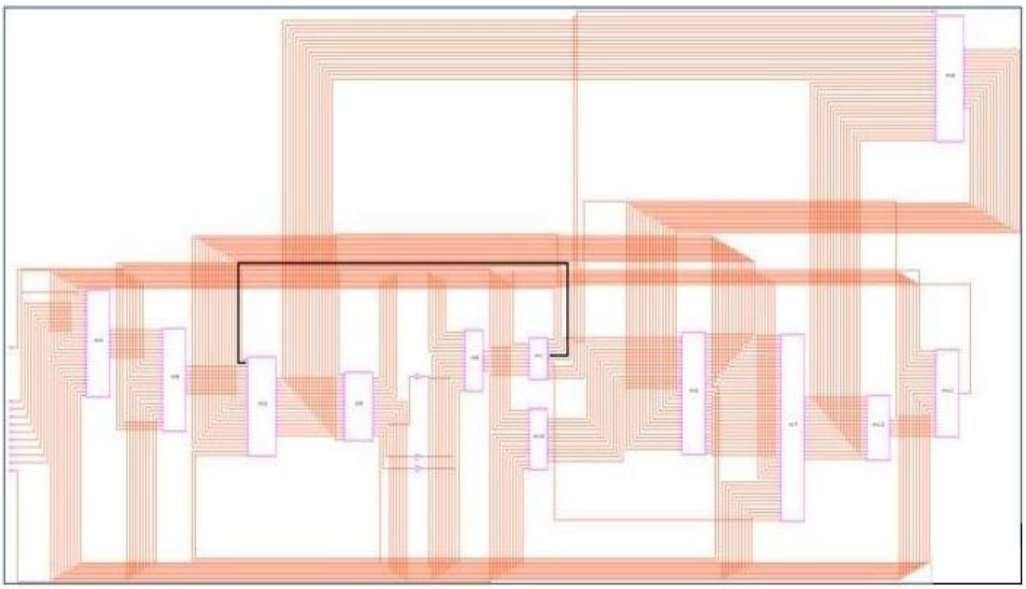
Կատարվել է նաև կորստի հզորության լրացուցիչ հետազոտություն, որում հաշվարկվել է Simple RISC պրոցեսորային ԻՍ-ում կորստի հզորության կախվածությունը ջերմաստիճանի և լարման արժեքների փոփոխություններից (նկ. 2.9), համաձայն որի՝ առավելագույն կորստի հզորությունը ստացվել է բարձր ջերմաստիճանի և լարման դեպքում: Հաշվի առնելով ստացված արդյունքները՝ simple

RISC պրոցեսորային ԻՍ-ն նախագծվել է հենց այդ լարման և ջերմաստիճանին համապատասխանող ստանդարտ բջիջների գրադարանով՝ ապահովելու համար վատագույն տարբերակը հզորության մոդելավորման տեսանկյունից:



Նկ. 2.9. Simple RISC պրոցեսորային ԻՍ-ում ջերմաստիճանի, լարման և կորստի հզորության կախվածությունը

Simple RISC պրոցեսորային ԻՍ-ի նախագծման ժամանակ օգտագործվել են հատուկ նշանակության տարրեր, ինչպիսիքն են՝ բարձր և ցածր շեմային լարմամբ սխեմաները, մակարդակի փոխակերպիչները, վիճակը պահպանող և վերականգնող ռեգիստրները, մեկուսացնող տրանզիստորները, տակտային ազդանշանի և սնման լարման անջատման տարրերը: Նախագծման ընթացքում մշակվել է ՄՀՁ ձևաչափը՝ առաջարկված մեթոդի ինտեգրման համար, որի արդյունքում ստացվել է տրամաբանական մակարդակի սխեմատեխնիկական պատկերը (նկ. 2.10):



Նկ. 2.10. Simple RISC պրոցեսորային ԻՍ-ի տրամաբանական մակարդակի սխեման

Առաջարկված մեթոդի կիրառմամբ simple RISC պրոցեսորային ԻՍ-ի փորձարարական արդյունքները համեմատվել են դասական, արդեն իսկ 2.3-ում քննարկված և գրականությունից ընտրված տարբերակների հետ (աղ. 2.4):

Աղյուսակ 2.4

Simple RISC 1200 պրոցեսորային ԻՍ-ի դասական, գրականությունից ընտրված, 2.3-ում քննարկված տարբերակների և առաջարկվող մոտեցմամբ նախագծերի ստացված համեմատական արդյունքները

Պարամետրերը	ԿՄՕԿ 28 նմ նախագծված RISC պրոցեսոր [84]	Դասական նախագիծ	Առաջարկվող նոր մեթոդով նախագիծ	Բազմաշեմային, բազմասնուցմամբ և ՀՇ մեթոդներով նախագիծ (OPEN RISC/ARC)	Բազմաշեմային և ՏԱՇ մեթոդներով նախագիծ (ORCA)	ՏԱՇ մեթոդով ընտրված գրականությունից
Տակտային հաճախություն *(ՄՀց)*	1200	1900	1900	200	200	1600
Ընդհանուր հզորություն *(մՎտ)*	58	68,55	39,73 (-42%)	12,481 (-17%)	78,7828 (-11%)	47(-34%)
Ընդհանուր մակերես *(մկմ²)*	193724,97 (+11%)	174527,6	180985,12 (+3,7 %)	37950,40172 (+15%)	705825,31 (+16,5%)	N/A

Այսպիսով, համադրված չորս մեթոդներով նախագծման պարագայում ստացվում է 42%-ով էներգասպառման նվազեցում՝ ի համեմատ դասական նախագծման, և 31,5%-ով նվազեցում՝ ի համեմատ գրականությունից ընտրված տարբերակի: Առաջարկված մեթոդի կիրառումը առավելագույն նվազարկումն է ապահովում բոլոր քննարկված տարբերակներից և գրականությունից հայտնի մեթոդներով նախագծերից, միաժամանակ ստացվել է մակերեսի մեծացում 3,7%-ով՝ ի համեմատ դասական նախագծի: Վերլուծելով բոլոր տվյալները՝ կարելի է արձանագրել, որ ստացված կիրառական արդյունքները՝ մասնավորապես էներգասպառման

նվազեցումը, բավականին արդյունավետ է, որով և հիմնավորվում է առաջարկված մոտեցման առավելությունը: Երրորդ, չորրորդ և հինգերորդ սյունակներում բերված տոկոսային հարաբերակցությունները կատարված են դրանց դասական տարբերակների հետ:

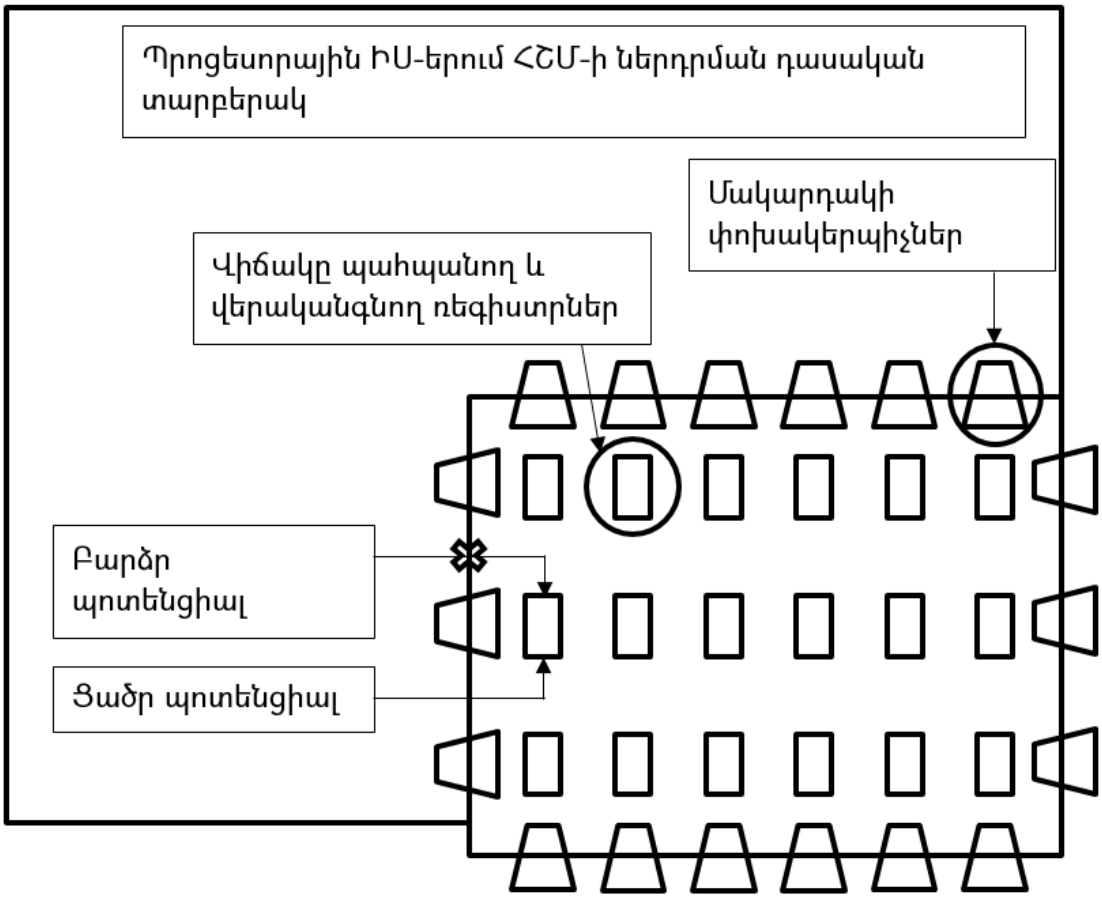
Այսպիսով, ամփոփելով առաջարկվող առաջին մեթոդը և իրագործման համար առաջարկված հզորության կառավարման հանգույցը՝ հիմնված տարբեր մոտեցումների համադրման և օպտիմալացման վրա, կարելի է եզրակացնել, որ այն արդյունավետ է միաժամանակյա հզորության ստատիկ և դինամիկ բաղադրիչների նվազարկման համար: Հնարավոր է եղել ապահովել համակարգի կայունությունը ժամանակային տեսանկյունից: Անհրաժեշտ է նշել նաև առաջարկված մեթոդի թերությունները, որոնք հիմնականում ազդեցություն են ունենում պրոցեսորային ԻՍ-ի ընդհանուր մակերեսի վրա՝ առաջարկված հզորության կառավարման հանգույցի և ներդրված սխեմատեխնիկական լուծումների հաշվին:

2.4. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման՝ առաջարկվող հզորության շրջափակման մեթոդի տարատեսակը

1.2–ում քննարկված պրոցեսորային ԻՍ-երում հզորության շրջափակման մեթոդի հիմնական թերություններն են՝ վիճակի պահպանման վերականգնման ժամանակը, աշխատանքային տարբեր ռեժիմների միջև ժամանակային համադրությունները, մեծ մակերեսը և այն էներգասպառումը, որը հիմնականում պայմանավորված է ավելացված լրացուցիչ հանգույցներով: Տվյալ հիմնահարցերը լուծելու նպատակով առաջարկվել է մեթոդի տարատեսակ, որն ապահովում է հզորության շրջափակման ընթացքում վիճակի պահպանման և վերականգնման հապաղումների կրճատում և ընդհանուր էներգասպառման նվազեցում: Մեթոդի ներդրման հետ մեկտեղ մշակվել են նաև համապատասխան սխեմատեխնիկական հանգույցներ՝ որպես ներդրման բաղկացուցիչ մասեր:

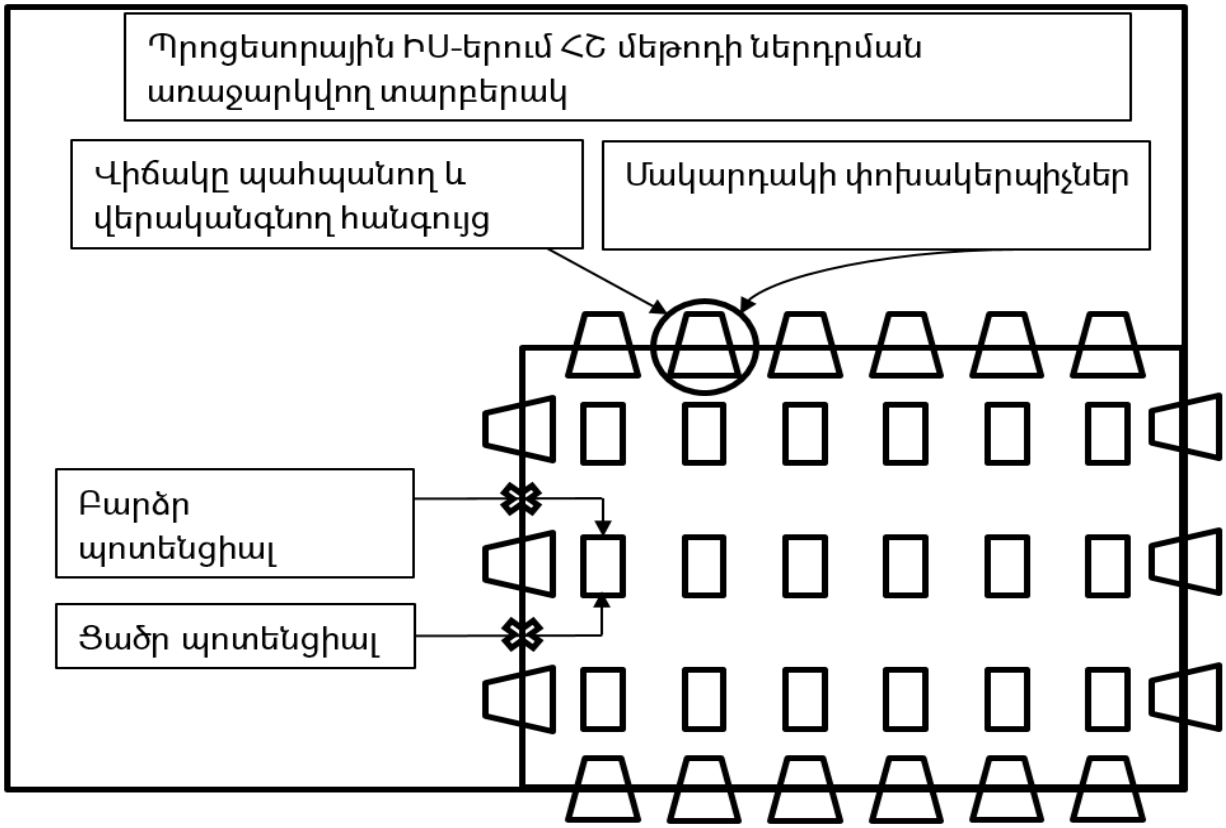
Առաջարկված մեթոդը պարունակում է պրոցեսորային ԻՍ-ի վիճակի պահպանման և վերականգնման նոր մոտեցումներ: Դասական հզորության շրջափակման մեթոդի համաձայն՝ ցածր էներգասպառման ռեժիմում վիճակի

պահպանումը և վերականգնումը կատարվում է ընտրված հանգույցի ներսում հիմնական ռեգիստրների միջոցով, որոնք փոխարինվում են հատուկ նշանակության վիճակի պահպանման և վերականգնման տրիգերներով: Ինչպես արդեն նշվել է, տվյալ մեթոդի կիրառման արդյունքում ստացվել է մակերեսի և ընդհանուր հզորության մեծացում՝ պայմանավորված փոխարինված ռեգիստրներով (որոնք մոտ 20%-ով ավելի մեծ մակերես և էներգասպառում ունեն՝ համեմատած դասական ռեգիստրների) և լրացուցիչ ներդրված հանգույցներով: Վիճակի պահպանման և վերականգման հնարավորությամբ ռեգիստրները սնուցվում են բարձր և ցածր պոտենցիալներից: Դասական ՀՇՄ-ում՝ ցածր էներգասպառման ռեժիմում, բարձր պոտենցիալն անջատվում է, իսկ ցածր պոտենցիալը մնում է միացված սնուցմանը, որն էլ հնարավորություն է տալիս պահպանել և հետագայում վերականգնել ռեգիստրի վիճակը: Հաշվի առնելով պրոցեսորային ԻՍ-երում ռեգիստրների լայնորեն կիրառությունը՝ դրանց փոխարինման հաշվին ստացվում է մակերեսի և հզորության զգալի մեծացում: Ներդրման արդյունքում ստացվում է այնպես, որ ցածր էներգասպառման ռեժիմում գտնվող տիրույթը ամբողջությամբ չի անջատվում սնուցումից՝ պահպանելով ցածր պոտենցիալով սնուցումը, որի հետևանքով էլ ստացվում է կորստյան հզորության մեծ արժեք: ՀՇՄ-ի ներդրման ժամանակ առաջանում է նաև տվյալների դողերում մակարդակի փոխատեղումներ կատարելու անհրաժեշտություն, ինչի հետևանքով էներգասպառման աճը էլ ավելի է մեծանում (նկ. 2.11):



Նկ. 2.11. Պրոցեսորային ԻՍ-ում ՀՇՄ-ի դասական կառուցվածքը

Հաշվի առնելով քննարկված մեթոդների թերությունները՝ առաջարկվել է մոտեցում՝ տվյալների պահպանումը իրականացնել ոչ թե առանձնացված ցածր էներգասպառման ռեժիմում գտնվող հանգույցի ներսում՝ վիճակը պահպանող և վերականգնող ռեգիստրների միջոցով, այլ այդ հանգույցի եզրագծերում [85]: Ընտրված հանգույցների վիճակի պահպանումը տեղափոխելով եզրագծեր՝ հնարավորություն է ընձեռվում ամբողջությամբ անջատել ցածր էներգասպառման ռեժիմում գտնվող տիրույթը թե՛ բարձր և թե՛ ցածր պոտենցիալներից՝ իջեցնելով ընդհուպ մինչև զրո կորստի հոսանքներով պայմանավորված ստատիկ հզորության արժեքը, միաժամանակ կրճատելով վիճակի վերականգնման համար անհրաժեշտ ժամանակը, քանի որ եզրագծերում պահպանված արժեքի վերականգնումը հնարավոր է իրականացնել ավելի արագ՝ ասինքրոն համակարգի ներդրմամբ (նկ. 2.12):



Նկ. 2.12. Պրոցեսորային ԻՍ-ում ՀՇՄ-ի առաջարկվող կառուցվածքը

Առաջարկվող մոտեցումն ունի դրական ազդեցություն պրոցեսորային ԻՍ-ի կայունության վրա՝ հնարավորություն ընձեռելով խուսափել մետակայուն վիճակից, որը պայմանավորված է վիճակի պահպանման և վերականգնման ֆունկցիայի՝ ռեգիստրային հանգույցներից տեղափոխմամբ, հաշվի առնելով տրիգերների այդ վիճակ ընկնելու հակումը: Տվյալ մեթոդի օպտիմալ ներդրումն ապահովելու համար ուսումնասիրվել են մի քանի հնարավոր եղանակներ, մասնավորապես՝ նոր հանգույցների տեղակայումը եզրագծում: Այս տարբերակի պարագայում պրոցեսորային ԻՍ-ի մակերեսը զգալի մեծանում է, ինչը և կրկին հանգեցնում է էներգասպառման աճի: Խուսափելու համար նմանատիպ իրավիճակից՝ մշակվել է նոր մոտեցում՝ պահպանել վիճակն արդեն իսկ առկա հանգույցներում, որոնք գտնվում են եզրագծում: Նմանատիպ հանգույցներ են լարման մակարդակի փոխակերպիչները, որոնք բաշխված են ընտրված տիրույթի եզրագծերում [85]:

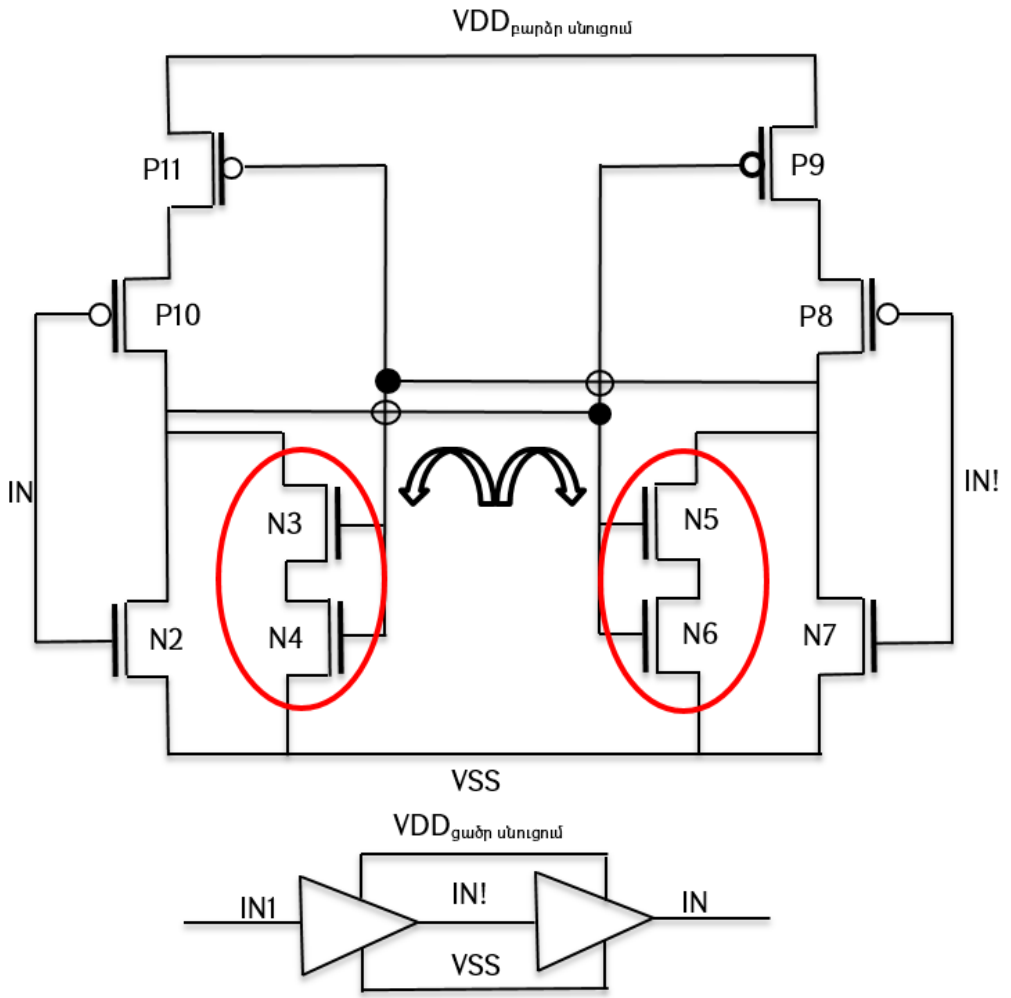
Դիտարկելով հնարավոր մոտեցումները և հանգույցները՝ մշակվել են նոր սխեմատեխնիկական լուծումներ լարման մակարդակի փոխակերխիչների (ԼՄՓ) համար: Որպես հետազոտման առարկա ընտրվել են ցածրից բարձր ԼՄՓ-ները, որոնք

ունեն ավելի բարդ կառուցվածք, մեծ հապաղումներ՝ ի տարբերություն բարձրից ցածր ԼՄՓ-ների: Հետազոտությունների արդյունքում առաջարկվել են երկու նոր ցածրից բարձր ԼՄՓ-ների կառուցվածքներ՝ առաջարկված մեթոդի ինտեգրումն ապահովելու համար:

Դասական կառուցվածքով ցածրից բարձր ԼՄՓ-ներն ունեն հիմնականում հետևյալ թերությունները [85].

- Բարձր հաճախականային տիրույթներում աշխատելու ընթացքում առաջանում է բավականին մեծ չափի N-ՄՕԿ տրանզիստորներ ունենալու անհրաժեշտություն (մոտ 4…5 անգամ մեծ, քան հաջորդաբար միացված վերին մասում տեղակայված P-ՄՕԿ տրանզիստորները):
- Դրական հետադարձ կապի արագության ոչ բավարար լինելու պատճառով ավելի շատ ժամանակ է անհրաժեշտ, որպեսզի ԼՄՓ-ի տրանզիստորները գան հաստատուն վիճակի, ինչը ավելորդ փոխանջատման հոսանքի սպառման պատճառ է դառնում:
- Նշված կարգի ԼՄՓ-ներում բավականին բարդ է ապահովել փոխանջատման խնդիրը՝ թույլ դրական հետադարձ կապի պատճառով:

Առաջարկվող առաջին տարբերակի արագագործ ԼՄՓ-ում օգտագործվել է լրացուցիչ տրանզիստորային հանգույց, որը հնարավորություն է տալիս զուգահեռ ճյուղով ապահովել լիցքաթափում, ունենալ <<ուժեղ զրո>>: Ապահովելով զուգահեռ տրանզիստորային միացում՝ գումարային դիմադրությունը իջնում է, ինչը հնարավորություն է տալիս ստանալ ավելի արագ անցում դեպի զրո մակարդակ, միաժամանակ՝ կրճատելով անցողիկ երևույթների տևողությունը և այդ ժամանակահատվածում հնարավոր էներգասպառումը, մասնավորապես՝ նրա դինամիկ բաղադրիչը: Նոր մոտեցման շնորհիվ համակարգը դառնում է ավելի արագագործ, կայուն՝ ապահովելով հուսալի փոխանջատում, ինչը նպաստում է մետակայուն վիճակի մեջ ընկնելու հավանականության նվազմանը: Առաջարկվող տարբերակի պարագայում անհրաժեշտություն չկա օգտագործելու մեծ չափի N-ՄՕԿ տրանզիստորներ՝ լիցքաթափում իրականացնելու համար, որի արդյունքում էլ ստացվում ԼՄՓ-ի ընդհանուր մակերեսի կրճատում (նկ. 2.13):

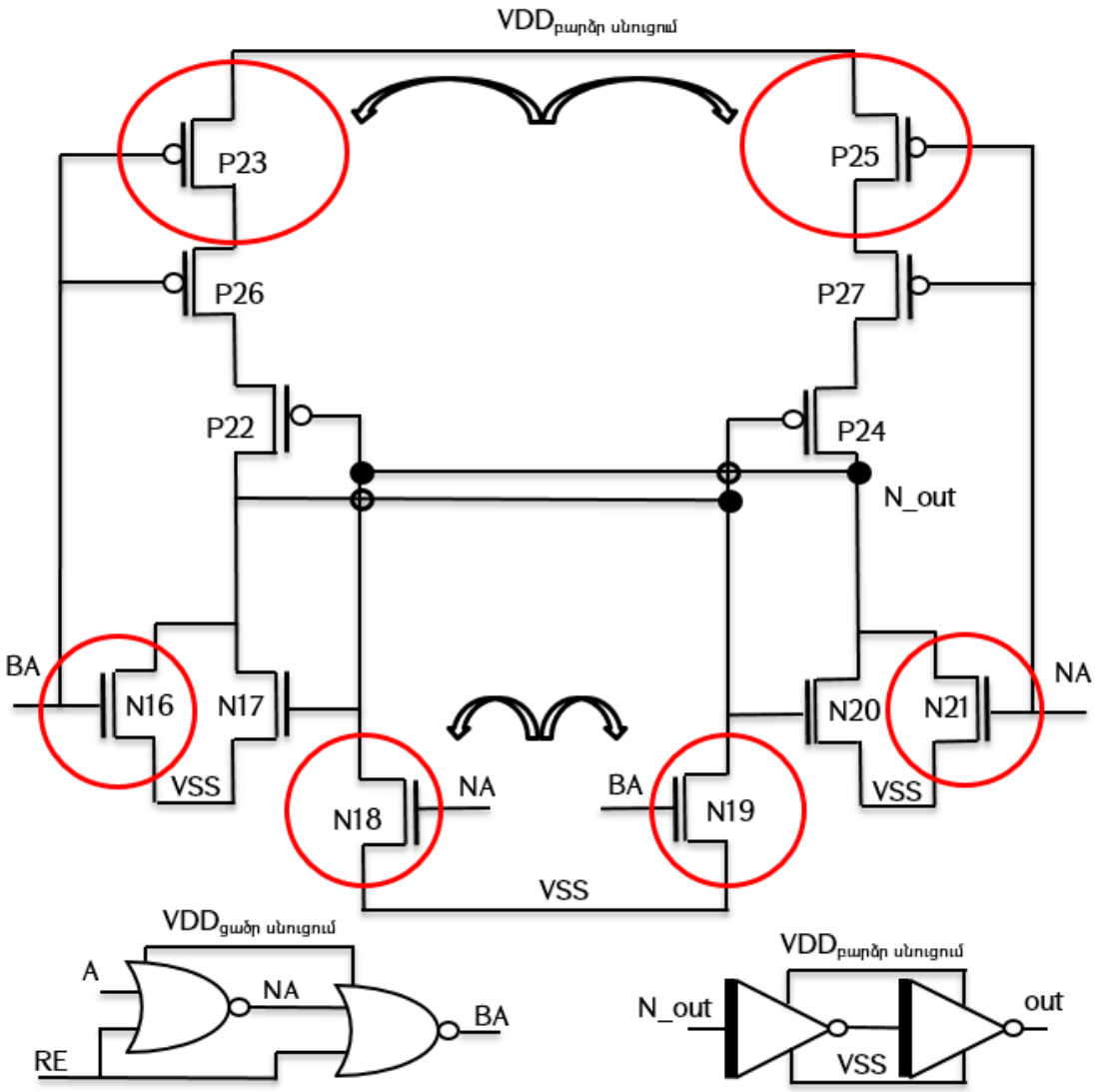


Նկ. 2.13. Ցածրից բարձր ԼՄՓ-ի առաջարկվող արագագործ կառուցվածքը

Առաջարկվող երկրորդ տարբերակի՝ վիճակը պահպանող և վերականգնող (ՎՊՎ) ԼՄՓ-ում օգտագործվում է նախորդ տարբերակի մոտեցումն ապահովելու համար բարձր արագագործություն: Առաջարկված տարբերակը պարունակում է նաև հետադարձ կապով իրար միացված երկու տրամաբանական շրջիչներ՝ վիճակը պահապանելու և վերականգնելու համար: Առաջարկվող կառուցվածքում գրեթե չկա մետակայուն վիճակում հայտնվելու հավանականություն, քանի որ զուգահեռ միացված N-ՄՕԿ (N18, N19) և հաջորդաբար միացված P-ՄՕԿ (P23, P25) տրանզիստորները սխեմատեխնիկական միացմամբ կապված են դրական հետադարձ կապին՝ սխեմային զերծ պահելով սկզբնական անկայուն վիճակ ընկնելու հավանականությունից:

Այսպիսով, տվյալ փոխակերպիչն ապահովում է երկու հիմնական գործառույթ՝ տվյալների փոխանցում մակարդակի փոփոխությամբ և վիճակի պահպանում՝

անհրաժեշտության դեպքում ապահովելով բարձր կայունություն և ցածր էներգասպառում (նկ. 2.14) [85]:

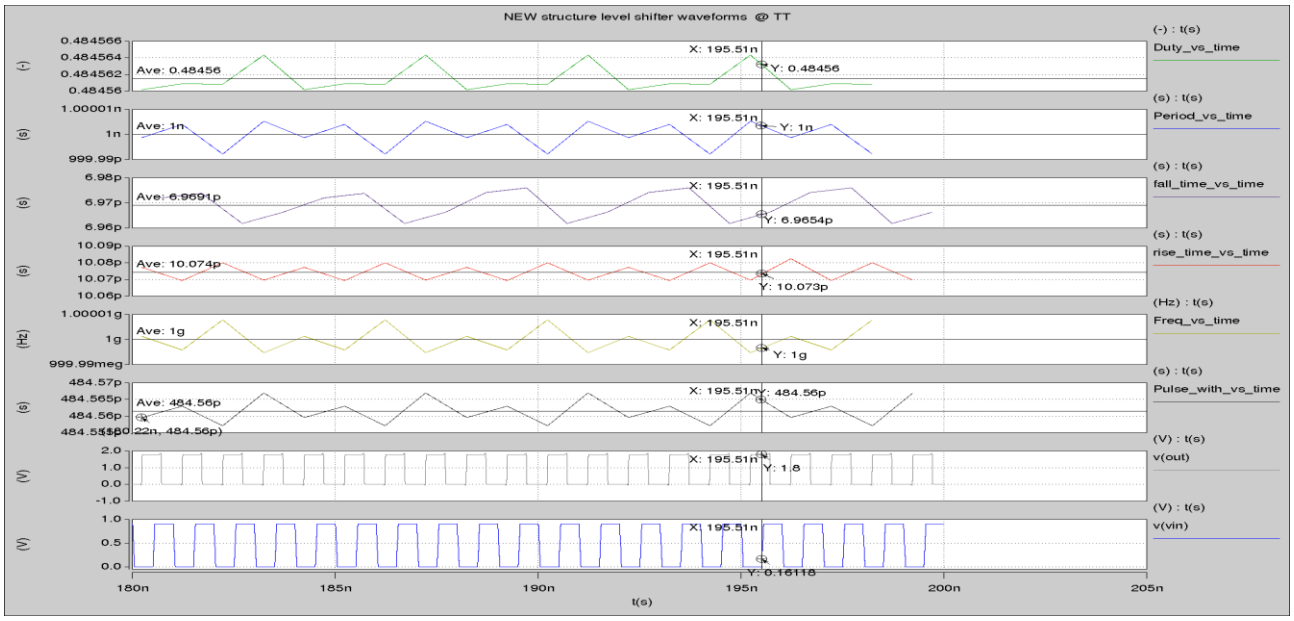


Նկ. 2.14. Ցածրից բարձր ԼՄՓ-ի առաջարկվող ՎՊՎ կառուցվածքը

Առաջարկվող մակարդակի փոխակերպիչն ունի երկու աշխատանքային վիճակ. առաջին՝ տվյալների փոխանցման և փոխակերպման ռեժիմ, երկրորդ՝ վիճակի պահպանման և վերականգնման ռեժիմ: Տարանջատելու համար այդ երկու ռեժիմները՝ մշակվել է կառավարող տրամաբանական հանգույց, որը բաղկացած է երկու <<ԿԱՄ-ՈՉ>> տարրերից՝ միացված հատուկ սխեմատեխնիկական լուծմամբ: Վիճակի պահպանման և վերականգնման ռեժիմում հայտնվելու ազդանշանների դասակարգումը հետևյալն է. ինչպես արտացոլված է սխեմայում, RE ազդանշանի տրամաբանական 1 ակտիվացնելուն պես՝ BA և NA ազդանշանները դառնում են տրամաբանական 0՝ փակելով մուտքային N-ՄՕԿ տրանզիստորները (N16 և N21): Միաժամանակ, բացվում են P-ՄՕԿ տրանզիստորները (P23 և P25, P26 և P27)՝

ապահովելով սնուցման լարման անկում ինտեգրված հիշող սխեմայի վրա, պահպանելով նախորդ վիճակը, հնարավորություն ընձեռելով սնուցման լարումը իջեցնել 0,95-ից մինչև 0,45 կամ ընդհուպ մինչև 0 ընտրված հանգույցում: Տվյալների փոխանցման և փոխակերպման ռեժիմում հայտնվելու ազդանշանների դասակարգումը հետևյալն է. RE ազդանշանին տրվում է տրամաբանական 0 մակարդակ, և միայն որոշ ժամանակ անց հնարավոր է ազդանշանի սնման լարումը 0-ից կամ 0,45-ից աճեցնել միչև 0,95: Այնուհետև կարող է կատարվել տվյալների փոխանցում: Պատկերված “ԿԱՄ” հանգույցը կարող է կրկին ապահովել տրամաբանական կայուն 1 “0,95Վ” ազդանշանը միայն այն դեպքում, երբ սնման լարումը ամբողջությամբ վերականգնվել է [85]:

Առաջարկված սխեմատեխնիկական լուծումների արդյունավետությունը գնահատելու համար կատարվել են մի քանի տարբերակների մոդելավորում և վերլուծություն HSPICE [86] և Custom Compiler [87] ծրագրային միջոցերով: Նախագծերը երեքն են՝ դասական, արագագործ և ՎՊՎ ԼՄՓ-ներ, բոլորի համար հիմք են ընդունվել սկզբնական նույն պայմանները. մուտքային լարման մակարդակը 0,95Վ է, ելքային լարման մակարդակը փոխակերպումից հետո՝ 1,8Վ: Մուտքային ազդանշանի հաճախությունը 1 ԳՀz է: ԼՄՓ-ների համար կատարվել են հետևյալ վերլուծությունները՝ լցման գործակցի, պարբերության, հաճախության, աճի և նվազման տևողությունների կախվածությունը ժամանակից, մուտքային և ելքային ազդանշանները և դրանց կախվածությունը ժամանակից (նկ. 2.15): Կատարվել են նաև մակերեսի և հզորության վերլուծությունները բոլոր երեք տարբերակների համար, հետազոտության ժամանակ օգտագործվել են նաև գրականությունից հայտնի տարբերակի հասանելի արդյունքները:



Նկ. 2.15. ՎՊՎ կառուցվածքով ԼՄՓ-ի ստացված ժամանակային բնութագրերը

Ստացվել է մոտ 48,4% լցման գործակից, անփոփոխ միջին հաճախություն՝ 1 *ԳՀց*, 200 *նվ* մոդելավորման ընթացքում: Աճի և նվազման ժամանակների տարբերությունները բնութագրում են համակարգի հավասարակշռությունը լիցքավորման և լիցքաթափման ժամանակ, որը նույնպես ստացվել է տեխնիկական առաջադրանքին համապատասխան և հաշվարկվել է (2. 1)-ով (աղ. 2.5) [85]:

$$T_{\text{աճման և նվազման ժամանակ}} = \frac{T_{\text{աճման}} - T_{\text{նվազման}}}{T_{\text{նվազման}}} * 100\% : \qquad (2.1)$$

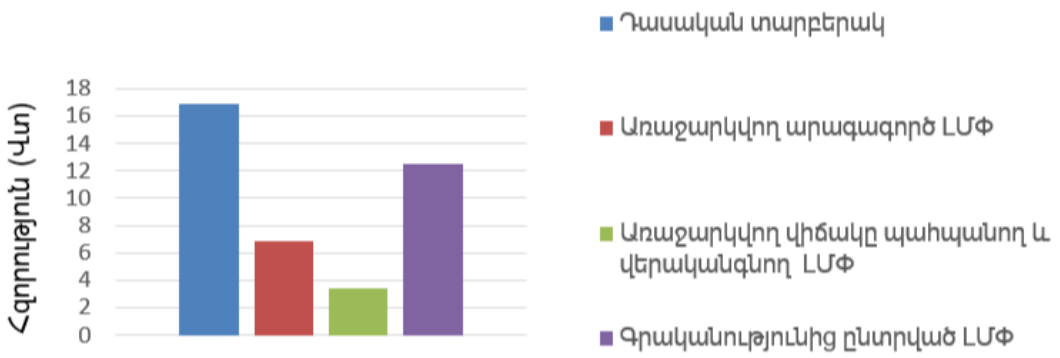
Աղյուսակ 2.5

Նախագծված ԼՄՓ-ների բնութագրերի համեմատական արդյունքները՝ կատարված

#Ելքային տվյալներ#	Դասական ԼՄՓ	Առաջարկվող արագագործ ԼՄՓ	Առաջարկվող վիճակը պահպանող և վերականգնող ԼՄՓ	Գրականությունից ընտրված տարբերակ
Լցոնման գործակից (%)	40,8	46,8	48,4	44,1
Մուտքի/Ելքի փոխ. (*նվ*)	3,65	1,8	2,1	4,745
Հզորություն (*մկՎտ*)	16,9	6,9	3,4/0,12	12,98
Մակերես (*մկմ²*)	9,18	8,415	10,94	10,3
#Մուտքի տվյալներ#				
Հաճախություն (*ԳՀց*)	1	1	1	1
Տեխնոլոգիական գոր. (*նմ*)	16	16	16	16
Մոդելավորման ժամ. (*նվ*)	200	200	200	200

չորս տարբերակների համար

Այսպիսով, առաջարկվել են նոր մոտեցումներով մակարդակի փոխակերպման հանգույցներ: Առաջարկվող երկու սխեմատեխնիկական կառուցվածքներն ապահովում են ելքային ազդանշանի 0,5…1,5 *ԳՀց* հաճախության տիրույթ: Առաջին արագագործ տարբերակում (նկ. 2.13) ապահովել են ԼՄՓ-ի արագագործությունը, հաստատուն



լման գործակցի ստացումը: Առաջարկվող երկրորդ ՎՊՎ ԼՄՓ-ի սխեմատեխնիկական հանգույցը (նկ. 2.14) ներառում է առաջին մոտեցման բոլոր առավելությունները, դրա հետ մեկտեղ ունի վիճակի պահպանման առանձնահատկություն. ներդրված մոտեցման շնորհիվ ապահովվում է համակարգի կայունությունը, զերծ պահելով մետաստաբիլ վիճակ ընկնելու հնարավորությունից: Իր վիճակը պահպանելու հնարավորության շնորհիվ՝ այն կարող է օգտագործվել բոլոր այն գերմեծ ինտեգրալ սխեմաներում, որոնք ունեն <<հանգստի ռեժիմ>>, ցածր էներգասպառման տիրույթ: Տվյալ սխեման լուծում է շատ հայտնի խնդիր, երբ անհրաժեշտություն է առաջանում կատարել ազդանշանի արագ վերականգնում և փոխանցում, որը հնարավոր չէր նախորդ տարբերակներում, քանի որ վիճակը կամ չէր պահպանվում, կամ շատ երկար ժամանակ էր պահանջվում վերականգնման համար: Առաջարկված ՎՊՎ ԼՄՓ-ի սխեմատեխնիկական հանգույցն ունի էներգասպառման արդյունավետություն 79,9% (նկ. 2.16) (իջեցում համեմատած դասական կառուցվածքի) և լման բարձր գործակից՝ 48,4%, միջին փոխանջատման տևողություն՝ 2,1 *նվ*: Մոդելավորվել է <<ստատիկ քնի>> ռեժիմը, ստացվել է նաև Էներգասպառումը այդ ռեժիմում, որը կազմել է 0,12 *մկՎտ*՝ ապահովելով կայունության բարձր գործակից՝ բոլոր քննարկված տարբերակների համեմատ:

Նկ. 2.16. Տարբեր ԼՄՓ-ների էներգասպառման համեմատական բնութագրերը

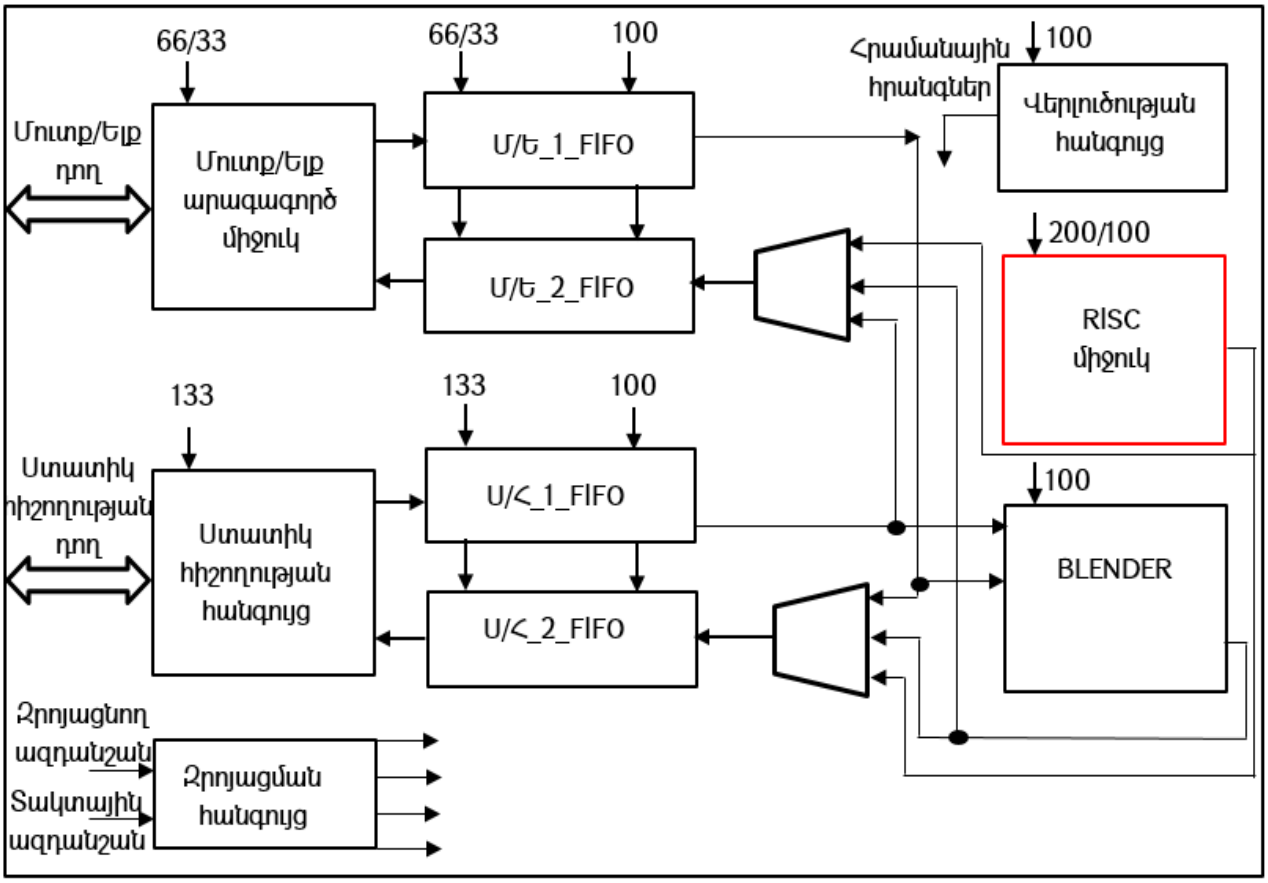
Հաշվի առնելով ստացված արդյունքները՝ կարելի է եզրակացնել, որ առաջարկված վիճակը պահպանող և վերականգնող տարբերակը լավագույնն էր տեխնիկական առաջադրանքը բավարարելու տեսանկյունից, անգամ այն պարագայում, երբ ստացվել էր մակերեսի մեծացում:

Հավաստիանալու համար պրոցեսորային ԻՍ-երում առաջարկված մեթոդի և դրա իրականացման համար մշակված սխեմատեխնիկական հանգույցների ներդրման արդյունավետության ստույգության մեջ՝ կատարվել են հետևյալ հետազոտությունները և փորձարարական գործողությունները՝ հիմք ընդունելով RISC ճարտարապետությամբ պրոցեսորային ԻՍ-երը և ՀՀԴ թվային միջուկը.

- ORCA պրոցեսորային ԻՍ-ում ներդրվել է առաջարկված հզորության շրջափակմամբ նախագծումը [88]:

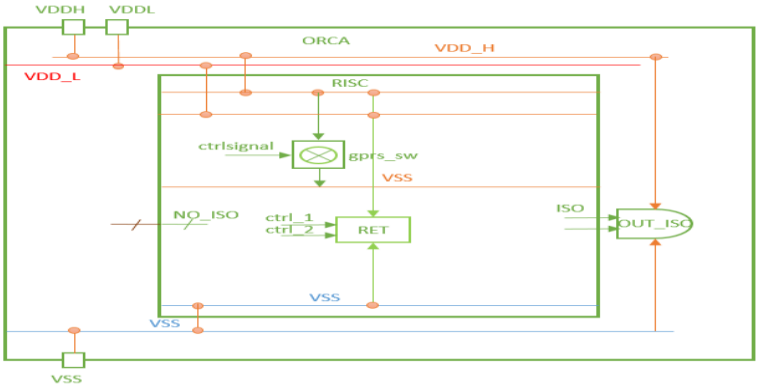
- ՀՀԴ թվային միջուկում ներդրվել է բազմասնուցում և հզորության շրջափակում մեթոդներով նախագծումը [89]:
- ORCA պրոցեսորային ԻՍ-ում ներդրվել է առաջարկված հզորության շրջափակմամբ նախագծման մեթոդը՝ օգտագործելով մշակված սխեմատեխնիկական հանգույցները:

ORCA պրոցեսորային ԻՍ-ում ներդրվել է առաջարկված հզորության շրջափակմամբ նախագծումը [88]: Հզորության շրջափակում մեթոդը կիրառվել է ORCA պրոցեսորային ԻՍ-ում՝ պարզելու համար առաջարկված լուծումների ստույգ լինելը և համեմատությունը առկա այլ տարբերակների հետ, այն դեպքում, երբ օգտագործվել են նույն պրոցեսորը, տեխնոլոգիական գործընթացը, մուտքային սահմնափակումները: Համեմատության արդյունքները հնարավորություն կընձեռեն՝ հասկանալու մեթոդի ներդրման ազդեցությունը RISC ճարտարապետության պրոցեսորների առանձնահատկությունների պարագայում: ORCA պրոցեսորային ԻՍ-ի խորացված վերլուծությունը և հետազոտությունը ցույց է տալիս, որ թվային միջուկը պարունակում է ավելի քան հազար ռեգիստրային հանգույցներ, որոնց էներգասպառումը կազմում է ընդհանուր հզորության մոտ 60%-ը: Վերլուծության այս արդյունքը հնարավորություն է տալիս ենթադրել, որ ORCA պրոցեսորային հանգույցում կարող է արդյունավետ լինել առաջարկված հզորության շրջափակման մեթոդի կիրառումը՝ էներգասպառման նվազարկման տեսանկյունից: Ռեգիստրային բոլոր հանգույցների փոփոխությունը՝ վիճակը պահպանող և վերականգնող տրիգերներով (հաշվի առնելով նրանց քանակը և կշիռը ընդհանուր հզորության ձևավորման մեջ) կարող է արդյունավետ լինել, միաժամանակ, առաջացնելով մակերեսի մեծացում: ORCA պրոցեսորային ԻՍ-ի ամենաբարձր հիերարխիկ հանգույցը՝ ORCA_TOP-ը, պարունակում է RISC միջուկը, որը և ընտրվել է որպես ՀՇՄ-ի ներդրման ենթահանգույց: RISC միջուկը պարունակում է ինչպես արագագործ, այնպես էլ դանդաղագործ հատվածներ (նկ. 2.17):



Նկ. 2.17 ORCA_TOP պրոցեսորային ԻՍ-ի ֆունկցիոնալ բլոկ-դիագրամը

ՀՇՄ-ի կիրառման ընթացքում օգտագործվել են փոխանջատող և մեկուսիչ սխեմաներ, ԼՄՓ-ներ, վիճակը պահպանող և վերականգնող ռեգիստրային հանգույցներ, որոնք համապատասխանաբար բաշխվել են ըստ ընտրված տիրույթի: Նախագծի ներդրման ընթացքում մշակվել է ՄՀՁ նկարագրությունը հատուկ ORCA պրոցեսորի համար՝ ՀՇՄ-ի կիրառմամբ, որտեղ նկարագրվել են հզորության տիրույթները իրենց արժեքներով, ինչպես նաև մեկուսիչ և ռեգիստրային տարրերի փոփոխության ալգորիթմը: Նախագծման արդյունքում ստացվել է ՄՀՁ դիագրամը՝ հզորության առանձնացված տիրույթներով, միջմիացման ուղիներով և լրացուցիչ սխեմատեխնիկական հանգույցներով, որոնք անհրաժեշտ են մեթոդի սահուն ներդրումն ապահովելու համար (նկ. 2.18):



Նկ. 2.18. ՀՇՄ-ով նախագծման համար ORCA պրոցեսորային ԻՍ-ի ՄՀՁ դիագրամը

Նախագիծը կատարվել է տրամաբանական մակարդակից մինչև ֆիզիկական մակարդակ՝ որպես հիմք ընդունելով RISC միջուկի հաճախությունը՝ 200 *Մhg*, մուտք/ելք և ստատիկ հիշողության հանգույցների հաճախությունը՝ 75 *Մhg*, ֆիզիկական նախագծման խտությունը՝ 30%: ՀՇՄ-ով նախագծման ստացված փորձարարական արդյունքների համեմատությունը դասական տարբերակի հետ բերված է ստորև (աղ. 2.6):

Աղյուսակ 2.6

ORCA պրոցեսորի ստացված արդյունքները հզորության շրջափակմամբ նախագծման և դասական տարբերակների դեպքում

Պարամետրերը	Հզորության շրջափակմամբ	Դասական նախագծում
Տակտային հաճախությունը (*ՄՀg*)	200	200
Մուտքային ազդանշանների անհրաժեշտ ժամանակային պաշարը (*նվ*)	27,48	20,21
Մուտքային ազդանշանների իրական ժամանակային պաշարը (*նվ*)	-24,86	-20,05
Անհրաժեշտ և իրական ժամանակային պաշարների տարբերությունը (*նվ*)	2,62	0,16
Ընդհանուր հզորությունը (*մլՎտ*)	69,42 ~(- 23%)	90,15
Մտավոր սեփականության հանգույցի մակերեսը (*մկմ²*)	16340,8	16340,796387
Ընդհանուր օգտագործված ստանդարտ բջիջների մակերեսը (*մկմ²*)	807616,35 (+27%)	589303,444805
Ընդհանուր մակերեսը (*մկմ²*)	823956,35 ~(+26%)	605644,241192

Հզորության շրջափակման մեթոդի կիրառմամբ նախագծված ORCA պրոցեսորային ԻՍ-ում ստացվել են հետևյալ արդյունքները. ընդհանուր հզորությունը իջել է ավելի քան 23%-ով՝ համեմատած դասական նախագծի, որը, հիմնականում պայմանավորված է ցածր էներգասպառման ռեժիմի սահմանմամբ և դրա արդյունքում կորստի հզորության կրճատմամբ: Միաժամանակ, տվյալ մեթոդի ներդրման ժամանակ մոտավորապես 27%-ով մեծացել է ընդհանուր ԻՍ-ի զբաղեցրած մակերեսը ԿԲ-ում, որն էլ հիմնականում պայմանավորված է ռեգիստրային բջիջների մակերեսի աճով, որը կազմել է 26%: 200 *ՄՀց* հաճախության պայմանում աշխատունակությունը պահպանվել է, միաժամանակ մեծացել է անհրաժեշտ և իրական ժամանակային պաշարների տարբերության արժեքը, ինչը նշանակում է, որ ԻՍ-ն դառնում է ավելի կայուն տեխնոլոգիական գործընթացի հնարավոր շեղումների նկատմամբ, մասնավորապես՝ պարազիտային ունակություններով պայմանավորված կորուստների, որոնք կարող են ունենալ մեծ ազդեցություն ԻՍ-ի ժամանակային պարամետրերի վրա:

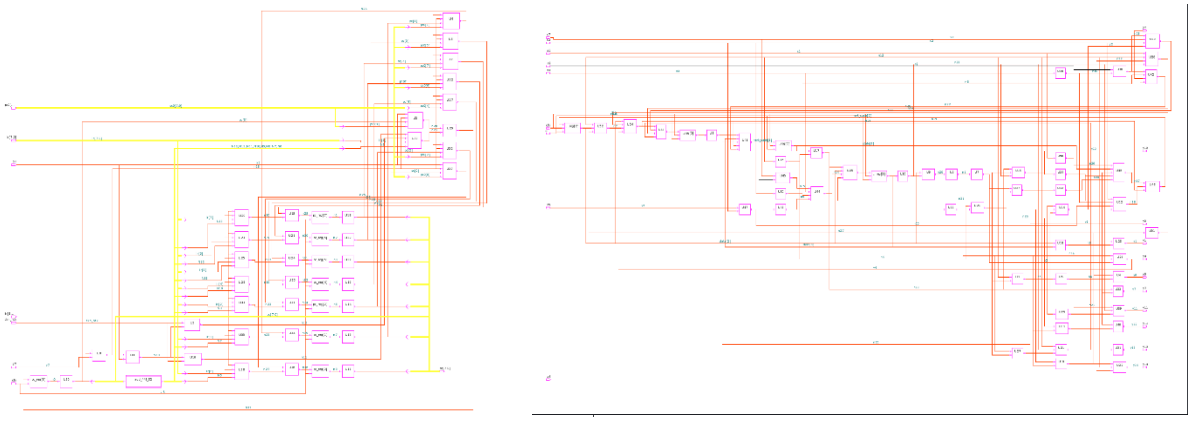
Այսպիսով, ամփոփելով ստացված արդյունքները, կարելի է եզրակացնել, որ ՀՇՄ-ի ներդրմամբ ORCA պրոցեսորային ԻՍ-ում ստացվել է բավականին արդյունավետ ընդհանուր հզորության նվազարկում (23%), միաժամանակ, պահպանելով ժամանակային պարամետրերի վրա նվազագույն ազդեցություն, այնուամենայնիվ, մակերեսի պարագայում ստացվում է զգալի մեծացում (27%): Այս մեթոդով RISC ճարտարապետության պրոցեսորների կիրառումը օպտիմալ կլինի այն դեպքերում, երբ մակերեսի մեծացումը կարող է մեծ մասամբ անտեսվել՝ համաձայն տեխնիկական առաջադրանքի:

ՀՀԴ թվային միջուկում ներդրվել է միացյալ բազմասնուցում և հզորության շրջափակում մեթոդներով նախագծումը [89]: Էներգասպառման նվազարկման անհրաժեշտությունը արդիական է նաև այնպիսի համակարգերում, ինչպիսիք են մուտք/ելք հանգույցները և դրանց թվային ազդանշանի մշակման բլոկները: Հաշվի առնելով այս հանգամանքը՝ կատարվել է հետազոտություն՝ պարզելու համար դրանցում էներգասպառման ընդհանուր հարցերը և ստացված արդյունքների համեմատությունը պրոցեսորային ԻՍ-երի հետ: Որպես հետազոտության առարկա ընտրվել է համապիտանի հաջորդական դողի թվային միջուկը, որը ներկայացնում է բարձր արտադրողականությամբ ազդանշանի մշակման հանգույց և բավականին լայն տարածում ունի արդի սարքավորումներում: Ուսումնասիրելով ՀՀԴ թվային միջուկի

կառուցվածքը, որոշում է կայացվել կիրառել ցածր էներգասպառման նախագծման երկու մեթոդները, որոնք են՝ բազմասնուցմամբ նախագծումը և հզորության շրջափակմամբ նախագծումը: Նախագծման ընթացքում ստեղծվել է ՄՀՁ նկարագրությունը՝ միասնական ինտեգրում ապահովելու համար թե՛ տրամաբանական և թե՛ ֆիզիկական նախագծման փուլերում: ՀՀԴ թվային միջուկի հիմնական ֆունկցիան կայանում է ազդանշանի հաջորդականից զուգահեռ և զուգահեռից հաջորդական փոփոխություններում: ՀՀԴ թվային միջուկը ներառում է հետևյալ ենթահանգույցները՝

- թեստավորման հանգուց (scan_mux),
- զրոյացնող տրամաբանության հագույց (reset_logic),
- զուգահեռ ազդանշանը հաջորդականի ձևափոխման հանգույց (ԶԱՀՁՀ),
- հաջորդական ազդանշանը զուգահեռի ձևափոխման հանգույց (ՀԱԶՁՀ):

ՀՀԴ թվային միջուկի ենթահանգույցների վերլուծությունը ցույց է տալիս, որ ԶԱՀՁՀ և ՀԱԶՁՀ հանգույցները պարունակում են մեծ քանակով ռեգիստրային հանգույցներ, ինչը պայմանավոված է նշված ենթահանգույցներում ընդգրկված պահպանող և փոխանցող տրիգերների փնջով: Այս ենթահանգույցները ընտրվել են որպես ցածր էներգասպառման տիրույթներ, որոնք միացված են ցածր պոտենցիալով սնման լարմանը: Միաժամանակ, նույն հանգույցներն ընտրվել են որպես սնման լարման անջատման տիրույթներ՝ հզորության շրջափակման մեթոդի ներդրման համար: Համապատասխանաբար ԶԱՀՁՀ և ՀԱԶՁՀ ենթահանգույցների բոլոր ռեգիստրները փոխարինվում են վիճակը պահպանող և վերականգնող ռեգիստրներով և ԼՄՓ-ներով: ՀՀԴ թվային միջուկի մի քանի նախագիծ է կատարվել, բոլորի դեպքում ընտրվել են նույն մուտքային սահմանափակումները՝ 300 *ՄՀց* տակտային հաճախություն, տեխնոլոգիական գործընթաց և այլն (նկ. 2.19ա): Նախագիծը կատարվել է Design Compiler ծրագրային միջոցով, և որպես ստանդարտ բջիջների գրադարան օգտագործվել է տիպային տարբերակը, որի սնուցման լարումը 0,95 Վ է (նկ. 2.19բ): ՀՀԴ թվային միջուկում լրացուցիչ տրամաբանական հանգույցների ներդրումը հանգեցրել է ֆունկցիոնալ թեստավորման բարդացմանը: Այս հիմնահարցի լուծման համար կիրառվել են ֆունկցիոնալ թեստավորման մոտեցումները [90]:



ա) բ)

Նկ. 2.19 ա) ՀՀԴ թվային միջուկի բազմասնուցմամբ նախագծի տրամաբանական կառուցվածքը, բ) ՀՀԴ թվային միջուկի հզորության շրջափակմամբ նախագծի տրամաբանական կառուցվածքը

Ստորև բերված են ՀՀԴ թվային միջուկի ստանդարտ և առաջարկվող նոր մոտեցմամբ նախագծերի ստացված համեմատական արդյունքները (աղ. 2.7):

Աղյուսակ 2.7

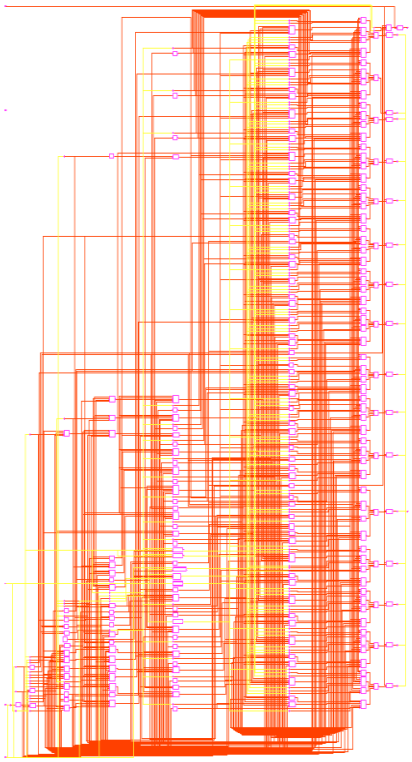
ՀՀԴ թվային միջուկի ստանդարտ և առաջարկվող նոր մոտեցմամբ նախագծերի ստացված համեմատական արդյունքները

Պարամետրերը	Նոր մոտեցմամբ նախագծում	Ստանդարտ նախագծում
Տակտային հաճախություն (*ՄՀg*)	300	300
Ընդհանուր հզորություն (*մՎտ*)	13,4 ~(-41%)	32,3
Ընդհանուր մակերես (*մկմ²*)	130,579252 ~(+11%)	116,579252

Այսպիսով, ստացվել է մոտ 41%-ով էներգասպառման նվազարկում առաջարկվող մոտեցմամբ նախագծման պարագայում, միաժամանակ ունենալով 11%-ով ավելացում ընդհանուր մակերեսի մեջ՝ ի հաշիվ ներդրված լրացուցիչ տարրերի: Ամփոփելով ստացված արդյունքները՝ կարելի է եղրակացնել, որ ՀՇՄ և բազմասնուցման մեթոդները, երբ ներդրվում են ճիշտ վերլուծության արդյունքում և համապատասխան ՄՀՁ-ի մշակմամբ, կարող են արդյունավետ լինել նաև այլ թվային սինքրոն

համակարգերում, ինչպիսիք են, օրինակ, մուտ/ելք սարքերի թվային մշակման հանգույցները:

ORCA պրոցեսորային ԻՍ-ում ներդրվել են առաջարկված բազմասնուցմամբ և հզորության շրջափակմամբ նախագծման մեթոդները՝ մշակված ԼՄՓ-ների հետ միասին: Ապահովելու համար առաջարկվող միջոցների ներդրումը, նախ՝ մշակվել է ստանդարտ բջիջների նոր գրադարան՝ որը պարունակում է առաջարկված սխեմատեխնիկական հանգույցներ, որոնք անհրաժեշտ են պրոցեսորային ԻՍ-ի նախագծում կիրառելու համար: Կատարվել է դրանց մոդելավորում տարբեր գործընթացների, լարումների, ջերմաստիճանների դեպքում՝ համապատասխան տեխնիկական առաջադրանքին: Մշակվել է ՄՀՁ նկարագրությունը հատուկ առաջարկված տարբերակի համար, որում նշված են վիճակի պահպանման և վերականգնման ֆունկցիա կատարող ինտեգրված ԼՄՓ-ները: Կատարվել է թվային ավտոմատ նախագծում՝ օգտագործելով վերը թվարկված մուտքային բաղադրիչները: Հատուկ նշանակության ԼՄՓ-ների ինտեգրումը կատարվել է տրամաբանական նախագծման փուլում՝ օգտագործելով հատուկ սահմանափակումներ (նկ. 2.20):



Նկ. 2.20. ORCA պրոցեսորային ԻՍ-ի առաջարկված հզորության շրջափակման մեթոդի կիրառման արդյունքում ստացված տրամաբանական կառուցվածքը

Այսպիսով, հաստատելու համար առաջարկված մեթոդի արդյունավետությունը՝ կատարվել է ստացված փորձարարական արդյունքների համեմատում թե՛ գրականությունից հայտնի տարբերակի, թե՛ դասական ՀՇՄ-ի տարբերակի և թե՛ առաջարկվող ՀՇՄ-ի և բազմասնուցման մեթոդներով նախագծերի միջև (աղ. 2.8):

Աղյուսակ 2.8

ORCA պրոցեսորի առաջարկվող և դասական տարբերակների ստացված արդյունքները

Պարամետրերը	Հզորության շրջափակմամբ	32 կարգանի RISC DLX միջուկը	Առաջարկվող տարբերակով (ՀՇՄ)
Տակտային հաճախություն *(ՄՀց)*	200	200	200
Ընդհանուր հզորություն *(մՎտ)*	69,42	98,7	35,53 (-51%)/(-64%)
Ընդհանուր մակերես *(մկմ²)*	823956,35	801626,79	852794,46 (+3,5%)/(+6%)

Այսպիսով, առաջարկվող մեթոդի և համապատասխան սխեմատեխնիկական հանգույցների ներդրման արդյունքում ստացվել է 51%-ով էներգասպառման նվազարկում՝ ի համեմատ առաջին՝ հզորության շրջափակման մեթոդով կատարված նախագծի: Միաժամանակ, մակերեսի մեծացումը 3,5%-ի չափով է, որը բավականին լավ ցուցացանիշ է, այն դեպքում, երբ ստացվել է հզորության նվազարկման էական արդյունք: Գրականությունից ընտրված տարբերակի համեմատ՝ ստացվել է 64%-ի էներգասպառման նվազարկում, միաժամանակ, մակերեսի մեծացումը կազմել է 6%: Ստացված տվյալների վերլուծության արդյունքում կարելի է եզրակացնել, որ առաջարկված մեթոդը իսկապես ապահովում է էներգասպառման զգալի արդյունավետություն, իսկ ներդրված սխեմատեխնիկական հանգույցները կատարում են առաջադրված խնդիրը՝ ապահովելով հուսալի փոխանջատում, արագագործություն, լայն հաճախականային տիրույթներում աշխատունակություն: Պրոցեսորային մակարդակից ստացվում է ստատիկ և դինամիկ հզորությունների կրճատում, վիճակի արագ վերականգնում և հետևաբար՝ տարբեր ռեժիմների միջև փոխանջատումների ժամանակային հապաղումների կրճատում: Միաժամանակ, ներդրված սխեմատեխնիկական հանգույցների հիմնական թերություններն են՝ մակերեսի

մեծացումն ու մուտքից ելք ժամանակային հապաղումների աճը, որոնք ընդհանուր առմամբ կարելի է ընդունելի համարել, քանի որ չեն հանգեցնում ֆունկցիոնալ սխալների:

2.5. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման համար առաջարկված դինամիկ լարման և հաճախության մասշտաբավորման եղանակը

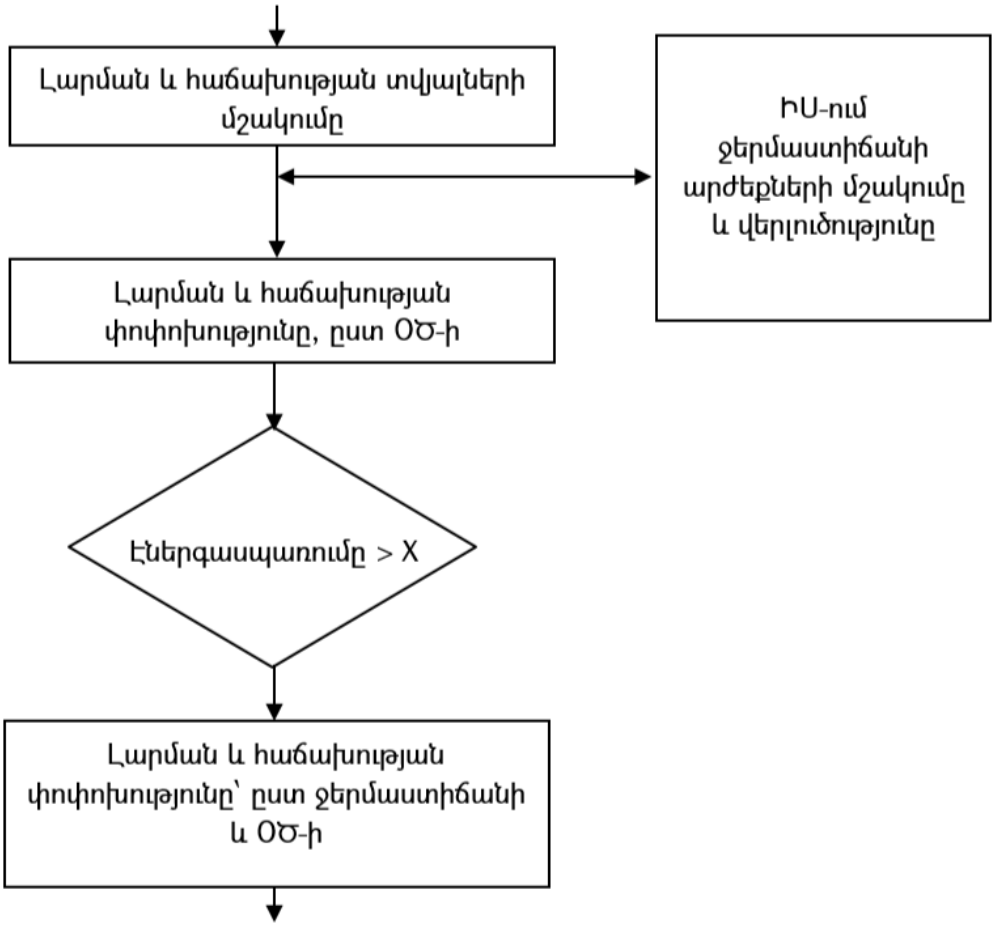
1.2–ում քննարկված պրոցեսորային ԻՍ-երում բազմասնուցման մեթոդի հիմնական թերություններն են՝ ժամանակային հապաղումները և դրանց վերլուծությունը, ԻՍ-ի մակերեսի մեծացումը և դրա հատակագծումը: Բազմասնուցման տարբեր կիրառությունները, մասնավորապես՝ դինամիկ լարման և հաճախության մասշտաբավորումը, ունի իրեն բնորոշ թերությունները՝ համակարգի կայունությունը, արագագործությունը և միայն օպերացիոն ծանրաբեռվածության (ՕԾ) հիման վրա կատարված՝ լարման և հաճախության դինամիկ փոփոխությունները: Տվյալ պարագայում առաջանում է համակարգի ավելի լայն պարամետրերի վերլուծության անհրաժեշտություն՝ խուսափելու համար մասշտաբավորման հնարավոր շեղումների արդյունքում իրական սարքում առաջացած սխալներից: Ներդնելով ԻՍ-ում ավելի շատ պարամետրերի վելուծության ալգորիթմ՝ հնարավորություն է ստեղծվում համակարգում լարման և հաճախության փոփոխություն կատարելիս հաշվի առնել այդ փոփոխությունների ազդեցությունը ընտրված պարամետրերի վրա:

Քննարկելով ԻՍ-ի տարբեր կարևորագույն պարամետրերը, որոնց արժեքների փոփոխությունը կարող է ազդեցություն ունենալ լարման կամ հաճախության վրա, որոշում է կայացվել ընդունել ջերմաստիճանային կախվածությունը առաջնային կարևորության միավոր ԻՍ-ում լարման և հաճախության դինամիկ մասշտաբավորում իրականացնելիս: Առաջարկվել է պրոցեսորային ԻՍ-ում բազմասնուցման մեթոդի՝ դինամիկ լարման և հաճախության մասշտաբավորման նոր մոտեցում՝ հիմնված ջերմաստիճանային կախվածության վրա:

Առաջարկվող մոտեցման պարագայում հնարավոր է լինում հզորության կառավարման հանգույցում, ակտիվ աշխատանքային ռեժիմում, ստանալ ջերմաստիճանային տվյալներ և դրանց արժեքների հիման վրա իրականացնել մեթոդը: Աշխատանքի ընթացքում հնարավորություն է ստեղծվում ընտրված

հանգույցներում, որտեղ ջերմաստիճանը բարձր է, դինամիկ մասշտաբավորել լարումը և հաճախությունը, փոքրացնելով դրանց արժեքները: Այն պարագայում, երբ ջերմաստիճանը ցածր է, կատարվում է հակառակ գործողությունը՝ մեծացնելով լարումը և հաճախությունը:

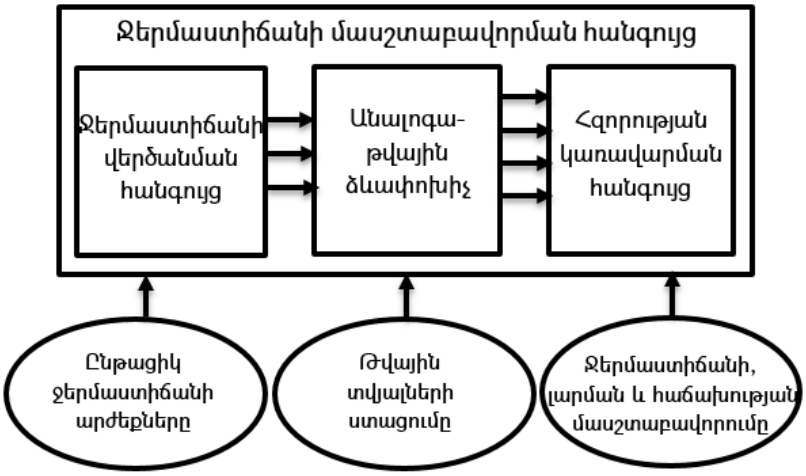
Ջերմաստիճանից կախվածության հաշվարկման գործողությունը կատարվում է դինամիկ լարման և հաճախության մասշտաբավորման տվյալների մշակման ենթահանգույցից անմիջապես հետո (նկ. 2.21): Օպերացիոն ծանրաբեռնվածությունը, որը դրված է մեթոդի դասական տարբերակի հիմքում, հնարավոր է, որ տրամաբանական մակարդակում հանգեցնի սխալների և փոխանցի դրանք մինչև ֆունկցիոնալ մակարդակ: Այդ իսկ պատճառով օպերացիոն ծանրաբեռնվածությանը զուգահեռ ներդրվում է նոր կախվածություն, որը կարող է իրականացնել պրոցեսորային ԻՍ-ի տարբեր ենթահանգույցներում ջերմաստիճանային տվյալների վերլուծություն՝ հասկանալու համար հնարավոր օպտիմալ լարման և հաճախության մասշտաբավորման արժեքը:



Նկ. 2.21. Առաջարկվող դինամիկ լարման և հաճախության մասշտաբավորման եղանակի բլոկ-դիագրամը

Ջերմաստիճանի արժեքների վերլուծությունը բնութագրում է ԻՍ-ի ֆիզիկական մակարդակում տեղի ունեցող գործընթացները և ունի ճշգրտության բարձր մակարդակ:

Ջերմաստիճանի արժեքների վերլուծությունը ակտիվ աշխատանքային ռեժիմում դյուրին գործ չէ: Անհրաժեշտ են համակարգեր, որոնք կապահովեն ջերմաստիճանային արժեքների որոշումը՝ ըստ առանձնացված տիրույթների: Առաջարկված մոտեցման իրականացման համար մշակվել է ջերմաստիճանի մասշտաբավորման հանգույց (նկ.2.22):



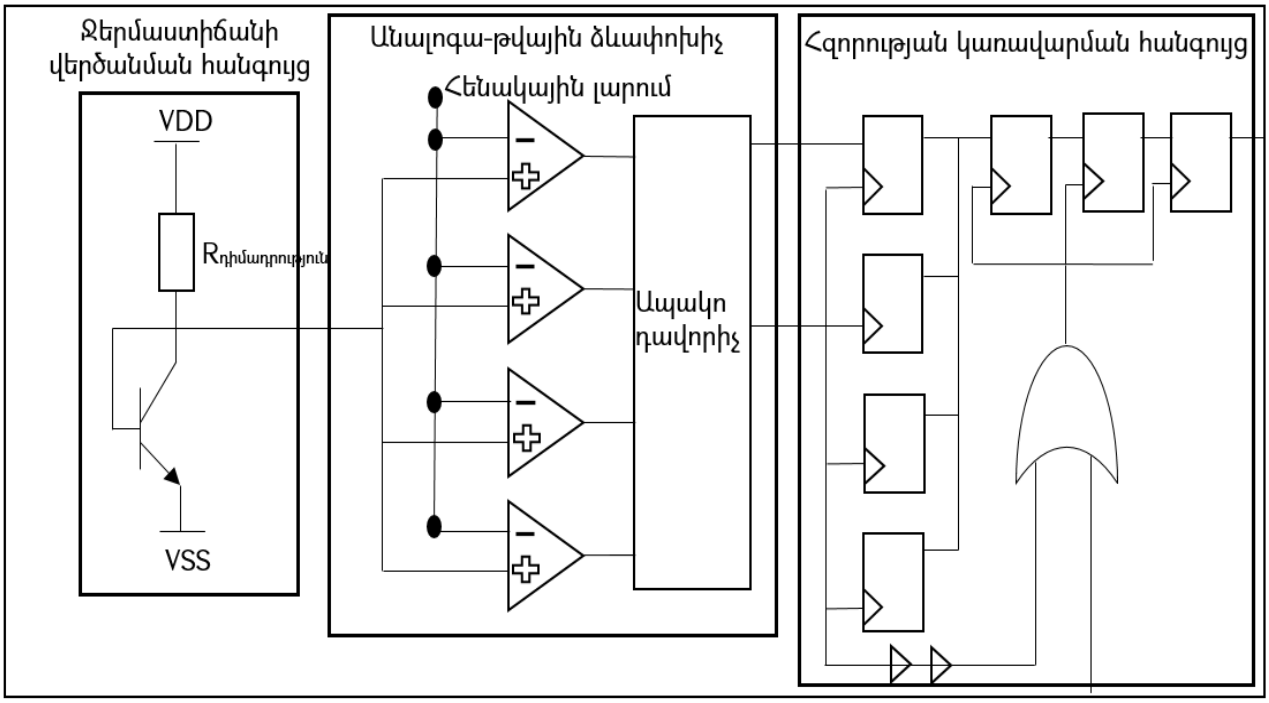
Նկ. 2.22. Առաջարկվող ջերմաստիճանի մասշտաբավորման հանգույցի կառուցվածքը

Ջերմաստիճանի մասշտաբավորման հանգույցը (ՋՄՀ) կազմված է հետևյալ երեք բլոկներից՝

- ջերմաստիճանի վերծանման հանգույց,
- անալոգա-թվային ձևափոխիչ,
- հզորության կառավարման հանգույց:

Ջերմաստիճանի վերծանման հանգույցի հիմնական նպատակն է ստանալ և վերլուծել ակտիվ աշխատանքային ռեժիմներում ընտրված ենթահանգույցներում ջերմաստիճանի ընթացիկ արժեքները: Անալոգա-թվային ձևափոխիչը անհրաժեշտ է նախորդ հանգույցից ստացված անալոգային ազդանշանները թվային կարգայնության բերելու համար: Վերջինը հզորության կառավարման հանգույցն է, որը պատասխանատու է մասշտաբավորման համար՝ հաշվի առնելով ԻՍ առանձին

տիրույթներում ջերմաստիճանի արժեքները: Այս հանգույցը արդեն իսկ ունի իրենում ինտեգրված համապատասխան լարման և հաճախության արժեքներ՝ ընտրված համաձայն ջերմաստիճանի և ՕԾ-ի փոփոխության: Հզորության կառավարման ենթահանգույցը ղեկավարում է տարբեր ցածր էներգասպառման (ՑԷ) ռեժիմները, որոնք արդեն իսկ ներդրված են դրանում: Առաջարկված տարբերակը պարունակում է երեք ՑԷ ռեժիմներ, որոնք սահմանվում են ՄՀՁ-ի միջոցով: ՑԷ առաջին ռեժիմը փոխանջատում է բոլոր տիրույթները, անջատելով դրանք սնուցումից՝ առաջացնելով պասսիվ վիճակ: Այս ռեժիմը հիմնականում կիրառվում է այն պարագայում, երբ պրոցեսորային ԻՍ-ն ամբողջությամբ գտնվում է <<սպասման վիճակում>> և չի կատարում ակտիվ գործողություններ: ՑԷ երկրորդ ռեժիմը փոխանջատում է ընտրված մի քանի տիրույթներ, որոնք տվյալ ռեժիմում ներգրավված չեն ակտիվ գործողությունների կատարմանը, մասնավորապես՝ գլխավոր նշանակության ռեգիստրները և օպերատիվ հիշողությունը: ՑԷ երրորդ ռեժիմը դինամիկ մասշտաբավորում է լարման և հաճախության արժեքները՝ համապատասխան ՋՄՀ-ից ստացված տվյալների (նկ.2.23):



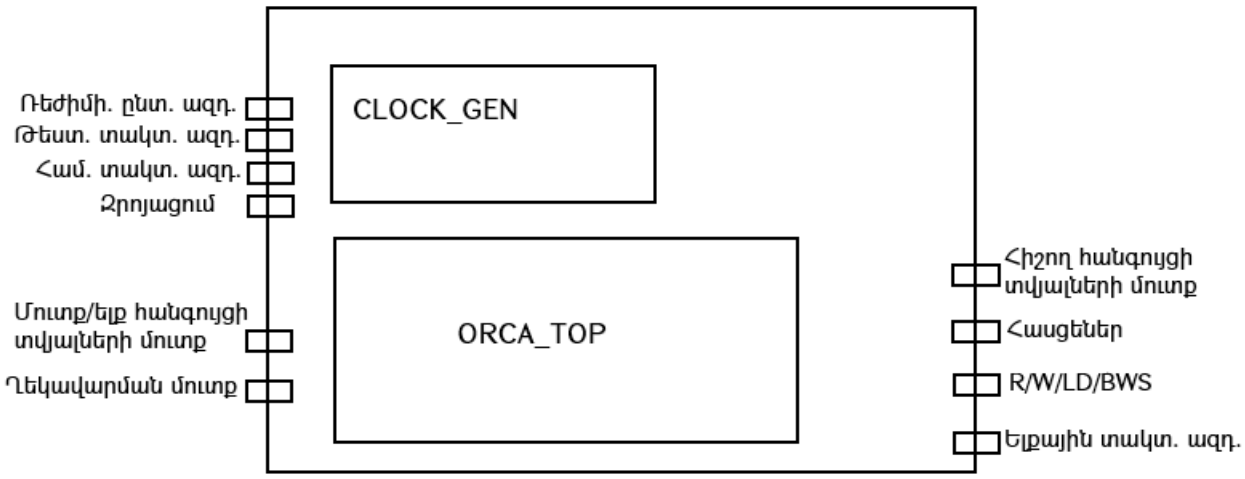
Նկ. 2.23. Առաջարկվող ջերմաստիճանի մասշտաբավորման հանգույցի սխեմատեխնիկական կառուցվածքը

Առաջարկված մոտեցման կիրառական նշանակությունը հիմնավորելու համար կատարվել են երկու նախագծեր RISC ճարտարապետության պրոցեսորային ԻՍ-երի հիման վրա.

- ORCA պրոցեսորային ԻՍ-ում ներդրվել է բազմասնուցում նախագծումը [91]:
- Simple RISC պրոցեսորային ԻՍ-ում ներդրվել է առաջարկված դինամիկ լարման և հզորության մասշտաբավորման եղանակը [92]:

Առաջին հետազոտությունը կատարվել է՝ պարզելու համար առաջարկված բազմասնուցման նախագծման խնդիրները և մարտահրավերները RISC ճարտարապետության ORCA պրոցեսորային ԻՍ-ի դեպքում: Երկրորդ հետազոտությունը կատարվել է՝ գնահատելու համար առաջարկված մոտեցման արդյունավետությունը simple RISC պրոցեսորային ԻՍ-ի պարագայում:

ORCA պրոցեսորային ԻՍ-ում ներդրված բազմասնուցման նախագծումը [91]: ORCA-ն 32-բիթ կենտրոնական մշակման հանգույցով պրոցեսոր է: Այն կազմված է հիմնական երկու հանգույցներից՝ CLOCK_GEN և ORCA_TOP: CLOCK_GEN-ն պարունակում է երկու PLL-ներ [93] և տակտային ազդանշանի բազմապատկիչներ, որոնք պատասխանատու են տակտային ազդանշանների գեներացման և մատակարարման համար: Պրոցեսորային ԻՍ-ը պարունակում է բարձր արագագործությամբ RISC միջուկ՝ հզորության պահպանման ռեժիմով: Հզորության պահպանման ռեժիմում BLENDER բլոկը անջատվում է, և RISC միջուկում տակտային ազդանշանի հաճախությունն իջնում է կիսով չափ: Բացի հզորության պահպանման ռեժիմից, ORCA-ն ունի նաև հիմնական երկու աշխատանքային ռեժիմներ՝ ֆունկցիոնալ և թեստավորման: Թեստավորման ռեժիմում տակտային ազդանշանները, որոնք նախատեսված են ֆունկցիոնալ ռեժիմում աշխատանքի համար, չեն օգտագործվում: Բոլոր ասինքրոն հանգույցները, որոնք ունեն տարբեր տակտային ազդանշանային տիրույթներ, մեկուսացված են երկմուտքանի FIFO-ներով [94] (նկ. 2.24):



Նկ. 2.24. ORCA պրոցեսորային ԻՍ-ի բլոկ-սխեման

ORCA պրոցեսորը պարունակում է նաև վեց թեստավորման ուղիներ, ինչը հնարավորություն է տալիս ունենալ բավականին մեծ թեստավորման ծածկույթ: Այս թեստավորման ուղիները իրականում վեց տեղաշարժող ռեգիստրային հանգույցներ են, որոնք կազմված են ազդանշանի ճակատով կառավարվող տրիգերներից, որոնց մուտքերը միացված են մուլտիպլեքսորային տարրերին: Տեղաշարժող ռեգիստրներն ակտիվանում են մուտքային <<թեստ ռեժիմ>> և <<թեստ ակտիվացում>> ազդանշանների միջոցով [95]: Աղ. 2.9-ում ներկայացված է ORCA պրոցեսորի ընդհանուր աշխատանքային ռեժիմների և ղեկավարող ազդանշանների ընդհանրացված ցանկը:

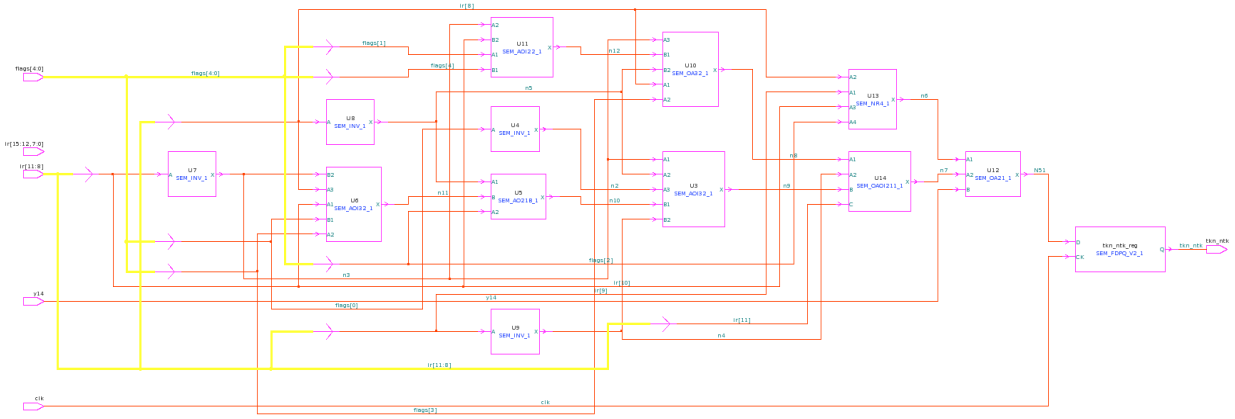
Աղյուսակ 2.9

ORCA պրոցեսորի ընդհանուր աշխատանքային ռեժիմները և ղեկավարող ազդանշանները

Pm66en	Ղեկավարում է մուտք/ելք հանգույցի աշխատանքային հաճախությունը, սահմանելով 33 (0) կամ 66 (1) *ՄՀg*
Հզորության պահպանման ռեժիմ	Ղեկավարում է RISC միջուկի աշխատանքային հաճախության արժեքը, սահմանելով 200 (0) կամ 100 (1) *ՄՀg*
Ռեժիմի ընտրության կառավարում	Ղեկավարում է ORCA –ի աշխատանքային ռեժիմները՝ թեստավորման ռեժիմ (1) կամ ֆունկցիոնալ ռեժիմ (0): Հավաստում է, որ բոլոր ռեգիստրները միացված են կամ թեստավորման, կամ

	ֆունկցիոնալ տակտային ազդանշաններին: Ղեկավարում է նաև բարձր մակարդակի զրոյացնող ազդանշանը:
Թեստ ակտիվացում	Ղեկավարում է ռեգիստրային հանգույցների աշխատանքային վիճակները, որոնք երկուսն են՝ տեղաշարժող (1) և ընդունող (0): Ֆունկցիոնալ ռեժիմի պարագայում ռեգիստրները գտնվում են ընդունում վիճակում:

ORCA պրոցեսորի նախագծման ընթացքում կատարվել է մուտքային նկարագրության ձևավորումից մինչև վերջնական տոպոլոգիական կառուցվածքի ստացումը՝ անցնելով թվային ավտոմատ նախագծման գործընթացի բոլոր քայլերով: Պրոցեսորի վերլուծությունը կատարվել է՝ նպատակ ունենալով պարզել ընդհանուր հզորության ձևավորման բնույթը, նրա ստատիկ և դինամիկ բաղադրիչների համամասնությունը առանձնացված հանգույցներում: Ըստ այդ վերլուծության արդյունքների՝ ընդհանուր էներգասպառման մոտ 60%-ը (59,85%) կազմում են ռեգիստրային հանգույցները, մոտ 33%-ը (32,92%)՝ տրամաբանական հանգույցները, 6,79%-ը՝ մտավոր սեփականության հանգուցները և տակտային ազդանշանային ծառը՝ 0,33%: Նախագծի կատարման արդյունքում ստացվել են տրամաբանական մակարդակի սխեմատեխնիկական նախագիծը (նկ. 2.25):



Նկ. 2.25. ORCA պրոցեսորային ԻՍ-ի տրամաբանական նախագիծը,

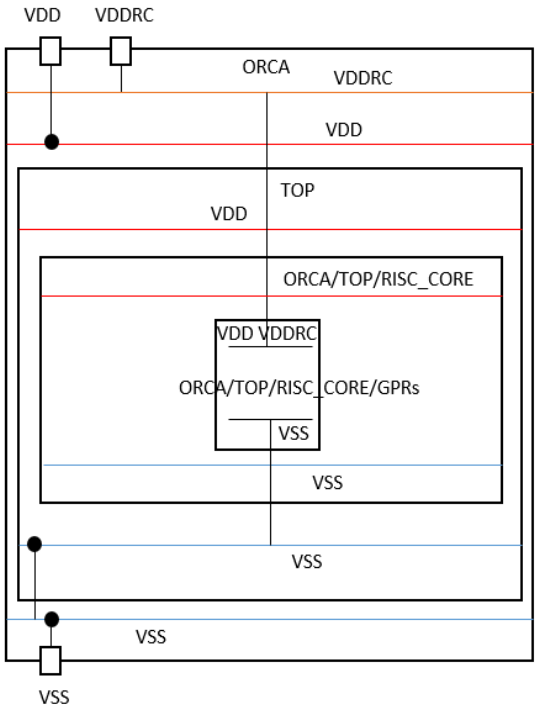
Փորձարարական արդյունքները ստացված ստանդարտ նախագծի պարագայում ապացուցում են, որ ժամանակային պարամետրերը բավարարել են տեխնիկական առաջադրանքը (աղ. 2.10):

Աղյուսակ 2.10

Ստանդարտ նախագծի դեպքում ORCA պրոցեսորի համար ստացված արդյունքները

Տակտային հաճախությունը (*ՄՀց*)	200
Մուտքային ազդանշանների անհրաժեշտ ժամանակային պաշարը (*նվ*)	20,21
Մուտքային ազդանշանների իրական ժամանակային պաշարը (*նվ*)	-20,05
Անհրաժեշտ և իրական ժամանակային պաշարների տարբերությունը (*նվ*)	0,15
Ընդհանուր հզորությունը (*մՎտ*)	88,52
Մտավոր սեփականության հանգույցի մակերեսը (*մկմ*^{2})	16340,796387
Ընդհանուր օգտագործված ստանդարտ բջիջների մակերեսը (*մկմ*^{2})	589303,444805
Ընդհանուր մակերեսը (*մկմ*^{2})	605644,241192

Վերլուծելով ստացված արդյունքները հզորության՝ մակերեսի և ժամանակային հապաղումների տեսանկյունից որոշում է կայացվել ներդնել բազմասնուցման նախագծման մեթոդը, ըստ որի RISC միջուկը ընտրվել է որպես ցածր էներգասպառման տիրույթ: RISC միջուկի իջեցված սնուցման լարումը ընտրվել է 0,78 Վ, իսկ մնացած հատվածների սնուցումը ընտրվել է 1,16 Վ: Ցածր էներգասպառման տիրույթների ընտրության և ներդրման համար կիրառվել է ՄՀՁ-ն՝ որպես թվային ավտոմատ նախագծման բաղկացուցիչ մաս: Նախագծի արդյունքում ստացվել է ՄՀՁ դիագրամը, որը գրաֆիկորեն բնութագրում է տարբեր լարման տիրույթները, դրանց համապատասխան մուտքերը և հատուկ նշանակության տարրերը, որոնք ներդրվել են ցածր էներգասպառման մոտեցման հետ միասին (նկ. 2.26): RISC միջուկը սնուցվում է $V_{\text{սնուցման լարում}}$RC-ով, որի արժեքն է՝ ցածր էներգասպառման տիրույթում՝ 0,78 Վ, պրոցեսորային ԻՍ-ի մյուս հատվածը սնուցվում է ստանդարտ՝ $V_{\text{սնուցման լարում}}$-ով, որի արժեքն է 1,16 Վ: Տվյալ պարագայում օգտագործվել են նաև ԼՄՓ-ներ, որոնք բաշխվել են ցածր լարման տիրույթի ամբողջ սահմանագծով:



Նկ. 2.26. ORCA պրոցեսորային ԻՍ-ի ՄՀՁ դիագրամը բազմասնուցման նախագծման դեպքում

Նախագծման ընթացքում օգտագործվել են հետևյալ սահմանափակումները ORCA պրոցեսորային համակարգի առանձին ենթաբլոկների համար: Մուտք/ելք հանգույցների և օպերատիվ հիշող հանգույցների աշխատանքային հաճախությունը 50 *ՄՀց* է, RISC միջուկի աշխատանքային հաճախությունը՝ 200 *ՄՀց*: Միաժամանակ, ֆիզիկական նախագծման խտությունը սահմանվել է 40%: Թե՛ ստանդարտ նախագծի և թե՛ ցածր էներգասպառման նախագծի սահմանափակումներն ընտրվել են միանման նախագծի կատարման ընթացքում: Բազմասնուցման նախագծման ժամանակ ստացված փորձարարական ընդհանրացված արդյունքները բերված են ստորև (աղ. 2.11):

Աղյուսակ 2.11

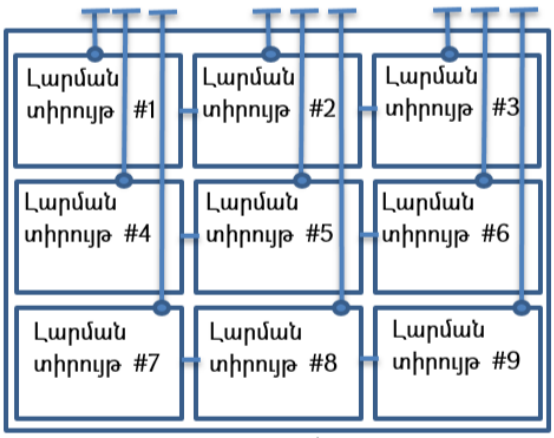
Բազմասնուցմամբ նախագծի դեպքում ORCA պրոցեսորի համար ստացված արդյունքները

Տակտային հաճախությունը (*ՄՀց*)	200
Մուտքային ազդանշանների անհրաժեշտ ժամանակային պաշարը (*նվ*)	20,21
Մուտքային ազդանշանների իրական ժամանակային պաշարը (*նվ*)	-20,20

Անհրաժեշտ և իրական ժամանակային պաշարների տարբերությունը (*նվ*)	0,01
Ընդհանուր հզորությունը (*մՎտ*)	75,46 (-15%)
Մտավոր սեփականության հանգույցի մակերեսը (*մկմ²*)	16340,796387
Ընդհանուր օգտագործված ստանդարտ բջիջների մակերեսը (*մկմ²*)	661980,75374 (+15%)
Ընդհանուր մակերեսը (*մկմ²*)	678321,550135 (+12%)

Ամփոփելով ստացված արդյունքները՝ նկատելի է, որ ընդհանուր հզորությունը կրճատվել է ավելի քան 15%-ով՝ համեմատած ստանդարտ նախագծի: Միաժամանակ, պրոցեսորային ԻՍ-ի մակերեսը մեծացել է մոտ ~12%-ով՝ մեծ մասամբ պայմանավորված ստանդարտ բջիջների տարրերի մակերեսի մեծացմամբ՝ մոտ 15%: Ժամանակային հապաղումներում տարբերությունները չնչին են, ապահովվել է 200 *ՄՀց* հաճախություն՝ ունենալով կորուստ մուտքից ելք հապաղման մեջ՝ 0,5 *նվ*: Ընդհանրացնելով ստացված արդյունքները՝ կարելի է եզրակացնել, որ ներդրված բազմասնուցման նախագծման մեթոդի գործնական կիրառումը RISC ճարտարապետության ORCA պրոցեսորային ԻՍ-ում, հատուկ ստեղծված ՄՀՁ-ի միջոցով, արդյունավետ է էներգասպառման նվազարկման տեսանկյունից, միևնույն ժամանակ առաջացնելով մակերեսի վատթարացում:

Simple RISC պրոցեսորային ԻՍ-ում կիրառված դինամիկ լարման և հաճախության առաջարկված մոտեցմամբ նախագիծ [92]: Ջերմաստիճանից կախվածության հաշվարկման առաջարկվող մոտեցման հետ միասին ինտեգրվել է ջերմաստիճանի մասշտաբավորման հանգույցը՝ որպես մեթոդի բաղկացուցիչ հատված: Դինամիկ լարման և հաճախության առաջարկված մեթոդի հետ համատեղ ներդրվել է նաև բազմասնուցման նախագծման մեթոդի ստատիկ լարման մասշտաբավորման նոր մոտեցում: Առաջարկվել է լարման տիրույթները բաշխել համապատասխան պրոցեսորային ԻՍ-ի ենթահանգույցների՝ ըստ աշխատանքային ծանրաբեռնվածության՝ չսահմանափակելով դրանց քանակը: Դասական




```
############### CREATE POWER DOMAINS ##############
create_power_domain TOP -port VDDH
create_power_domain RISC -port VDDL1
create_power_domain FSM -port VDDL2
create_power_domain GPRs -port VDDL3
create_power_domain USB -port VDDL4
create_power_domain SRAM -port VDDL5
create_power_domain ALU -port VDDL6
create_power_domain Mux2 -port VDDL7
create_power_domain Load/Store -port VDDL8
create_power_domain DEC -port VDDL9
############## ADD PORT STATE INFO ##################
add_port_state VDDH   -state {HV 1.12}
add_port_state VDDL1 -state {LV1 0.107}
add_port_state VDDL2 -state {LV2 0.102}
add_port_state VDDL3 -state {LV3 0.97}
add_port_state VDDL4 -state {LV4 0.92}
add_port_state VDDL5 -state {LV5 0.87}
add_port_state VDDL6 -state {LV6 0.82}
add_port_state VDDL7 -state {LV7 0.77}
add_port_state VDDL8 -state {LV8 0.71}
add_port_state VDDL9 -state {LV8 0.68}
add_port_state risc_sw/out -state {LV 8 0.68}-state {OFF off}
add_port_state VSS -state {GND  0}
```

տարբերակում օգտագործվում են երկու կամ երեք լարման տիրույթներ: Առաջարկվող տարբերակում ընտրվում են N քանակի լարման տիրույթներ N պրոցեսորային ենթահանգույցների համար, և դրանք բաշխվում են ամենաբարձր և ամենացածր լարման պոտենցիալների միջև (նկ. 2.27.ա): Այս դեպքում անհրաժեշտություն չկա նախագծելու առանձին ԼՄՓ-ներ լրացուցիչ լարման տիրույթների համար, քանի որ օգտագործված ԼՄՓ-ները, որոնք անցում են կատարում ամենաբարձր պոտենցիալից դեպի ամենացածր պոտենցիալը կամ հակառակը, կարող են կիրառելի լինել նաև այդ լարումների միջակայքում ընդգրկված տիրույթներում: Այս մոտեցումը հնարավորություն է տալիս կատարել էներգասպառման, արագագործության կառավարում և նվազարկում՝ հիմնվելով պրոցեսորային ԻՍ-ի աշխատանքային ռեժիմների վրա, օգտագործելով ՄՀՁ նկարագրությունը (նկ. 2.27.բ) և հզորության կառավարման հանգույցը:

ա) բ)

Նկ. 2.27. ա) Առաջարկված բազմասնուցման նախագծման կառուցվածքային պատկերը, բ) առաջարկված բազմասնուցման մեթոդով մշակված ՄՀՁ նկարագրության հատվածը

Նախագիծն իրականացվել է՝ օգտագործելով տրամաբանական (Design Compiler) և ֆիզիկական (IC Compiler) ավտոմատ նախագծման ծրագրային միջոցները: Ջերմաստիճանի վերծանման հանգույցը նախագծվել է առանձնացված կարգով (Custom Compiler) և այնուհետև ինտեգրվել է պրոցեսորային ԻՍ-ում: Ընտրված մուտքային սահմանափակումները համարժեք են դասական տարբերակով նախագծի և նախորդ տարբերակում կիրառվածներին՝ ապահովելու համար ստացված արդյունքների համեմատումը նույն պայմանների պարագայում:

Կատարվել է առաջարկված նախագծի փորձարարական տվյալների համեմատություն դասական, նախորդ տարբերակում քննարկված և գրականությունից ընտրված տարբերակների միջև (աղ. 2.12):

Աղյուսակ 2.12

Առաջարկված մոտեցմամբ, դասական բազմասնուցման, գրականությունից հայտնի և նախորդ քննարկված տարբերակների միջև արդյուքների համեմատությունը simple RISC պրոցեսորի դեպքում

Պարամետրերը	Դասական տարբերակով	Նախորդ առաջարկված տարբերակ	Առաջարկվող տարբերակով	ԴԼՀՄ հանգույց՝ ընտրված գրակ.
Տակտային հաճախություն (*ՄՀg*)	2	1,9	2	1…123
Ընդհանուր հզորություն (*մլՎտ*)	68,55	39,73 (-42%)	31,53 (-54%)	45
Ընդհանուր մակերես (*մկմ²*)	174527,6	180985,12 (+3,7%)	195463,92 (+8%)	185689,85

Առաջարկված մոտեցմամբ նախագծի պարագայում ստացվել է 54%-ով նվազեցում՝ համեմատած դասական տարբերակի, 12%-ով՝ ի համեմատ նախորդ քննարկված տարբերակի և ~30%-ով՝ ի համեմատ առկա ԴԼՀՄ տարբերակի: Մակերեսի դեպքում առաջարկվող տարբերակն ունի վատագույն ցուցանիշը, և ստացվել է 8%-ով մեծացում՝ համեմատած նախորդ տարբերակի: Բոլոր տարբերակներում էլ հնարավոր է եղել բավարարել ժամանակային սահմանափակումները՝ համապատասխան տեխնիկական առաջադրանքին:

Այսպիսով, ամփոփելով դինամիկ լարման և հաճախության առաջարկված եղանակը՝ կարելի է եզրակացնել, որ այն արդյունավետ էր պրոցեսորային ԻՍ-երի էներգասպառման նվազարկման տեսանկյունից՝ ապահովելով լարման և հաճախության արժեքների փոփոխության ընթացքում ջերմաստիճանային գործակցի ազդեցության հաշվարկը: ԿԲ-ում ԻՍ-ի զբաղեցրած մակերեսի պարագայում ստացվել է մեծացում, որը պայմանավորված է ներդրված ջերմաստիճանի մասշտաբավորման հանգույցով և լրացուցիչ ԼՄՓ-ներով: Առաջարկվող տարբերակը կարող է լայնորեն կիրառվել այն պրոցեսորային ԻՍ-երում, որտեղ մակերեսի մեծացումը մեծ հաշվով կարելի է ընդունելի համարել:

Եզրակացություններ

1. Առաջարկվել է RISC ճարտարապետությամբ պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկման եղանակ, որում ցածր հզորության նախագծման մեթոդների համադրության շնորհիվ՝ առկա լուծումների համեմատ ապահովվում է էներգասպառման ժամանակակից պահանջները բավարարող չափով փոքրացում:

2. Մշակվել է հզորության շրջափակման մեթոդի տարատեսակ, որի դեպքում ցածր էներգասպառման ռեժիմում վիճակի պահպանման և վերականգնման ֆունկցիայի՝ պրոցեսորային ինտեգրալ սխեմայի ներսից դեպի եզրագծեր տեղափոխման հաշվին ստացվում է էներգասպառման ստատիկ բաղադրիչի զգալի նվազեցում՝ կիսահաղորդչային բյուրեղի վրա ինտեգրալ սխեմայի զբաղեցրած մակերեսի թույլատրելի մեծացման պարագայում:
3. Հզորության շրջափակման մեթոդում առաջարկված մոտեցման կիրառությունն ապահովելու նպատակով կառուցվել են ելքային ազդանշանի 0,5…1,5 *ԳՀg* հաճախականային միջակայքում աշխատող, վիճակի պահպանման և վերականգնման հնարավորությամբ հատուկ լարման մակարդակի փոխակերպիչներ, որոնք լրացուցիչ լիցքաթափում իրականացնող տրանզիստորային հանգույցի հաշվին ապահովում են զգալիորեն ցածր էներգասպառում, բարձր արագագործություն և պահպանում են համակարգի կայունությունը՝ բացառելով մետակայուն վիճակում հայտնվելու հնարավորությունը:
4. Առաջարկվել է պրոցեսորային ինտեգրալ սխեմաների դինամիկ լարման և հաճախության մասշտաբավորման եղանակ, որի դեպքում ջերմաստիճանի կախվածությունը հաշվի առնելով՝ ստացվել է էներգասպառման էական նվազարկում, միաժամանակ մեծացնելով կիսահաղորդչային բյուրեղում ինտեգրալ սխեմայի զբաղեցրած մակերեսը:

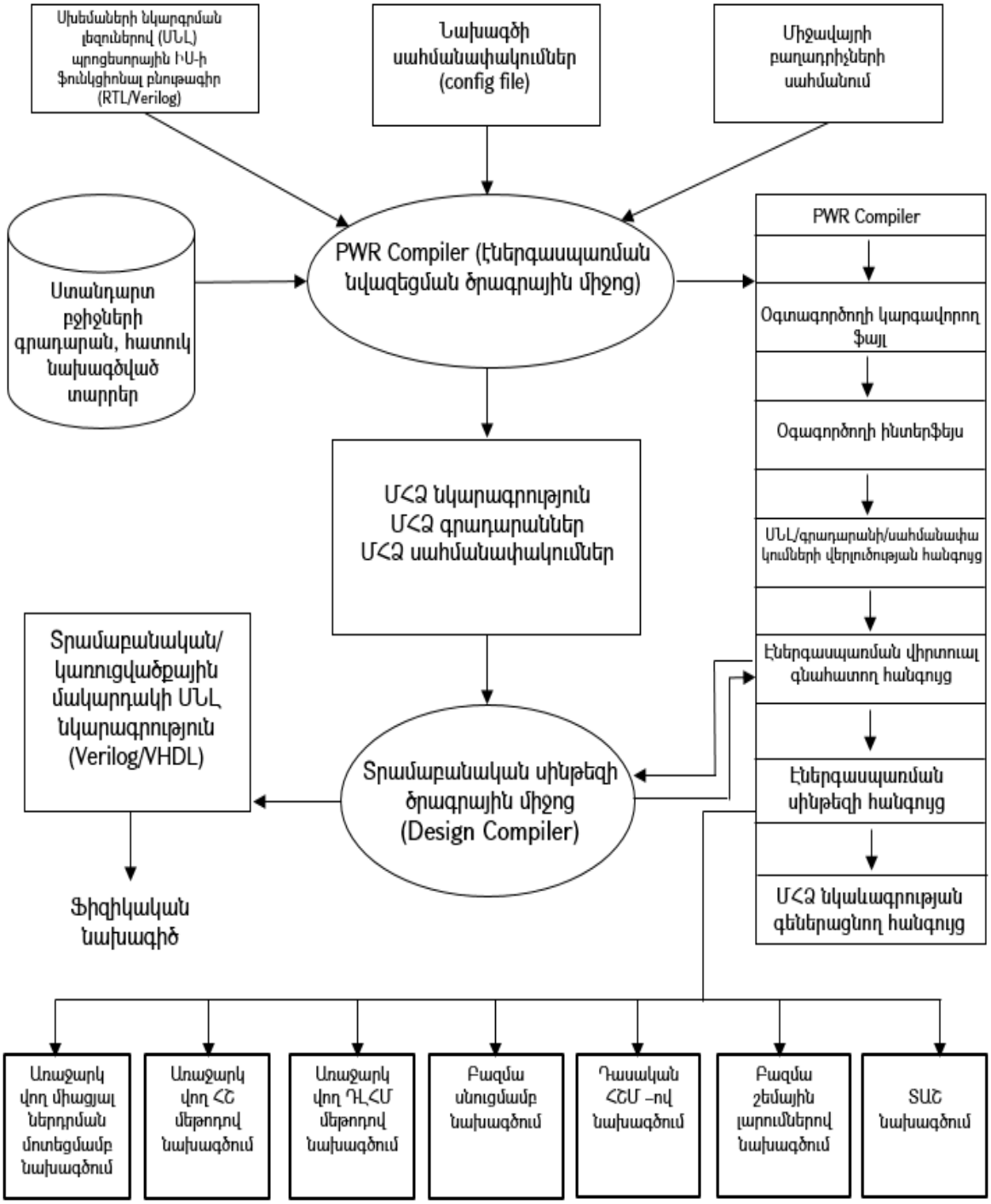
ԳԼՈՒԽ 3. ՊՐՈՑԵՍՈՐԱՅԻՆ ԻՆՏԵԳՐԱԼ ՍԽԵՄԱՆԵՐԻ ԷՆԵՐԳԱՍՊԱՌՄԱՆ ՆՎԱԶԵՑՄԱՆ ԾՐԱԳՐԱՅԻՆ ՄԻՋՈՑԻ ՆԿԱՐԱԳՐՈՒԹՅՈՒՆԸ ԵՎ ՕԳՏԱԳՈՐԾՈՒՄԸ

Նախորդող գլուխներում առաջարկված մեթոդները և մոտեցումները ներդրված են մշակված PWR Compiler ծրագրային միջոցում: PWR Compiler ծրագրային միջոցն իրագործվել է C [96], Qt [97], python [98] ծրագրավորման և սկրիպտավորման լեզուներով և նախատեսված է Windows 7 [99] և ավելի նոր օպերացիոն համակարգերով օգտագործման համար: Ծրագրային միջոցն իր աշխատանքի ընթացքում օգտագործում է Սինոփսիս ընկերության ծրագրային միջոցներից Design Compiler-ը [100]:

PWR Compiler ծրագրային միջոցը նախատեսված է առաջարկվող երեք մոտեցումների ինտեգրումն ապահովելու համար. դրանք են՝ ցածր հզորությամբ նախագծման մեթոդների միացյալ համադրման մոտեցումը և հանգույցը, հզորության շրջափակմամբ առաջարկված մեթոդը և սխեմատեխնիկական լուծումը, դինամիկ լարման և հզորության մասշտաբավորման առաջարկված մեթոդը: Ծրագրային միջոցը կարգավորվում է մուտքային ձևակերպման ֆայլի միջոցով, որը կարող է պարունակել ծրագրային միջոցով կիրառվող տեխնիկական սահմանափակումներ: Պրոցեսորային

ԻՍ-երի նախագծման առանձնահատկություններից ելնելով՝ ծրագրային միջոցը համընթաց է գործում թվային ավտոմատ նախագծման գործընթացին և ավտոմատացնում է դրա ցածր հզորության նախագծման բաղադրիչը: Ծրագրային միջոցի հիմնական գործառույթն է ցածր էներգասպառման լավագույն մեթոդի ընտրությունը տվյալ աշխատանքում առաջարկվածներից և արդեն իսկ հայտնի մեթոդներից՝ հաշվի առնելով ընտրված պրոցեսորային ԻՍ-ի առանձնահատկությունները և բնութագրերը: Ծրագրային միջոցում ինտեգրված են առաջարկված պրոցեսորային ԻՍ-ում կիրառության համար մշակված հզորության կառավարման հանգույցը, վիճակը պահպանող և վերականգնող ԼՄՓ-ն և ջերմաստիճանի վերծանման հանգույցը (նկ. 3.1):

PWR Compiler ծրագրային միջոցի կիրառումը կրճատում է ցածր էներգասպառման նախագծման ժամանակը 7…12 անգամ:



Նկ. 3.1. Առաջարկվող PWR Compiler ծրագրային միջոցի կառուցվածքը

Մշակված ծրագրային միջոցի աշխատանքի ընթացքում էներգասպառման գնահատումը ճշգրիտ իրականացնելու համար օգտագործվում է Design Compiler-ի հզորության կանխատեսման/գնահատման ֆունկցիան: Համակարգի արագագործության ապահովման նպատակով ստեղծված է հնարավորություն՝ կատարելու հզորության գնահատումը PWR Compiler-ի ներսում՝ նվազեցնելով ճշգրտության աստիճանը:

3.1. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման ծրագրային միջոցի կառուցվածքը և աշխատանքի սկզբունքը

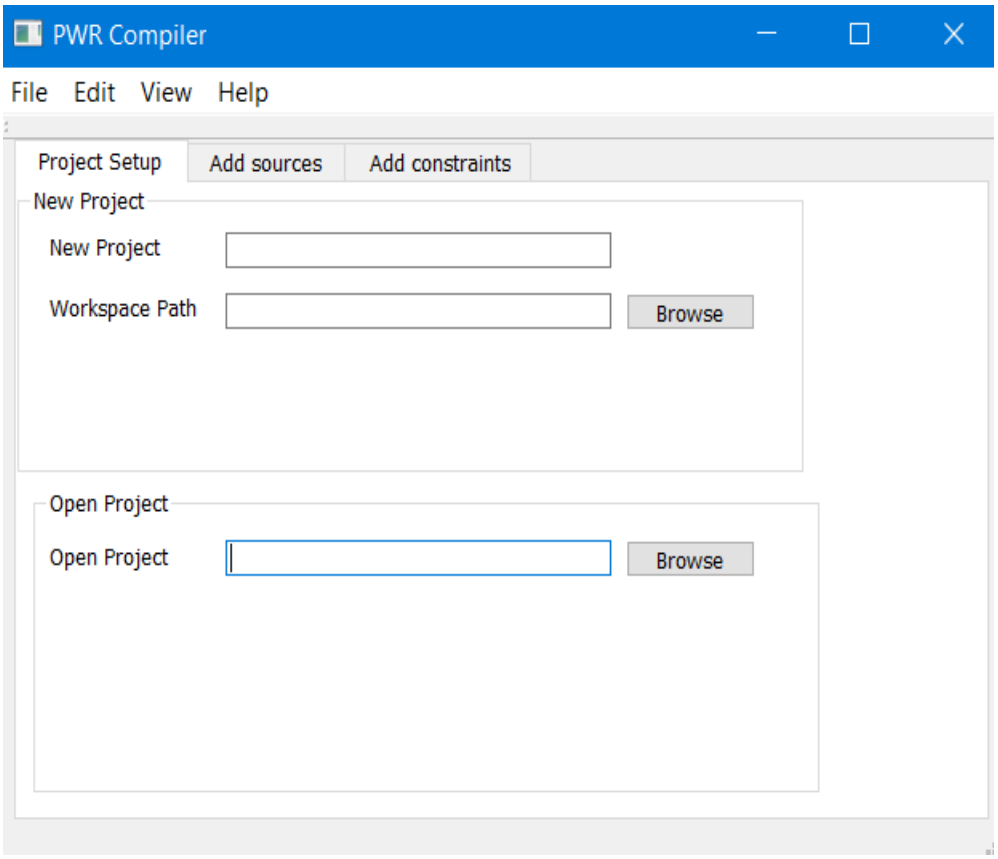
Պրոցեսորային ԻՍ-ի ցածր էներգասպառմամբ նախագծումը բարդ հիմնահարց է, որը պահանջում է բավականին երկար ժամանակ իրագործման համար և խորքային գիտելիքներ ցածր հզորությամբ նախագծման մեթոդների մասին: Անհրաժեշտություն է առաջանում որոշել ցածր էներգասպառման արդյունավետ մեթոդի ընտրությունը՝ կախված պրոցեսորային ԻՍ-ի ճարտարապետության առանձնահատկություններից: PWR Compiler ծրագրային միջոցի հիմնական նպատակն է ավտոմատացված կերպով վերլուծել և որոշել ցածր էներգասպառման նախագծման հայտնի և առաջարկված որևէ մեթոդի ներդրումը կամ մի քանի մեթոդների համադրումը: Այնուհետև, համապատասխան ընտրված մեթոդի՝ ծրագրային միջոցը գեներացնում է միասնական հզորության ձևաչափի նկարագրություն, որն անհրաժեշտ է տրամաբանական սինթեզի ծրագրային միջոցին՝ ցածր էներգասպառմամբ նախագիծը տրամաբանական մակարդակից դեպի կառուցվածքայինը անցումն ապահովելու համար: PWR Compiler-ի ներդրմամբ նվազարկվում է նախագծողի կողմից կատարվող աշխատանքը, որը մինչ այդ կատարվում էր առանց ավտոմատ նախագծման ծրագրային միջոցի և կարող էր տևել մինչև մի քանի օր: Ծրագրային միջոցի ներդրման պարագայում հնարավոր է այդ գործողությունը կատարել առավելագույնը մեկ ժամվա ընթացքում՝ կախված պրոցեսորային ԻՍ-ի բարդության աստիճանից և ցածր էնրգասպառման մեթոդի ընտրությունից:

PWR Compiler ծրագրային միջոցը նախագծման տեսանկյունից կազմված է 6 հիմնական փուլերից և դրանց համապատասխան պատուհաններից ՝

1. Setup input files.
2. User interface.
3. Input files analyzer.
4. Virtual power estimator.
5. Low power synthesizer.
6. UPF [101] netlister.

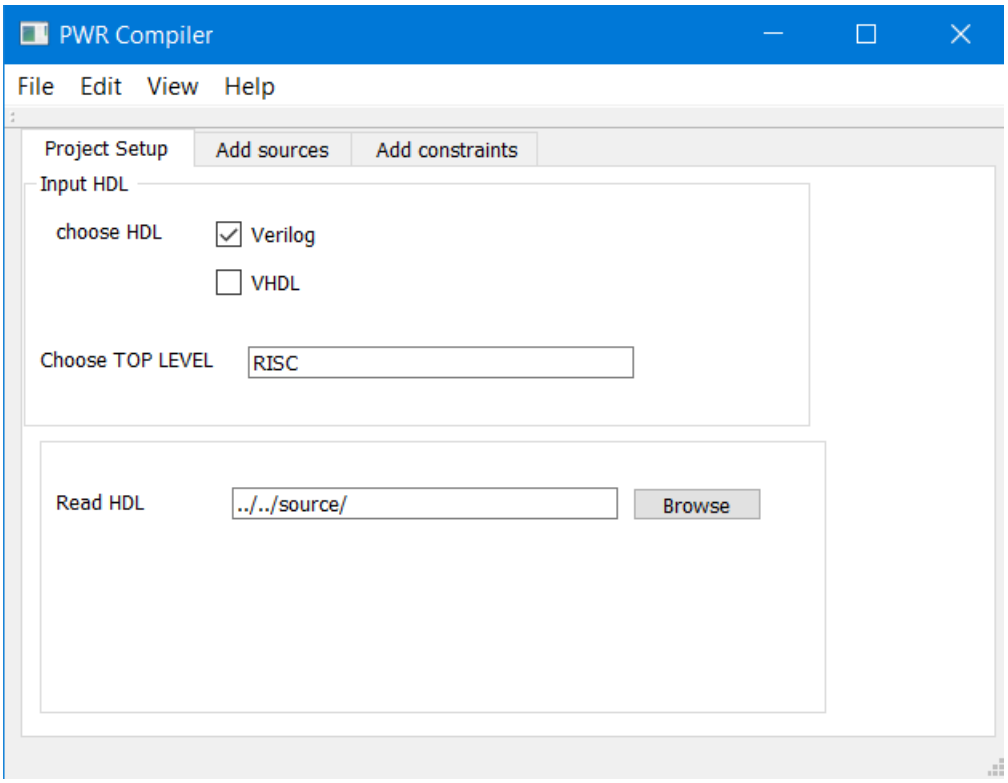
Ծրագրի առաջին փուլը սկսվում է <<setup input files>> հրամանային պատուհանների հանգույցից: Այս փուլում մուտքագրվում են հետևյալ բաղադրիչները՝

- նախագծի անվան սահմանումը, տեղանքի ընտրությունը, արդեն գոյություն ունեցող նախագծի վերաթողարկումը (նկ. 3.2),
- պրոցեսորային ԻՍ-ի ՍՆԼ նկարագրությունը (Verilog [102] / VHDL [103], RTL),
- նախագծի սահմանափակումները (constraints),
- միջավայրի բաղադրիչների սահմանումը (Environment setup),
- ստանդարտ բջիջների և հատուկ տարրերի գրադարանը (std .lib/.db):



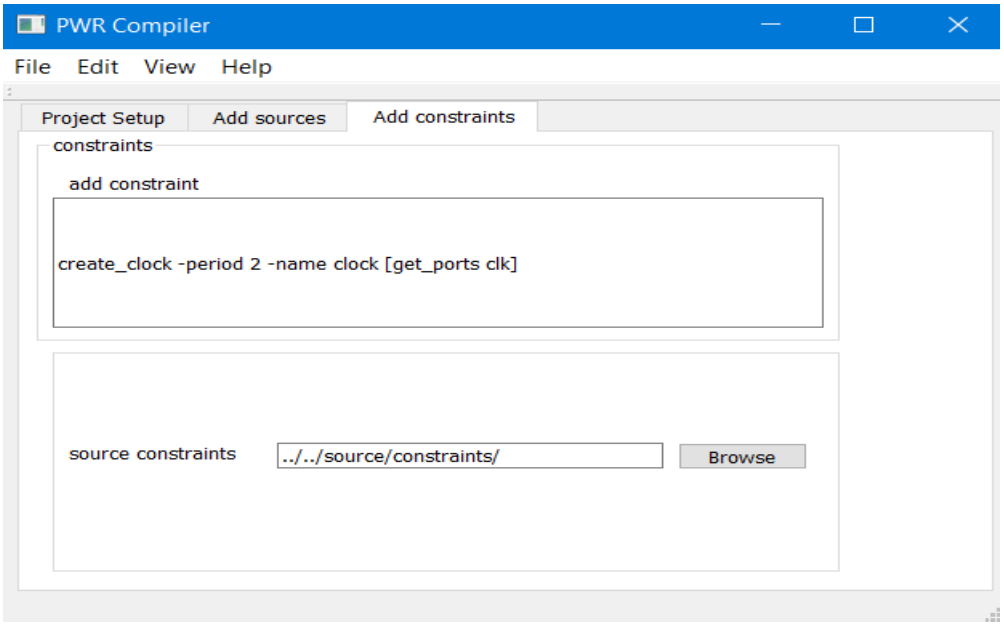
Նկ. 3.2. Առաջարկվող PWR Compiler ծրագրային միջոցի մուտքային պատուհանը

<<choose HDL>> դաշտում հնարավորություն կա ընտրելու երկու HDL-ներից մեկը՝ [Verilog] կամ [VHDL]: Բազմահիերարխիկ նախագծերի դեպքում հնարավորություն կա ընտրելու ամենաբարձր մակարդակը՝ <<choose TOP Level>> դաշտում նշելով բարձր մակարդակի module-ի անունը: <<read HDL>> դաշտը հնարավորություն է տալիս ընտրելու և մուտքագրելու մեկ կամ մի քանի HDL նկարագրություններ՝ համաձայն նախապես սահմանված տեսակի [Verilog/VHDL] (նկ. 3.3):



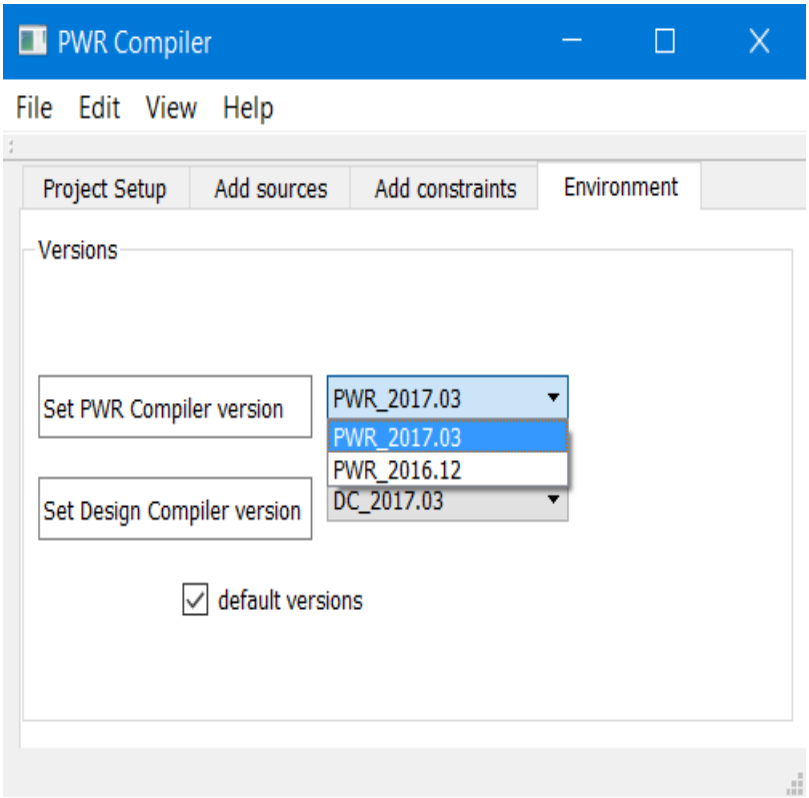
Նկ. 3.3. Առաջարկվող PWR Compiler ծրագրային միջոցի մուտքային բաղադրիչների սահմանումը

<<source constraints>> դաշտում հնարավորություն կա մուտքագրելու կամ ընդհանուր սահմանափակումները մեկ ֆայլով (tcl ձևաչափով), կամ առանձնացված սահմանափակումները <<Interactive GUI>>-ի միջոցով (նկ. 3.4):



Նկ. 3.4. Առաջարկվող PWR Compiler ծրագրային միջոցում սահմանափակումների սահմանումը

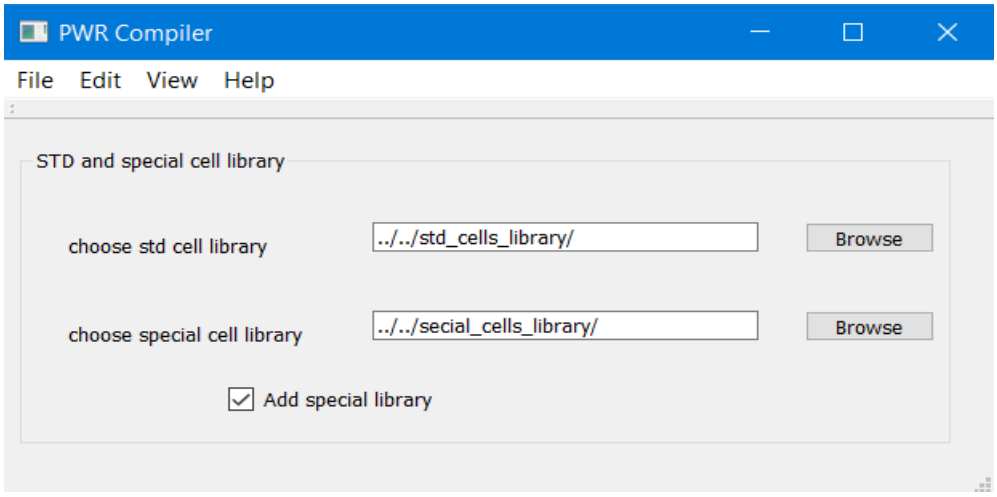
<<Environment setup>> դաշտում հնարավորություն կա ընտրելու PWR Compiler և Design Compiler ծրագրային միջոցների հասանելի տարբերակներից որևէ մեկը: Տվյալ դաշտերի լրացումները պարտադիր չեն, և դրանց բաց թողնելու պարագայում ծրագրային միջոցը լռելյայն ընտրում է նախապես սահմանված տարբերակները (նկ. 3.5):



Նկ. 3.5. Առաջարկվող PWR Compiler և Design Compiler ծրագրային միջոցների աշխատանքային տարբերակների ընտրությունը

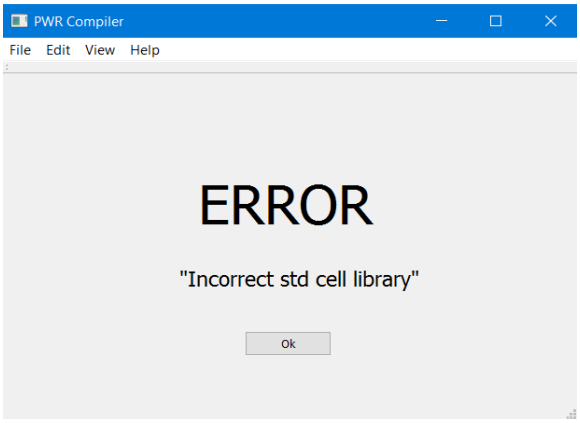
<<choose std cell library>> դաշտում կարելի է ընտրել ստանդարտ բջիջների գրադարանի <<.db>> բաղադրիչը, որը նախատեսված է ծրագրային միջոցների համար: Պատուհանում <<optional>> հնարավորությամբ կարելի է մուտքագրել հատուկ նախագծված տարրերի գրադարաններ, ինչպիսիք են՝ արագագործ ԼՄՓ-ները, ՎՊՎ ԼՄՓ-ները, հզորության կառավարման և ջերմաստիճանի վերծանման հանգույցների <<.db>> բաղադրիչները (նկ. 3.6): Որպես կանոն, հատուկ գրադարանների անհրաժեշտությունն առաջանում է այն դեպքում, երբ հիմնական

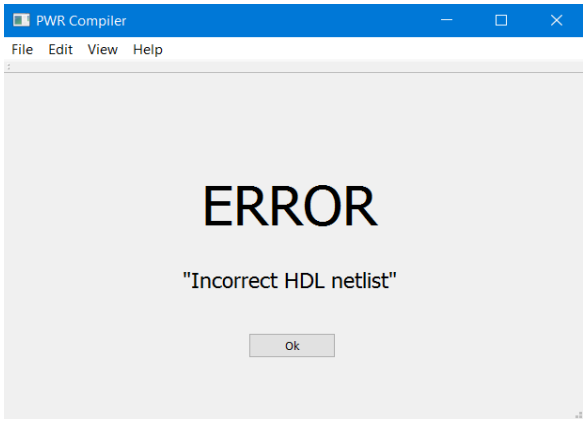
գրադարանները չեն պարունակում ցածր էներգասպառման նախագծման համար անհրաժեշտ տարրեր:

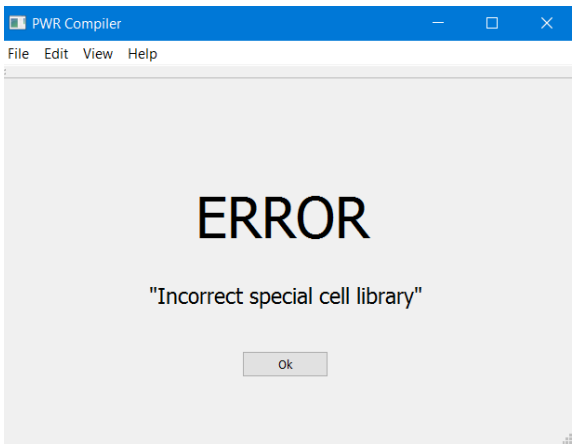


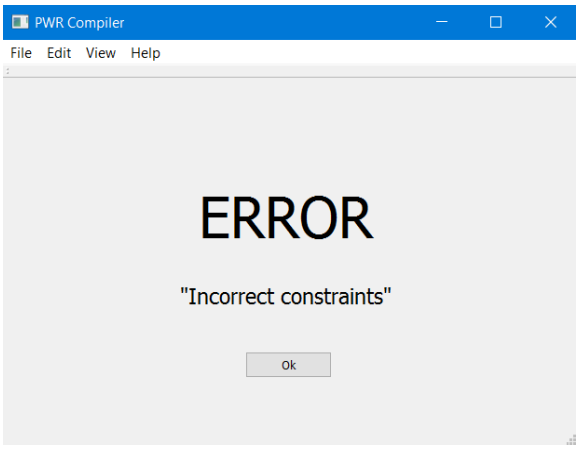
Նկ. 3.6. Առաջարկվող PWR Compiler ծրագրային միջոցի ստանդարտ բջիջների գրադարանի սահմանումը

Համակարգի մուտքային պարտադիր բաղադրիչներից ցանկացածի սխալ լրացվելու կամ դատարկ լինելու պարագայում ծրագրային միջոցը թողարկում է հատուկ հաղորդագրություն՝ <<Error>> գրությամբ (նկ. 3.7):







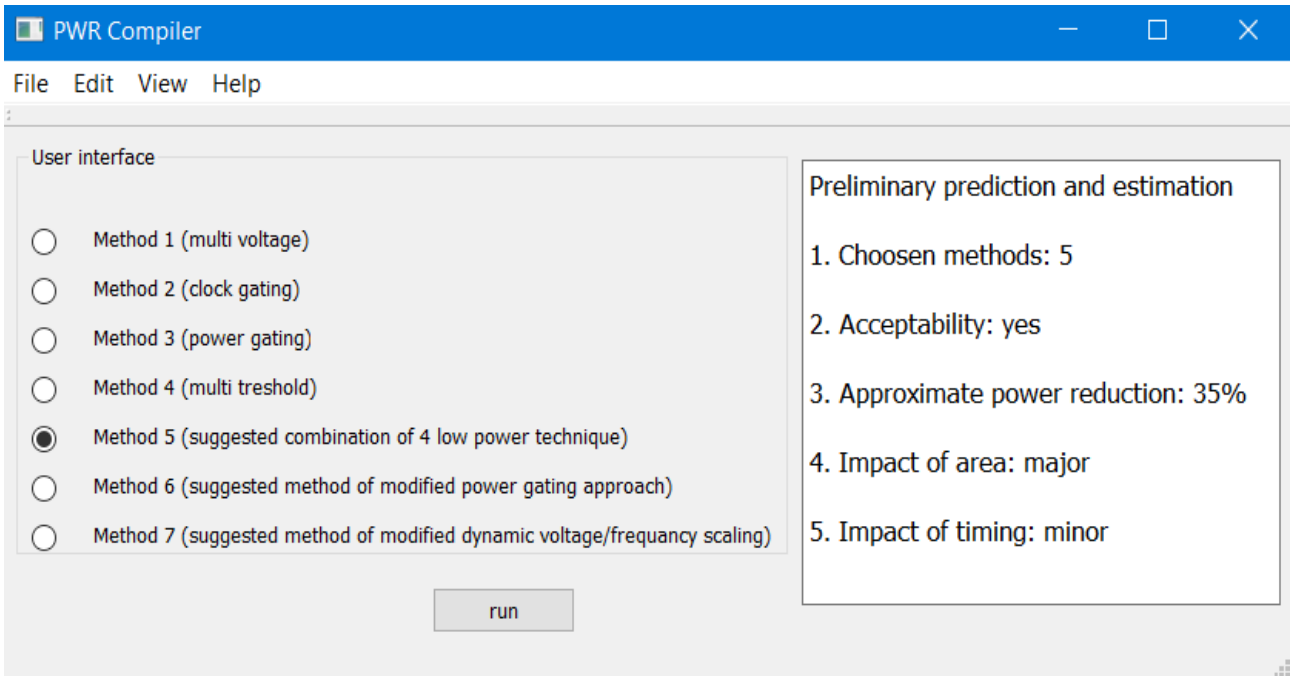


Նկ. 3.7. Առաջարկվող PWR Compiler ծրագրային միջոցում սխալանքի պատուհանները

Հաջորդ պատուհանը <<User interface>>-ն է, որը նախատեսված է նախագծողի կողմից հատուկ նշումների կատարման համար: Այս պատուհանում հնարավորություն է տրվում կատարելու նշումներ՝ ցածր էներգասպառման որ մեթոդի կիրառումն է գերադասելի նախագծողի կողմից: Տվյալ պարագայում նախագծողի կողմից կատարված նշումները ծրագրային միջոցն ընդունում է որպես առաջնային ցանկալի տարբերակներ, բայց դրանք պարտադիր չեն: Նախագծողը կարող է ընտրել արդեն իսկ ծրագրային միջոցում ներդրված մեթոդներից որևէ մեկը կամ դրանց համադրությունը (նկ. 3.8).

- <<Method 1>> կոճակ. բազմասնուցման մեթոդ,
- <<Method 2>> կոճակ. տակտային ազդանշանի շրջափակման մեթոդ,
- <<Method 3>> կոճակ. հզորության շրջափակման մեթոդ,
- <<Method 4>> կոճակ. բազմաշեմային մեթոդ,
- <<Method 5>> կոճակ. առաջարկվող ցածր էներգասպառման տարբեր մեթոդների համադրման մոտեցումներ,
- <<Method 6>> կոճակ. առաջարկվող մոտեցումներ հզորության շրջափակման մեթոդում,
- <<Method 7>> կոճակ. առաջարկվող մոտեցումներ դինամիկ լարման և հզորության մասշտաբավորման մեթոդում:

Այն դեպքում, երբ հնարավորություն չի լինում կատարելու առաջադրված նշումները, դրանք հակասում են տվյալ պրոցեսորային ԻՍ-ի միկրոճարտարապետությանը: Ծրագրային միջոցը համապատասխան նախազգուշացնող ազդակ է ուղարկում և ցուցադրում այն ընթացիկ պատուհանում: Ծրագրային միջոցը հնարավորություն է ընձեռում նախազգուշացնող ազդակը ստանալուց հետո ձևափոխել մուտքային նշումները՝ ապահովելու համար աշխատանքի շարունակականությունը և առաջադրված ճշգրիտ նշումների կատարումը: Ծրագրային միջոցն արգելափակում է երկրորդ անգամ նույն նշումների հաջորդականության մուտքագրումը՝ խուսափելու համար կրկնություններից և ժամանակի կորուստներից:

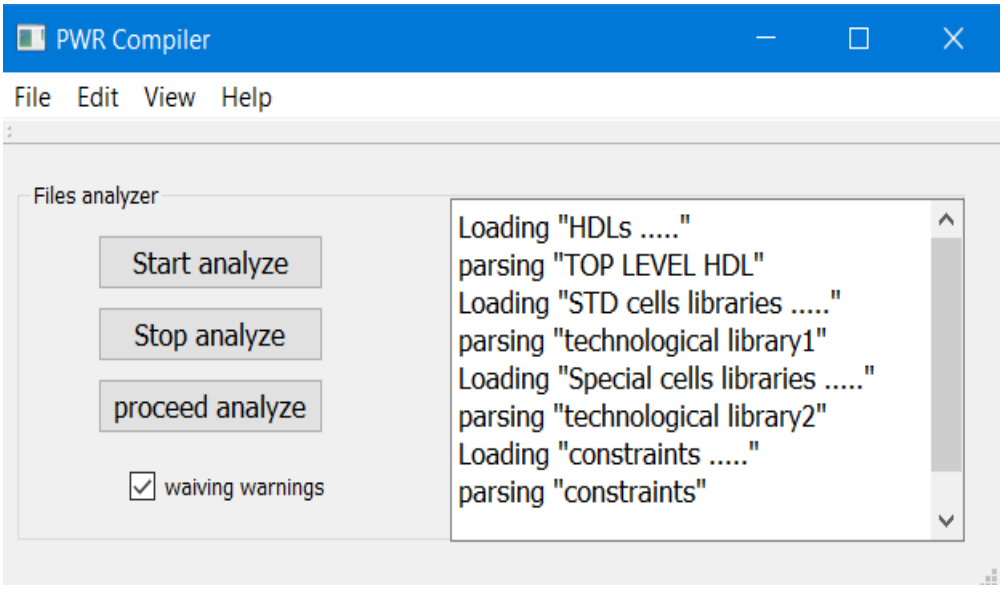


Նկ. 3.8. Առաջարկվող PWR Compiler ծրագրային միջոցում նախագծողի կողմից կատարվող նշումները

Ծրագրի երրորդ փուլը <<Input files analyzer>>-ն է, որի հիմնական նպատակն է մուտքային բաղադրիչների վերլուծության իրականացումը՝ հաշվի առնելով դրանց ցածր էներգասպառման հնարավորությունները: Հիմնական բաղադրիչը, որը առաջնահերթ ենթարկվում է վերլուծության, ՍՆԼ նկարագրությունն է, քանի որ այն բնութագրում է պրոցեսորային ԻՍ-ի միկրոճարտարապետությունը, որն ունի առաջնային նշանակություն ցածր էներգասպառման մեթոդների ինտեգրման տեսանակյունից:

Մուտքային փաթեթի հաջորդ վերլուծությունը սահմանափակումներն են, որոնք կարող են լինել անհատական կամ միասնական տարբերակով: Սահմանափակումների առաջնային վերլուծության նպատակն է գնահատել դրանց համադրությունը մուտքային ՍՆԼ-ի, ցածր էներգասպառման մեթոդների և տեխնոլոգիական բաղադրիչների հետ: Այս պատուհանում <<start analyze>> հրամանի կոճակով ազդարարվում է վերլուծության սկիզբը: <<stop analyze>> հրամանի կոճակով ազդարարվում է վերլուծության կանգնեցում: <<proceed analyze>> հրամանի կոճակով հնարավոր է աշխատանքը դադարացնելուց հետո ապահովել գործընթացի շարունակականությունը: Տվյալ փուլում կատարվում է ցածր էներգասպառման

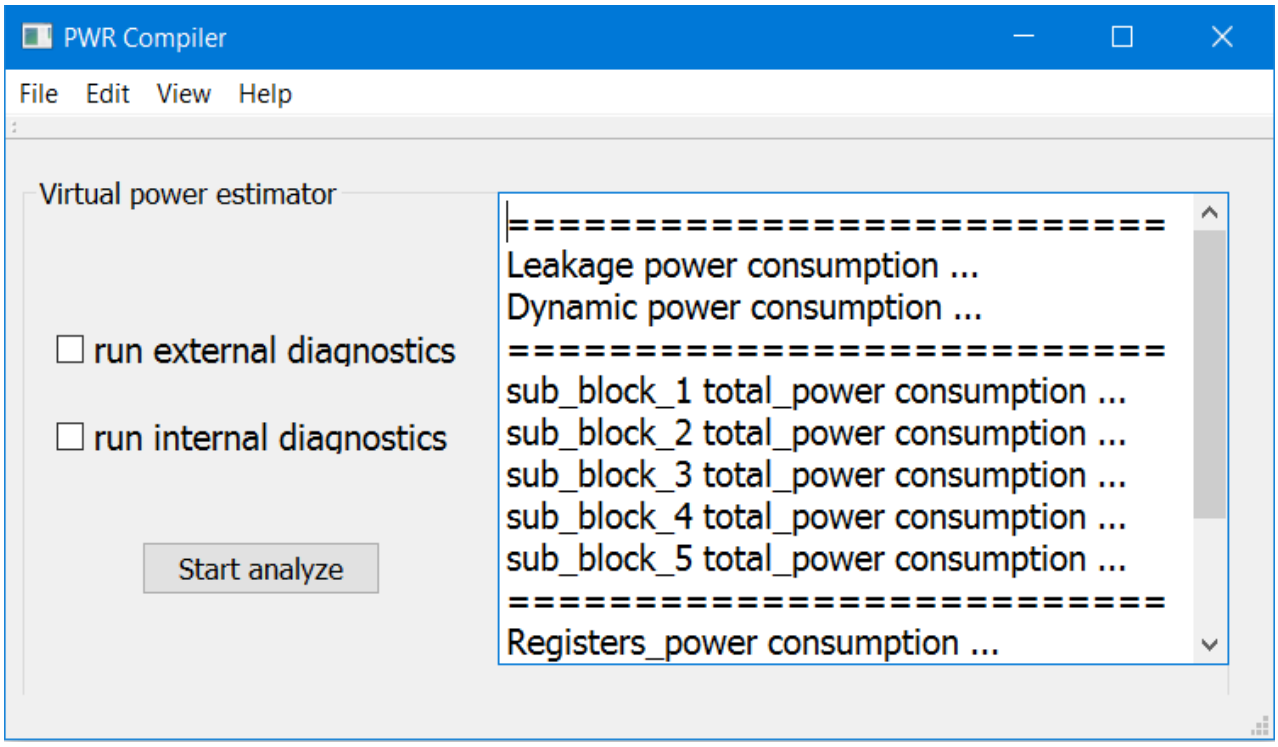
մեթոդների՝ ըստ հանգույցների և դրանց պարամետրերի հնարավոր կիրառման նախնական գնահատում (նկ. 3.9):



Նկ. 3.9. Առաջարկվող PWR Compiler ծրագրային միջոցում աշխատանքի կասեցման և վերաթողարկման պատուհանը

Էներգասպառման վիրտուալ գնահատում իրականացնող հանգույցը՝ <<Virtual power estimator>>, կատարում է ընդհանուր հզորության կանխատեսվող արժեքների վերծանում և դրանց բնույթի բացահայտում: Այս փուլում մշակված ծրագրային միջոցը արդյունավետ էներգասպառման գնահատում կատարելու համար օգտագործում է Design Compiler-ը: Այդ վերլուծությունը կատարելու համար անհրաժեշտ է սեղմել <<run external diagnostics>> կոճակը: Պատուհանում բերվում են ընդհանուր հզորության արժեքները՝ դրա առանձնացված ստատիկ և դինամիկ բաղադրիչներով, ինչպես նաև էներգասպառման բաշխվածությունը՝ ըստ պրոցեսորային ԻՍ-ի ենթահանգույցների: Բերված են նաև ընդհանրացված արժեքներ՝ ըստ տրամաբանական տարրերի և ռեգիստրների վրա բաշխված էներգասպառման: Առանձնացված կարգով բերված է նաև այս կամ այն մեթոդի ներդրման արդյունքում կիրառված լրացուցիչ տարրերի, մասնավորապես՝ լարման մակարդակի փոխակերպիչների, հզորության կառավարման և ջերմաստիճանի վերծանման հանգույցների էներգասպառումը: Հնարավոր է նաև կատարել էներգասպառման վերլուծություն PWR Compiler ծրագրային միջոցով, առանց արտաքին աղբյուրների ներգրավվման: Դրա համար անհրաժեշտ է սեղմել <<run internal diagnostics>> կոճակը, որի արդյունքում կրճատվում է ընդհանուր վերլուծության ժամանակը մոտ

20%-ով, բայց, միևնույն ժամանակ, նվազում է նաև ճշգրտության աստիճանը մոտ 15%-ով (նկ. 3.10):



Նկ. 3.10. Առաջարկվող PWR Compiler ծրագրային միջոցի վիրտուալ էներգասպառման գնահատումը

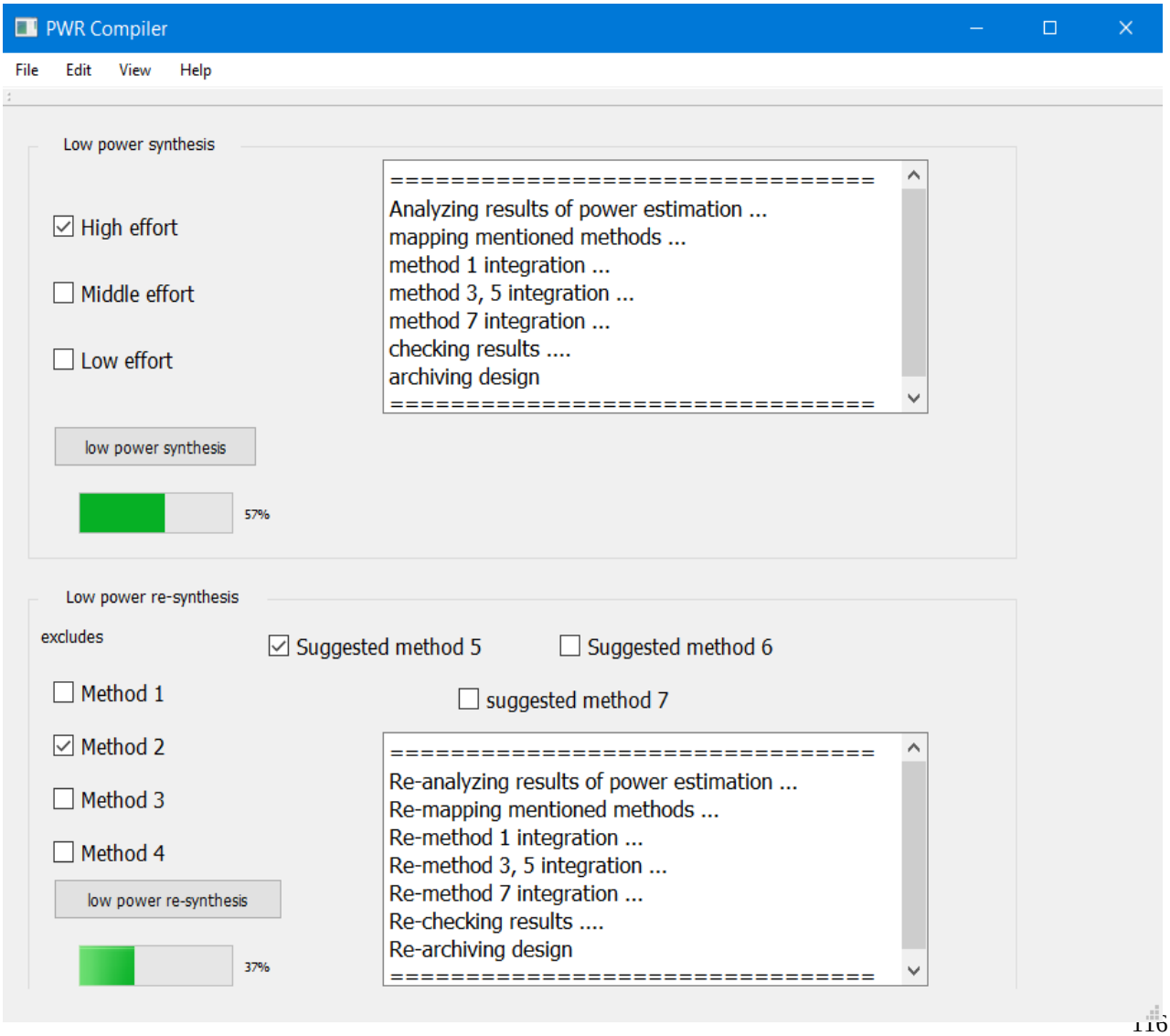
PWR Compiler ծրագրային միջոցում վիրտուալ էներգասպառման գնահատումը շատ կարևոր է և հետագայում հիմք է հանդիսանում՝ կատարելու համար առաջարկված ցածր էներգասպառմամբ մեթոդների ինտեգրումը պրոցեսորային ԻՍ-երում:

Էներգասպառման նվազարկմանը միտված սինթեզը՝ <<Low power synthesizer>>, PWR Compiler ծրագրային միջոցի հիմնական հանգույցներից մեկն է, որը պատասխանատու է ցածր էներգասպառման ընտրված մեթոդի ինտեգրման համար: Տվյալ հանգույցում կատարվում է արդեն իսկ ստացված էներգասպառման և արագագործության վերլուծության տվյալների ամփոփում, և ի սկզբանե փորձ է արվում՝ կիրառելու նախագծողի կողմից նշումների բաժնում մատնանշված ցանկալի ցածր էներգասպառման մեթոդը կամ մեթոդները և դրանց համադրումը: Այնուհետև շարունակվում են նվազարկումը և բալանսավորումը՝ գտնելու համար հնարավոր ամենաարդյունավետ տարբերակը: Սինթեզի կատարման համար անհրաժեշտ է սեղմել

<<start low power synthesis>> կոճակը: Այս հրամանին զուգահեռ գոյություն ունի նաև հնարավորություն՝ ընտրելու կիրառվող ջանքի աստիճանը, որը չի հանդիսանում պարտադիր կիրառման դաշտ՝

- ուժեղ ջանք {high effort},
- միջին ջանք {middle effort},
- թույլ ջանք {low effort}:

Այն տարբերակում, երբ այդ դաշտը լրացված չէ, ծրագրային միջոցը լռելյայն ընտրում է միջին ջանքի կիրառմամբ տարբերակը, որը բալանսավորում է բարձր և ցածր ջանքերով նախագծման առավելությունները և թերությունները (նկ. 3.11): Դրանք միմյանցից տարբերվում են հիմնականում ակտիվ աշխատաժամանակի սահմանափակումով և կատարված աշխատանքի բավարար որակական հատկանիշներով:



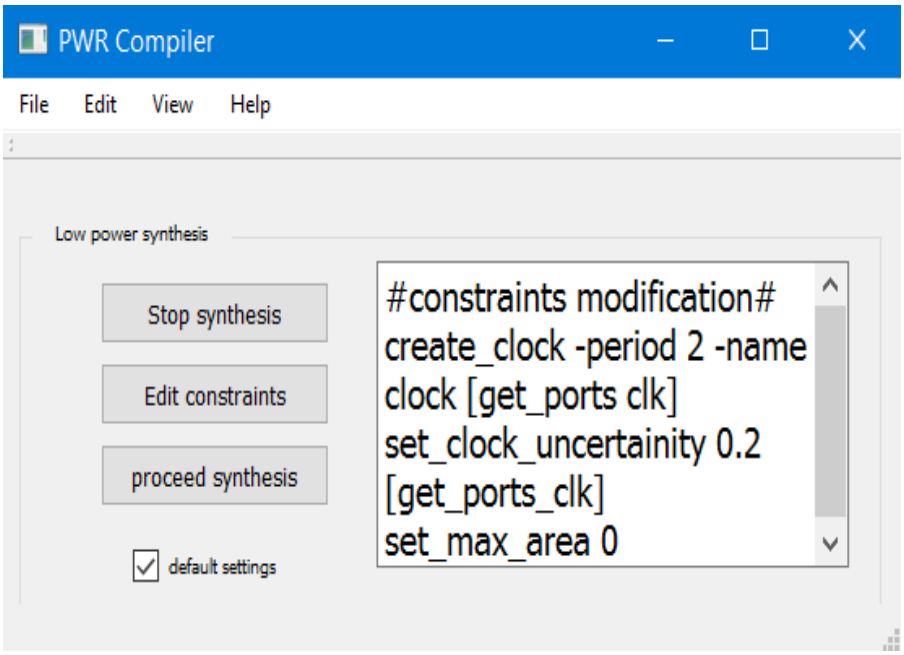
Նկ. 3.11. Առաջարկվող PWR Compiler ծրագրային միջոցում նշումների և բացառությունների սահմանումը

Այս պարագայում ծրագրային միջոցում նախագծողի կողմից կարող են կիրառվել հստակ բացառման մեխանիզմեր, օրինակ.

<<exclude >>

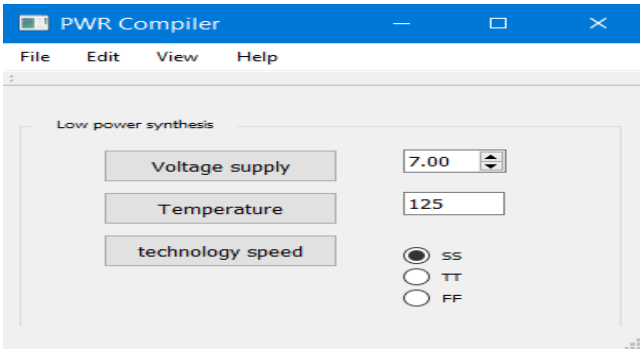
- → method 1,
- → method 2,
- → method 3,
- → method 4,
- → suggested method 1,
- → suggested method 2,
- → suggested method 3.

Լրացնելով տվյալ դաշտերից որևէ մեկը կամ մի քանիսը, ծրագրային միջոցը ավտոմատացված եղանակով իր ալգորիթմից դուրս է հանում ընտրված տարբերակը: Նշված տարբերակը լավագույն լուծումը չէ ծրագրային տեսանկյունից, քանի որ կրճատում է հնարավոր օպտիմալ լուծումների քանակը, բայց և նախագծողի տեսանկյունից դիտարկվում է որպես լրացուցիչ հնարավորություն: Հաջորդ առավելությունը, որը ընձեռվում է, փոփոխություններ կատարելն է սահմանափակումներում, բայց ոչ ֆունկցիոնալ, այլ արժեքների մակարդակով: Դա կարելի է իրականացնել <<constraints modification>> դաշտում (նկ. 3.12):



Նկ. 3.12. Առաջարկվող PWR Compiler ծրագրային միջոցում սահմանափակումների փոփոխության պատուհանը

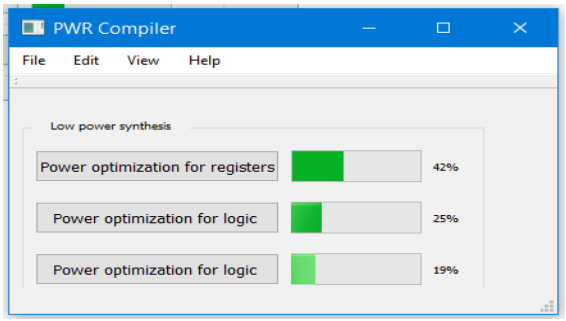
Ցածր էներգասպառմամբ սինթեզի փուլում հնարավոր է նաև ընտրել տեխնոլոգիական գործընթացի արագագործությունը, լարումների և ջերմաստիճանների արժեքները: (նկ. 3.13):



Նկ. 3.13. PWR Compiler ծրագրային միջոցի միջավայրային պարամետրերը

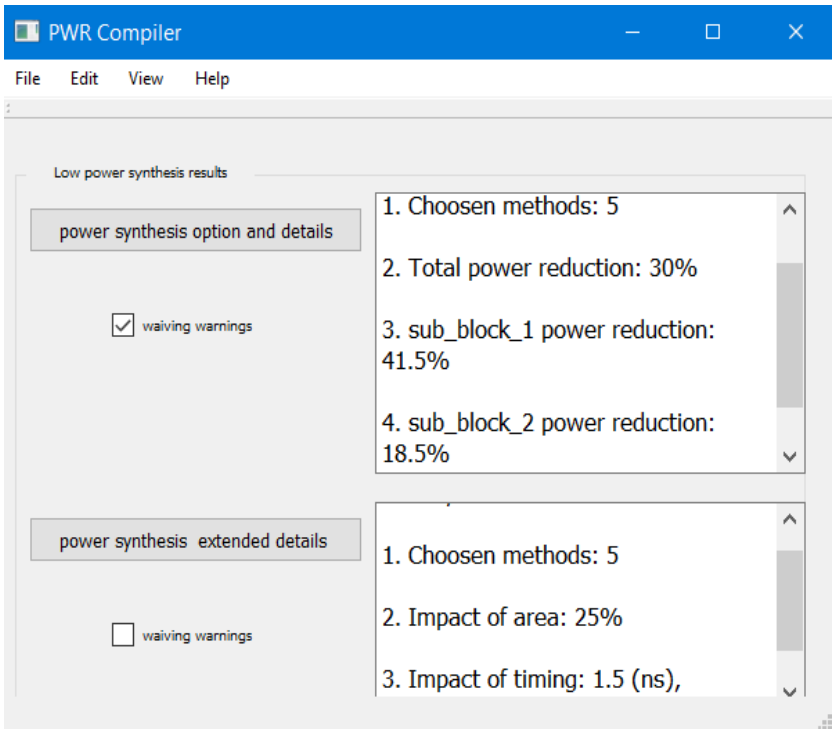
Այսպիսով, այս փուլից հետո PWR Compiler ծրագրային միջոցով ստացվում է առաջարկված և հայտնի մեթոդների ինտեգրումը պրոցեսորային ԻՍ-երում՝ իրագործելու համար անհրաժեշտ ՄՀՁ նկարագրությունը: Ծրագիրը ցածր էներգասպառմամբ սինթեզի գործողությունը կատարում է՝ հաշվի առնելով առանձին ենթահանգույցների արագագործության արժեքները և դրանց հնարավոր փոփոխությունները ժամանակի ընթացքում, որից էլ պարզ է դառնում առաջարկվող դինամիկ լարման և հաճախության մասշտաբավորման մեթոդի կիրառման

հնարավորությունը: Աշխատանքի ընթացքում վերլուծվում են այն ենթահանգույցները, որոնցում մեծ քանակ են կազմում ռեգիստրային տարրերը, որոնք կարող են կիրառելի լինել դասական և առաջարկվող հզորության շրջափակման մեթոդների ներդրման ընթացքում: Միաժամանակ, կատարելով ստատիկ և դինամիկ հզորության վերլուծություններ՝ հնարավորություն է ստեղծվում ներդնելու ցածր էներգասպառման հայտնի և առաջարկվող մոտեցումները, մասնավորապես՝ առաջարկվող չորս մեթոդների համադրված ինտեգրումը (նկ. 3.14):



Նկ. 3.14. PWR Compiler ծրագրային միջոցի ցածր էներգասպառմամբ սինթեզը

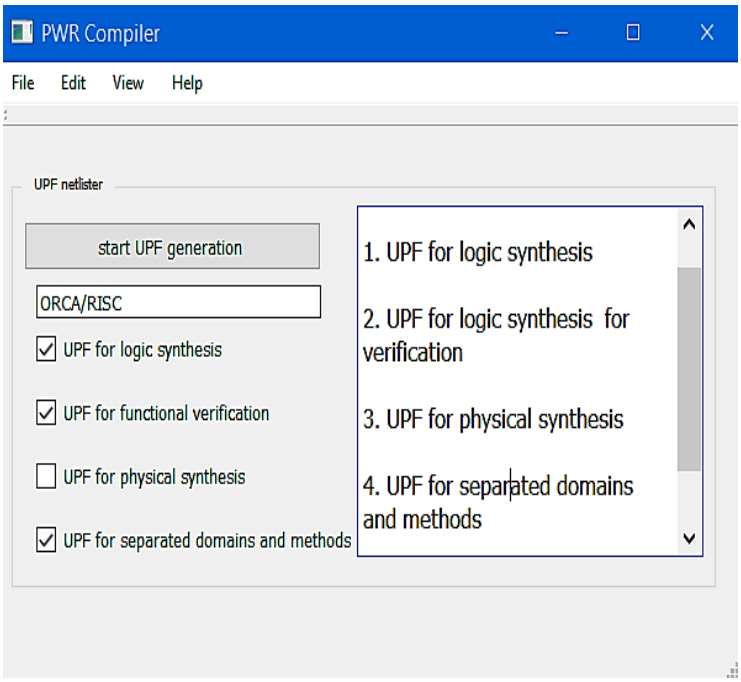
Ստացված արդյունքների և ինտեգրված մեթոդների մասին տեղեկություններ ստանալու համար պետք է սեղմել << power synthesis option and details>> կոճակը, որով բացված նոր պատուհանի աջ կողմում ի հայտ են գալիս ինտեգրված մեթոդները՝ առանձնացված համարակալումով և դրանց ամբողջական անվանումներով: Աջ կողմում ցույց են տրվում ընտրված մեթոդի ինտեգրման արդյունքում ստացված հզորության համեմատական արդյունքները՝ բերված տոկոսներով ըստ ենթահանգույցների, և միասնական հզորության կանխատեսվող նվազարկված արժեքը և դրա տոկոսային համեմատությունը՝ մինչ ծրագրային միջոցի կիրառումը ստացված տվյալների հետ: <<extended details>> կոճակի սեղմումով հնարավոր է տեսնել նաև կանխատեսվող ժամանակային պարամետրերի և մակերեսի հնարավոր փոփոխությունների մասին տեղեկությունները (նկ. 3.15):



Նկ. 3.15. PWR Compiler ծրագրային միջոցի ցածր էներգասպառման տարբերակները

Միասնական հզորության ձևաչափով գեներացումը (UPF netlister) կատարվում է պրոցեսորային ԻՍ-ի արդեն իսկ նախորդ փուլում ընտրված և ինտեգրված ցածր էներգասպառման մեթոդների հիման վրա (նկ. 3.16): ՄՀՁ նկարագրությունը գեներացնելու համար անհրաժեշտ է սեղմել << start UPF generation >> կոճակը և լրացնել անհրաժեշտ լրացուցիչ տարբերակների դաշտերը, մասնավորապես՝ ՄՀՁ-ի հետևյալ պարամետրերը ակտիվացնելու կամ ապաակտիվացնելու միջոցով՝

- <<UPF for logic synthesis >> → (ՄՀՁ՝ նախատեսված տրամաբանական սինթեզի համար),
- <<UPF for functional verification >> → (ՄՀՁ՝ նախաատեսված ֆունկցիոնալ ստուգման համար),
- <<UPF for physical synthesis >> → (ՄՀՁ՝ նախատեսված ֆիզիկական սինթեզի համար),
- <<UPF for separated domains and methods >> → (ՄՀՁ՝ առանձնացված մեթոդների և ենթահանգույցների համար):



Նկ. 3.16. PWR Compiler ծրագրային միջոցի ՄՀՁ-ի գեներացման տարբերակները

Այն դեպքերում, երբ ընտրված չէ ՄՀՁ տարբերակներից ոչ մեկը, ծրագրային միջոցը լռելյայն ընտրում է <<UPF for physical synthesis>> դաշտը և գեներացնում է համապատասխան ՄՀՁ նկարագրությունը ֆիզիկական սինթեզի ծրագրային միջոցի համար:

Առանձնացված դաշտում անհրաժեշտ է նաև սահմանել գեներացվող ՄՀՁ-ի անվանումը: Այն պարագայում, երբ չի ընտրվում անհրաժեշտ տեղանքը, որտեղ պետք է տեղակայվեն ստացված ՄՀՁ-ները, ապա ծրագրային միջոցը լռելյայն տեղակայում է դրանց այն տեղանքում, որտեղից գործարկվել էր ծրագրային միջոցը: Ի տարբերություն այս դաշտի, որը պարտադիր չէ, ՄՀՁ-ի անվան սահմանումը պարտադիր պահանջ է:

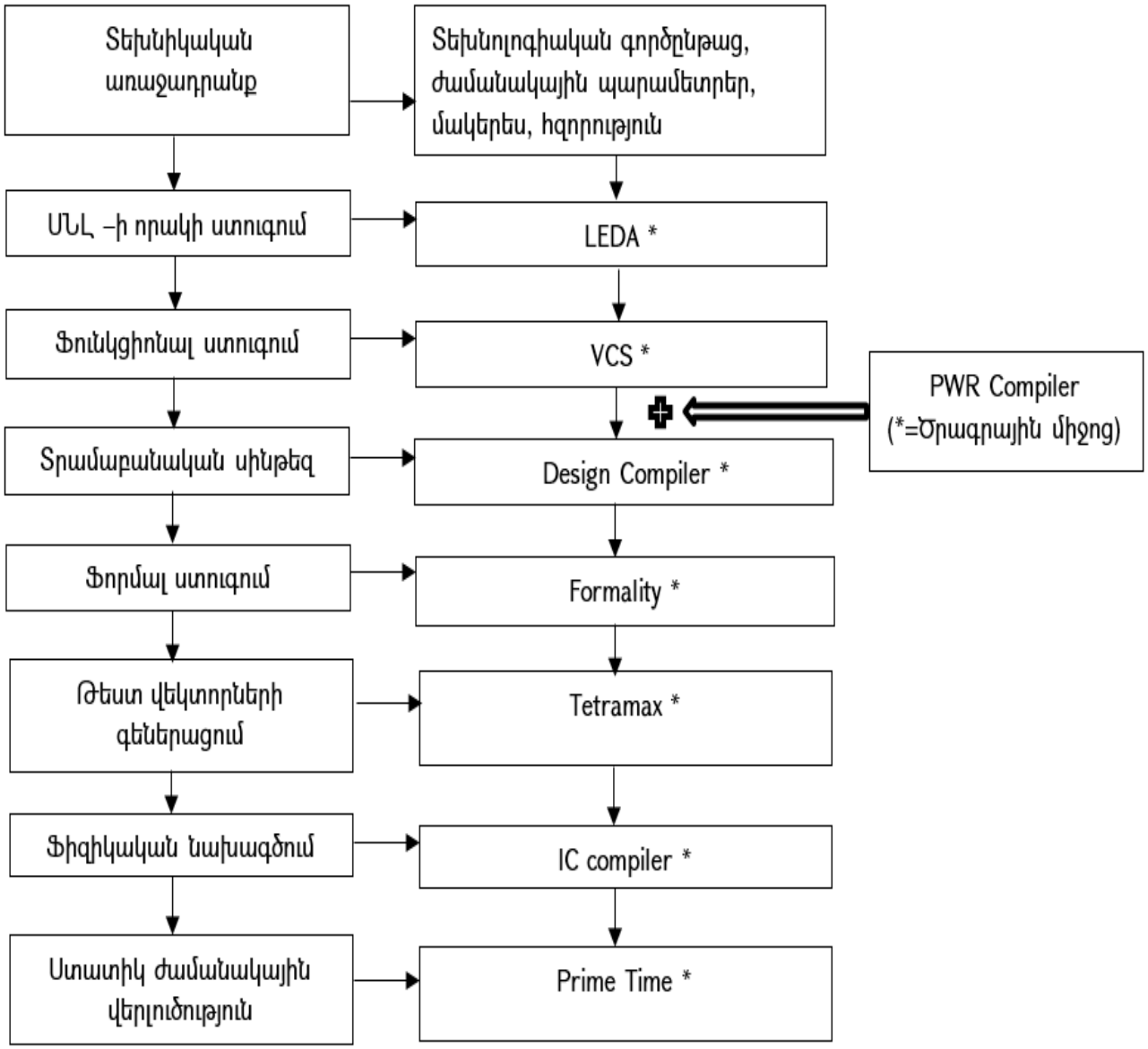
3.2. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման ծրագրային միջոցով նախագծումը

Գնահատելու համար մշակված ծրագրային միջոցի արդյունավետությունը՝ կատարվել է PWR Compiler-ի փորձարարական արդյունքների ստուգում RISC ճարտապետության ORCA պրոցեսորային ԻՍ-ի ցածր էներգասպառմամբ նախագծման միջոցով: Պրոցեսորային ԻՍ-ի նախագծումը ինչպես PWR Compiler ծրագրային

միջոցով, այնպես էլ առանց դրա ներգրավմամբ՝ կատարվել է թվային ավտոմատ նախագծման գործընթացի միջոցով, ստեղծելով նույն մուտքային պայմանները և միջավայրը: Թվային ավտոմատ նախագծման գործընթացի կիրառման արդյունքում պրոցեսորային ԻՍ-ի նախագծման ժամանակահատվածը կրճատվել է, ինչը հնարավորություն է տալիս բավարարել ժամանակակից շուկայի պահանջները և մարտահրավերները: Այս գործընթացը բաղկացած է երկու տիպի ծրագրային միջոցներից՝ սինթեզի և ստուգման: Առաջարկվող գործընթացը տարբերվում է դասական տարբերակից միայն PWR Compiler ծրագրային միջոցի ներդրմամբ (նկ. 3.17):

Սխեմաների նկարագրման լեզուների միջոցով կազմվում է պրոցեսորի վարքային մակարդակի նկարագրությունը, հաջորդ քայլում LEDA [104] ծրագրային միջոցով ստուգվում է վարքային նկարագրի սինթեզի ենթարկվելու հնարավորությունը, քանի որ ծրագրային տեսանկյունից հնարավոր է այնպիսի փոփոխականների ընդգրկում վարքային նկարագրում, որոնք սինթեզի ենթակա չլինեն: Այդպիսի փոփոխականների օրինակներ են forever–ը, կրկնակի ցիկլերը և այլն, որոնք հնարավոր չէ իրագործել սխեմատեխնիկական մակարդակում, բայց ճիշտ են ծրագրային տեսանկյունից: LEDA ծրագրային միջոցը ստուգում է նաև մուտքային նկարագրի որակը սինթեզի տեսանկյունից: Հաջորդ փուլում վարքային նկարագիրը ենթարկվում է ֆունկցիոնալ ստուգման VCS [105] ծրագրային միջոցով, որի մուտքին տրվում են երկու բաղադրիչներ՝ վարքային նկարագիրը և ալգորիթմական մակարդակի նկարագիրը. վերջինս պարունակում է անհրաժեշտ ազդանշանները՝ ֆունկցիոնալ ստուգումն իրականացնելու համար:

Դասական գործընթացում հաջորդ քայլով որոշվում են ցածր էներգասպառման նախագծման մեթոդները և դրանց համար մշակված ՄՀՁ նկարագրությունները:



Նկ. 3.17. PWR Compiler-ով թվային ավտոմատացված նախագծման գործընթացը

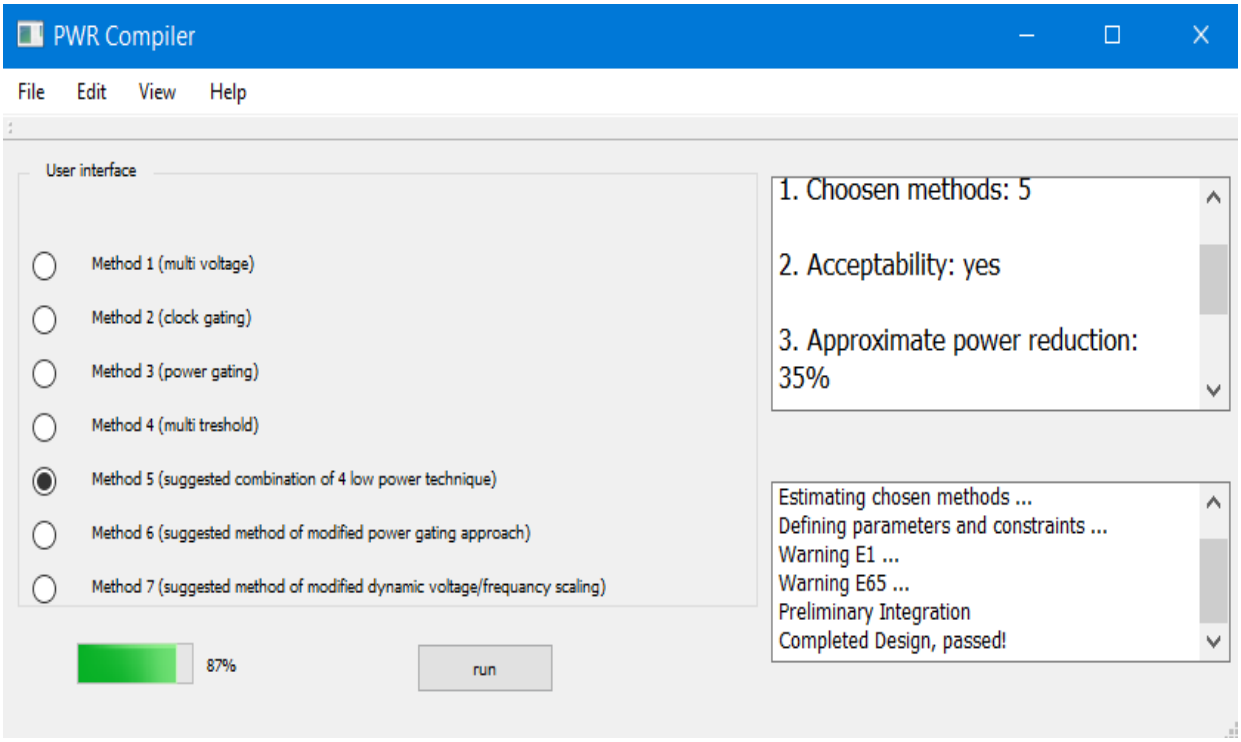
Առաջարկվող տարբերակում այդ գործողությունները կատարվում են PWR Compiler ծրագրային միջոցով: Հաջորդ փուլում արդեն իսկ ընտրված մեթոդով կատարվում է տրամաբանական սինթեզ՝ Design Compiler ծրագրային միջոցով: Տրամաբանական սինթեզի ծրագրային միջոցի մուտքային բաղադրիչներն են՝ վարքային նկարագիրը, ստանդարտ բջիջների գրադարանը, նախագծի սահմանափակումները, ՄՀՁ նկարագրությունը: Այս փուլում տեղի է ունենում աբստրակցիոն մակարդակի իջեցում տեխնոլոգիապես անկախ վարքայինից դեպի տեխնոլոգիայից կախված կառուցվածքային մակարդակը՝ օգտագործելով ստանդարտ բջիջները: Ելքային բաղադրիչներ են՝ կառուցվածքային մակարդակի

նկարագրությունը, ստացված արդյունքների մասին տեղեկությունները և ddc ձևաչափով նկարագիրը, որը պարունակում է կառուցվածքային միացումներ և սահմանափակումներ: Հաջորդիվ կատարվում է ֆորմալ ստուգում Formality ծրագրային միջոցով [106]՝ համոզվելու համար վարքային և կառուցվածքային նկարագրերի համարժեքության մեջ: Հաջորդ քայլով կատարվում է թեստ-վեկտորների գեներացում Tetramax ծրագրային միջոցով [107], որոնք անհրաժեշտ են վերջնական արտադրված ԻՍ-ի ստուգման համար: Թեստ-վեկտորները բացահայտում են երկու տիպի սխալներ՝ ֆիզիկական և ֆունկցիոնալ: Ֆիզիկական սինթեզը տվյալ գործընթացի կարևորագույն հանգույցներից մեկն է, որը պատասխանատու է ֆիզիկական նախագծի պատրաստման և վերջնական արտադրական ֆայլերի ստացման համար, տվյալ դեպքում օգտագործվել են IC Compiler [108] և IC Compiler II [109] ծրագրային միջոցները: Գործընթացի վերջում կատարվում է ստատիկ ժամանակային վերլուծություն Prime Time [110] ծրագրային միջոցով՝ հասկանալու համար, թե վերջնական ֆիզիկական նախագծից հետո ինչպիսի փոփոխությունների են ենթարկվել ԻՍ-ի հապաղումները:

Նկարագրված դասական և առաջարկված գործընթացի կատարմամբ նախագծվել է ORCA պրոցեսորային ԻՍ-ն: ՍՆԼ-ով կատարված նկարագրությունները բազմահիերարխիկ կառուցվածքով են, ORCA_TOP-ը սահմանվել է որպես ամենաբարձր աբստրակցիայի մակարդակ ունեցող <<module>>: Մուտքային սահմանափակումները ընտրվել են միանման և մուտքագրվել ընդհանրացված կարգով: PWR Compiler-ով տարբերակում ORCA պրոցեսորային ԻՍ-ի համար կատարվել են <<Setup input files>> գործողությունները, որոնք գտնվում են առաջնային պատուհանում: Հաջորդիվ կատարվել է նախագծողի կողմից իրականացված հատուկ նշումների ընտրություն՝ <<User-interface>>, մասնավորապես, ORCA պրոցեսորային ԻՍ-ի առանձնահատկությունները հաշվի առնելով՝ ընտրվել է չորս ցածր էներգասպառման մեթոդների համադրման մոտեցումը՝ ծրագրային միջոցով իրականացնելու համար (նկ. 3.18): Տվյալ որոշումն ընդունելու համար հիմք են ծառայել ORCA-ի արագագործությունը և ծանրաբեռնվածությունը:

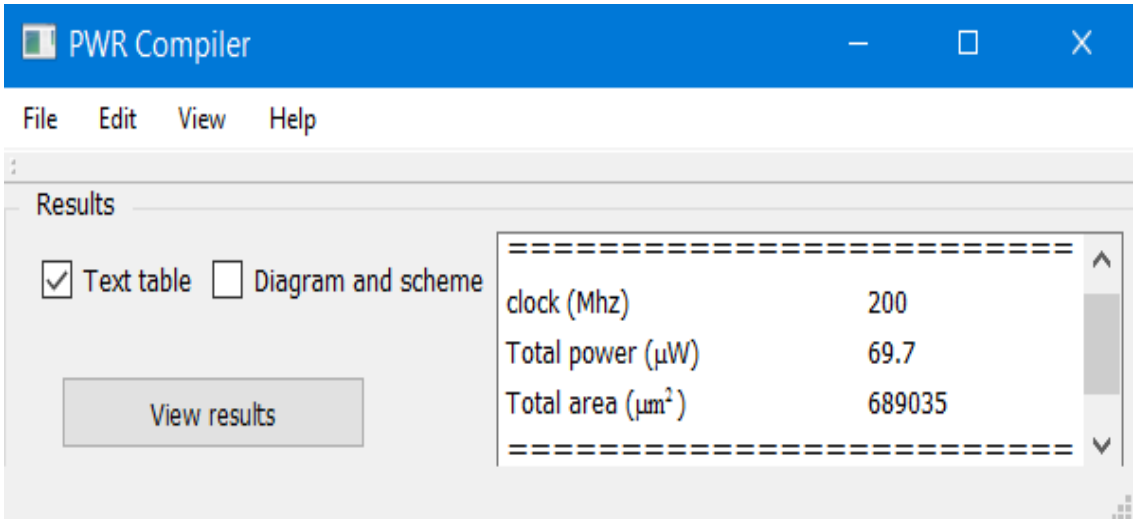
Հաջորդ քայլով կատարվել են ORCA-ի մուտքային տվյալների վերլուծություն և վիրտուալ էներգասպառման գնահատում: Առաջին մոտեցմամբ ծրագրային միջոցը

ընտրված մեթոդի և պրոցեսորային ԻՍ-ի միկրոճարտարապետության միջև ինտեգրումը սահուն կազմակեպելու համար ոչ մի հակասություն չի հայտնաբերել:



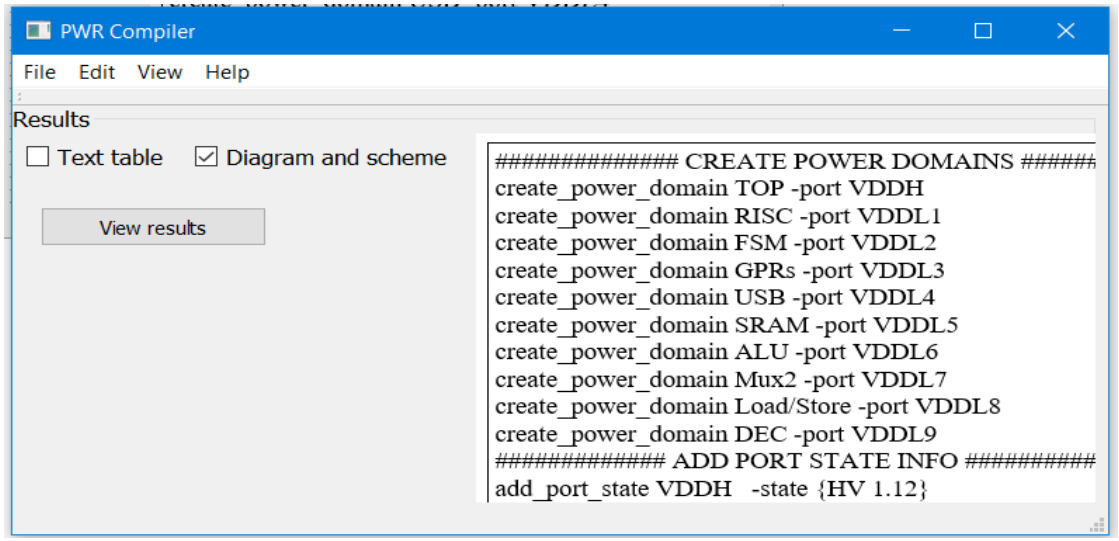
Նկ. 3.18. PWR Compiler-ով ORCA պրոցեսորային ԻՍ-ի նախագծման գործընթացը

PWR Compiler ծրագրային միջոցով իրականացված ցածր էներգասպառմամբ սինթեզի արդյունքները կարելի է դիտարկել <<view results>> կոճակի միջոցով: Այն դեպքում, երբ ակտիվացվում է <<Text table>> դաշտը, ծրագրային միջոցը պատուհանում պատկերում է տեքստային աղյուսակային արդյունքները (նկ. 3.19):



Նկ. 3.19. PWR Compiler-ով ORCA պրոցեսորային ԻՍ-ի ստացված աղյուսակային արդյունքները

<<view results>> կոճակը սեղմելիս, ակտիվանում է <<Diagram and scheme>> դաշտը, PWR Compiler-ի արդյունքների պատուհանում պատկերվում են արդյունքները՝ ձևավորված ՄՀՁ նկարագրությամբ (նկ. 3.20):



Նկ. 3.20. PWR Compiler-ով ORCA պրոցեսորային ԻՍ-ի ստացված ՄՀՁ նկարագրությունը

3.3. Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման ծրագրային միջոցի արդյունավետության գնահատումը

Առաջարկված PWR Compiler ծրագրային միջոցի փորձարարական աշխատանքի արդյունքում ստացված տվյալները ցույց են տալիս բարձր արդյունավետություն և արտադրողականություն: Տվյալ ծրագրով կատարվել է RISC ճարտարապետության ORCA պրոցեսորային ԻՍ-ի նախագծումը: Մուտքային հաճախությունն ընտրվել է 200 *ՄՀց*, տեխնոլոգիական գործընթացը՝ 28 *նմ*: PWR Compiler-ում ինտեգրված են հզորության նվազարկման առաջարկված մեթոդները և մոտեցումները.

- Չորս հիմնական եղանակների համադրման և դրանց միաժամանակյա ներդրման մոտեցումը: Տվյալ մեթոդի իրականացումն ապահովելու համար մշակվել է հատուկ կառավարման համակարգ:

- Հզորության շրջափակման մեթոդում առաջարկված մոտեցումները:
- Դրանց իրագործումն ապահովելու համար մշակվել և ներդրվել են հատուկ նշանակության ԼՄՓ-ներ:
- Դինամիկ լարման և հաճախության մասշտաբավորման առաջարկվող նոր մոտեցումներ՝ հիմնված նոր պարամետրերի ներգրավման և դրանց արժեքների հաշվարկման վրա: Մշակվել է ջերմաստիճանային փոփոխությունները հաշվարկող ենթահանգույց:
- Կատարվել է էներգասպառման, արագագործության և պրոցեսորային ԻՍ-ի նախագծման ժամանակի ընդհանրացված վերլուծություն:

Բերված բոլոր կետերի կատարումն ապահովում է առաջարկված մեթոդների և մոտեցումների գնահատումը: Կարգաբերումները և նախագծերը կատարվել են երկու տարբերակների համար՝ դասական կարգով նախագծված, առանց PWR Compiler ծրագրային միջոցի և կիրառելով PWR Compiler-ը՝ որպես թվային ավտոմատ նախագծման գործընթացի բաղկացուցիչ մաս: Երկու տարբերակների արժեքների միջև համեմատությունը ներկայացված է աղ. 3.1-ում

Աղյուսակ 3.1

PWR Compiler ծրագրային միջոցով և դասական (ձեռքով) տարբերակով նախագծված ORCA պրոցեսորի ստացված համեմատական արդյունքները

Պարամետրերը	Առաջարկված ծրագրային միջոցով նախագծում	Դասական (ձեռքով) տարբերակով նախագծում
Տակտային հաճախությունը (*ՄՀց*)	200	200
Մուտքային ազդանշանների անհրաժեշտ ժամանակային պաշարը (*նվ*)	23,78	20,21
Մուտքային ազդանշանների իրական ժամանակային պաշարը (*նվ*)	-22,57	-20,20
Անհրաժեշտ և իրական ժամանակային պաշարների տարբերությունը (*նվ*)	1,21	0,01
Ընդհանուր հզորությունը (*մվտ*)	34,7116 (-54%)	75,46
Ընդհանուր մակերեսը (*մկմ*^{2})	16961,75 (+3,8%)	16340,79
Նախագծման ժամանակը (*ժամ*)	6	42

Առաջարկվող տարբերակի պարագայում արդեն իսկ <<Virtual power estimation>> փուլից նկատվում են, մոտեցման ճշգրտությունը և արդյունավետությունը: Այսպիսով՝ նախագծի կատարման ժամանակը 7 անգամ կրճատվել է, ինչը կարևորագույն հիմնահարց է արդի պրոցեսորային ԻՍ-երում: Ստացվել է ընդհանուր էներգասպառման նվազեցում մոտ 54%-ով՝ առկա և առաջարկվող տարբեր մոտեցումների համատեղման հաշվին: ԿԲ-ում զբաղեցրած մակերեսի մեծացումը ընդամենը 3,8% է:

Ամփոփելով կարելի է հավաստել, որ PWR Compiler ծրագրային միջոցը բավականին արդյունավետ է՝ RISC ճարտարապետության պրոցեսորային ԻՍ-երի ցածր էներգասպառմամբ նախագծում իրականացնելու համար:

Եզրակացություններ

1. Ատենախոսության մեջ առաջարկված պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկմանն ուղղված մեթոդները, եղանակները և սխեմատեխնիկական հանգույցները ներառվել են մշակված PWR Compiler ծրագրային միջոցում, որը, բացի օգտագործողի համար հարմարավետ ինտերֆեյսով օժտված լինելուց, ապահովում է նաև նախագծի կատարման ժամկետի մինչև յոթ անգամ կրճատում:
2. Մշակված PWR Compiler ծրագրային միջոցի արդյունավետության գնահատման նպատակով այն փորձարկվել է բազմաթիվ սխեմաներում, մասնավորապես՝ RISC ճարտարապետության պրոցեսորային ինտեգրալ սխեմայի վրա: Ժամանակային հապաղումների գործնականում անփոփոխ մնալու պարագայում ընդհանուր էներգասպառումը կրճատվել է մոտ 54%-ով՝ կիսահաղորդչային բյուրեղում ինտեգրալ սխեմայի զբաղեցրած մակերեսի ընդամենը 3,8% աճի դեպքում:

ԵԶՐԱՀԱՆԳՈՒՄ

1. Առաջարկվել են պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկման սկզբունքներ, որոնց հիման վրա նախագծումն ապահովում է սնման աղբյուրից ծախսվող հոսանքների բավարար չափով փոքրացում:
2. Առաջարկվել է RISC ճարտարապետությամբ պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկման եղանակ, որում ցածր հզորության նախագծման մեթոդների համադրության շնորհիվ՝ առկա լուծումների համեմատ ապահովվում է էներգասպառման ժամանակակից պահանջները բավարարող չափով փոքրացում:
3. Մշակվել է հզորության շրջափակման մեթոդի տարատեսակ, որի դեպքում ցածր էներգասպառման ռեժիմում վիճակի պահպանման և վերականգնման ֆունկցիայի՝ պրոցեսորային ինտեգրալ սխեմայի ներսից դեպի եզրագծեր տեղափոխման հաշվին ստացվում է էներգասպառման ստատիկ բաղադրիչի զգալի նվազեցում՝ կիսահաղորդչային բյուրեղի վրա ինտեգրալ սխեմայի զբաղեցրած մակերեսի թույլատրելի մեծացման պարագայում:
4. Հզորության շրջափակման մեթոդում առաջարկված մոտեցման կիրառությունն ապահովելու նպատակով կառուցվել են ելքային ազդանշանի 0,5…1,5 *ԳՀg* հաճախականային միջակայքում աշխատող, վիճակի պահպանման և վերականգնման հնարավորությամբ հատուկ լարման մակարդակի փոխակերպիչներ, որոնք լրացուցիչ լիցքաթափում իրականացնող տրանզիստորային հանգույցի հաշվին ապահովում են զգալիորեն ցածր էներգասպառում, բարձր արագագործություն և պահպանում են համակարգի կայունությունը՝ բացառելով մետակայուն վիճակում հայտնվելու հնարավորությունը:
5. Առաջարկվել է պրոցեսորային ինտեգրալ սխեմաների դինամիկ լարման և հաճախության մասշտաբավորման եղանակ, որի դեպքում ջերմաստիճանի

կախվածությունը հաշվի առնելով՝ ստացվել է էներգասպառման էական նվազարկում, միաժամանակ մեծացնելով կիսահաղորդչային բյուրեղում ինտեգրալ սխեմայի զբաղեցրած մակերեսը:

6. Ատենախոսությունում առաջարկված պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազարկմանն ուղղված մեթոդների, եղանակների և սխեմատեխնիկական հանգույցների հիման վրա մշակված PWR Compiler ծրագրային միջոցի արդյունավետության գնահատման նպատակով այն փորձարկվել է բազմաթիվ սխեմաներում, մասնավորապես՝ RISC ճարտարապետության պրոցեսորային ինտեգրալ սխեմայի վրա: Ժամանակային հապաղումների գործնականում անփոփոխ մնալու պարագայում ընդհանուր էներգասպառումը կրճատվել է մոտ 54%-ով՝ կիսահաղորդչային բյուրեղում ինտեգրալ սխեմայի զբաղեցրած մակերեսի ընդամենը 3,8% աճի դեպքում, միաժամանակ ապահովելով նախագծի կատարման ժամկետի մինչև յոթ անգամ կրճատում:

ՕԳՏԱԳՈՐԾՎԱԾ ԳՐԱԿԱՆՈՒԹՅՈՒՆ

1. Rabaey J. Low Power Design Essentials. –New York: Springer Science, 2009. -365p.
2. Chih-Tang, S. Evolution of the MOS transistor-from conception to VLSI //Proceedings of the IEEE. -1988. -76(10). -P.1280-1326.
3. Understanding Moore's law: four decades of innovation / D.C. Brock and G.E. Moore, eds. - Chemical Heritage Foundation, 2006. – 123p.
4. High energy density lithium batteries: materials, engineering, applications / K.E. Aifantis, S.A. Hackney, and R.V. Kumar, eds. - John Wiley & Sons, 2010. – 254p.
5. Melikyan V.Sh., Babayan E.H. Low power design of digital ICs (based on SAED 90nm Educational Design Kit): Instructional guidelines advanced laboratory works. – Yerevan: SEUA “Chartaraget”, 2013. -80p.
6. Liu, D. and Svensson, C. Power consumption estimation in CMOS VLSI chips // IEEE Journal of Solid-State Circuits. -1994. -29(6). –P.663-670.
7. Borkar, Shekhar. Design challenges of technology scaling // IEEE Micro. – 1999. - 19.4. –P. 23-29.
8. Rabaey J., Chandrakasan A., and Nikolic B. Digital Integrated Circuits. - A Design Perspective. - 2nd ed.- Prentice Hall, 2003. - Chp. 6. -P. 235–308.
9. Leakage in nanometer CMOS technologies / Siva G. Narendra, and Anantha P. Chandrakasan, eds. - Springer Science & Business Media, 2006. – 307 p.
10. Baker J. CMOS Circuit Design, Layout and Simulation. -2010. - 1152 p.

11. Su, Haihua, Frank Liu, Anirudh Devgan, Emrah Acar, and Sani Nassif. Full chip leakage estimation considering power supply and temperature variations // Proceedings of the 2003 international symposium on Low power electronics and design. ACM. - 2003. –P. 78-83.
12. Liu, Z. and Kursun, V. Leakage power characteristics of dynamic circuits in nanometer CMOS technologies // IEEE Transactions on Circuits and Systems II: Express Briefs. –2006. -53(8). -P. 692-696.
13. Leakage power analysis and reduction for nanoscale circuits // A. Agarwal, S. Mukhopadhyay, A. Raychowdhury, Roy, K., IEeE Micro. - 2006.- 26(2). -P. 68-80.
14. Vashishtha V., Gujja A. and Clark L.T. Delay and power tradeoffs for static and dynamic register files // Circuits and Systems (ISCAS): IEEE International Symposium. -2015, May. -P. 2900-2903.
15. Kim, Se Hun, and Mooney Vincent J. Sleepy keeper: a new approach to low-leakage power VLSI design // In Very Large Scale Integration: 2006 IEEE IFIP International Conference. - 2006. -P. 367-372.
16. Klavakolanu, S.S., Raju, M.K., Noorbasha, F. and Kanth, B.R. A review report on low power VLSI systems analysis and modeling techniques // Signal Processing And Communication Engineering Systems (SPACES): IEEE International Conference. – 2015. –P. 142-146.
17. Gu, R.X. and Elmasry, M.I. Power dissipation analysis and optimization of deep submicron CMOS digital circuits // IEEE Journal of solid-state circuits. -1996. -31(5). – P. 707-713.
18. Weste, N.H. and Eshraghian, K. Principles of CMOS VLSI design (Vol. 188) - New York: Addison-Wesley, 1985. -703p.

19. Abbas, Z. and Olivieri, M. Impact of technology scaling on leakage power in nano-scale bulk CMOS digital standard cells // Microelectronics Journal. - 2014. - 45(2). – P. 179-195.

20. Performance prediction for multiple-threshold 7nm-FinFET-based circuits operating in multiple voltage regimes using a cross-layer simulation framework / S. Chen, Y. Wang, X. Lin, Q. Xie, // SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S).- 2014. –P. 1-2.

21. Chu Z., Y. Xia and L. Wang. Multi-supply voltage (MSV) driven SoC floorplanning for fast design convergence. Integration, the VLSI Journal. -2016. -52. -P. 335-346.

22. Multiple On-Chip Power Supply Systems / P. Inna, R. Jakushokas, M. Popovich, A.V. Mezhiba, // On-Chip Power Delivery and Management. - Springer International Publishing, 2016. -P. 603-617.

23. Kao J., Chandrakasan A. and Antoniadis D. Transistor sizing issues and tool for multi-threshold CMOS technology // Proceedings of the 34th annual Design Automation Conference. ACM. -1997, June. -P. 409-414.

24. Lim, K.H. and Kim, T. An optimal algorithm for allocation, placement, and delay assignment of adjustable delay buffers for clock skew minimization in multi-voltage mode designs // Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific. – 2011, January. –P. 503-508.

25. Le Sueur, Etienne, and Gernot Heiser. Dynamic voltage and frequency scaling: The laws of diminishing returns // Proceedings of the international conference on Power aware computing and systems. – 2010. -P. 1-8.

26. U.S. Patent 6,314,522. Multi-voltage level CPU module / W.W. Chu, D.H. Chen.- Acqis Technology, Inc., 2001.

27. Shimazaki, Y., Zlatanovici, R. and Nikolic, B. A shared-well dual-supply-voltage 64-bit ALU // IEEE Journal of Solid-State Circuits. – 2004. -39(3). -P. 494-500.

28. Mutoh Shin'ichiro, Satoshi Shigematsu, Yasuyuki Matsuya, Hideki Fukuda, and Junzo Yamada. A 1 V multi-threshold voltage CMOS DSP with an efficient power management technique for mobile phone application // Solid-State Circuits Conference, Digest of Technical Papers. 42nd ISSCC, IEEE International. – 1996. –P. 168 -169.

29. Ahmad, G., Kumar, Y. and Sahu, P.K. High performance multi threshold voltage level converter for multi-VDD systems. // IEEE Engineering and Systems (SCES), 2013 Students Conference. -2013, April. –P. 1-4.

30. Zhai B., Blaauw D., Sylvester D., Flautner K. Theoretical and practical limits of dynamic voltage scaling // Design Automation Conference, 7-11 Jul., 2004.- San Diego, 2004.- P. 868-873.

31. U.S. Patent 8,370,654. AVS-adaptive voltage scaling / M. Hasko, E. Reches, R. Ecker and I. Bourstein, Marvell Israel (MISL) Ltd. - 2013.

32. Ansari, M.S. and Tripathi, S.K. Low Power Design Techniques: Classical and beyond CMOS Era // Design and Modeling of Low Power VLSI Systems, IGI Global. -2016. – P. 1-26.

33. Panda P., Silpa B., Shrivastava A., Gummidipudi K. Power-efficient System Design.- Springer, 2010.- 253 p.

34. Kursun V, Friedman E.G. Multi-voltage CMOS circuit design. -John Wiley & Sons, 2006, Aug. 30. –221p.

35. Liu R.H. Multi-voltage Adaptive Voltage Scaling SoC Reference Design // National Semiconductor Corporation. -2006, Nov. 15. –P. 1-8.

36. Sitik, C. and Taskin, B. Multi-corner multi-voltage domain clock mesh design // Proceedings of the 23rd ACM international conference on Great lakes symposium on VLSI. -2013, May. –P. 209-214.

37. Calhoun B., Chandrakasan A. Ultra dynamic Voltage scaling (UDVS) using sub threshold operation and local Voltage dithering // IEEE Journal of Solid-State Circuits.- 2006.- Vol. 41.- P. 238-245.

38. Anis, M. and Elmasry, M. Multi-threshold CMOS digital circuits–managing leakage power. – 2003. - Vol. 3.- 46p.

39. Multi-threshold asynchronous circuit design for ultra-low power / A. Bailey, A. Al Zahrani, G. Fu, J. Di, // Journal of Low Power Electronics. - 2008. - 4(3). - P. 337-348.

40. Kao, J.T. and Chandrakasan, A.P. Dual-threshold voltage techniques for low-power digital circuits // IEEE Journal of Solid-state circuits. -2000. -35(7). -P. 1009-1018.

41. Pedram M., Rabaey J.M. Power aware design methodologies. -Springer Science & Business Media, 2002, Jun 30. –519p.

42. Kim Chris, and Kaushik Roy. Dynamic VTH scaling scheme for active leakage power reduction // Proceedings of the conference on Design, automation and test in Europe, IEEE Computer Society.- 2002. –P. 163.

43. Pandey, B., Yadav, J., Pattanaik, M., and Rajoria, N. Clock gating based energy efficient ALU design and implementation on FPGA // IEEE Energy Efficient Technologies for Sustainability (ICEETS): 2013 International Conference. -2013, April. -P. 93-97.

44. A gated clock scheme for low power scan testing of logic ICs or embedded cores / Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, // IEEE Test Symposium, 2001: Proceedings. 10th Asian. -2001. –P. 253-258.

45. Mahmoodi, H., Tirumalashetty, V., Cooke, M., and Roy, K. Ultra low-power clocking scheme using energy recovery and clock gating // IEEE transactions on very large scale integration (VLSI) systems. -2009. -17(1). -P. 33-44.

46. DCG: deterministic clock-gating for low-power microprocessor design / H. Li, S. Bhunia, Y. Chen, K. Roy, // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. -2004. -12(3). -P. 245-254.

47. The impact of clock gating schemes on the power dissipation of synthesizable register files / M. Mueller, A. Wortmann, S. Simon, M. Kugel, // IEEE Circuits and Systems, ISCAS'04: Proceedings of the International Symposium.- 2004, May 23. - Vol. 2. - P. II-609.

48. Wu Q., Pedram M., Wu X. Clock-gating and its application to low power design of sequential circuits // IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications. -2000, Mar. -47(3) –P. 415-420.

49. Shashidar R., Kumar R.S., MahalingaSwamy A.M., and Roopa M FPGA Implementation of Low Power Pipelined 32-Bit RISC Processor Using Clock Gating // Proceedings of the International Conference on Data Engineering and Communication Technology. -Springer Singapor. -2017. -P. 769-777.

50. Donno, M., Ivaldi, A., Benini, L. and Macii, E. Clock-tree power optimization based on RTL clock-gating // Proceedings of the 40th annual Design Automation Conference, ACM. -2003, June. –P. 622-627.

51. A 90-nm variable frequency clock system for a power-managed Itanium architecture processor / T. Fischer, J. Desai, B. Doyle, S. Naffziger, // IEEE Journal of Solid-State Circuits. -2006. -41(1). -P. 218-228.

52. Jiang, H., Marek-Sadowska, M. and Nassif, S.R. Benefits and costs of power-gating technique // Computer Design: VLSI in Computers and Processors, ICCD 2005: Proceedings the IEEE International Conference. -2005, October. –P. 559-566.

53. Usami, K. and Ohkubo, N. A design approach for fine-grained run-time power gating using locally extracted sleep signals // IEEE Computer Design, ICCD 2006: International Conference. -2007, October. -P. 155-161.

54. Calhoun, B.H., Honore, F.A. and Chandrakasan, A.P. A leakage reduction methodology for distributed MTCMOS // IEEE Journal of Solid-State Circuits. -2004. -39(5). -P. 818-826.

55. Anis, M., Mahmoud, M., Elmasry, M., and Areibi, S. Dynamic and leakage power reduction in MTCMOS circuits using an automated efficient gate clustering technique // Proceedings of the 39th annual Design Automation Conference. ACM. -2002, June. –P. 480-485.

56. Shin, Y., Seomun, J., Choi, K.M., and Sakurai, T. Power gating: Circuits, design methodologies, and best practice for standard-cell VLSI designs // ACM Transactions on Design Automation of Electronic Systems (TODAES). -2010. -15(4). -P.28.

57. Chowdhury, M.H., Gjanci, J., and Khaled, P. Innovative power gating for leakage reduction // Circuits and Systems, ISCAS 2008: IEEE International Symposium. -2008, May. -P. 1568-1571.

58. A multi-mode power gating structure for low-voltage deep-submicron CMOS ICs / S. Kim, S.V. Kosonocky, D.R. Knebel, K. Stawiasz, // IEEE Transactions on Circuits and Systems II: Express Briefs. -2007. -54(7). -P. 586-590.

59. Minimizing inductive noise in system-on-a-chip with multiple power gating structures / S. Kim, S.V. Kosonocky, D.R. Knebel, K. Stawiasz, // IEEE Solid-State Circuits

Conference, ESSCIRC'03: Proceedings of the 29th European. - 2003, September. - P. 635-638.

60. Low Power Methodology Manual For System-on-Chip Design / Michael Keating, David Flynn, Robert Aitken, Alan Gibbons, - Springer, 2007. -299p.
61. U.S. Patent 7,479,801. Power gating techniques able to have data retention and variability immunity properties / S. Bhattacharya, International Business Machines Corporation. -2009.
62. U.S. Patent 9,632,561. Power-gating media decoders to reduce power consumption / A. Lindahl and A.J. Guetta, Apple Inc. -2017.
63. Power management of datacenter workloads using per-core power gating / J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, // IEEE Computer Architecture Letters. -2009. -8(2). -P. 48-51.
64. Mistry, J.N., Al-Hashimi, B.M., Flynn, D., and Hill, S. Sub-clock power-gating technique for minimising leakage power during active mode // IEEE Design, Automation & Test in Europe Conference & Exhibition (DATE). -2011, March. -P. 1-6.
65. Dynamic state-retention flip-flop for fine-grained power gating with small design and power overhead / S. Henzler, G. Georgakos, M. Eireiner, T. Nirschl, // IEEE Journal of Solid-State Circuits. - 2006. -41(7). -P. 1654-1661.
66. Shahid A., Arif S., Qadri M.Y. and Munawar S. Power Optimization Using Clock Gating and Power Gating: A Review // Innovative Research and Applications in Next-Generation High Performance Computing, IGI Global. -2016. -P. 1-20.
67. U.S. Patent 7,622,955. Power savings with a level-shifting boundary isolation flip-flop (LSIFF) and a clock controlled data retention scheme / R. Vilangudipitchai, S.K. Gururajarao, H.T. Mair, A. Wang, U.U. Ko and Honnavara- S. Prasad. - Texas Instruments Incorporated. -2009.

68. Roy, S., Ranganathan, N., and Katkoori, S. State-retentive power gating of register files in multicore processors featuring multithreaded in-order cores // IEEE Transactions on Computers. -2011. -60(11). -P. 1547-1560.
69. U.S. Patent 8,242,826. Retention flip-flop / S.A. Chi, S.T. Shen, J.A. Lee and Y.H.Lee. - Taiwan Semiconductor Manufacturing Co., Ltd., -2012.
70. ElShoukry, M., Tehranipoor, M. and Ravikumar, C.P. Partial gating optimization for power reduction during test application // IEEE Test Symposium: Proceedings at 14th Asian. -2005, December. -P. 242-247.
71. U.S. Patent 7,652,513. Texas Instruments Incorporated. Slave latch controlled retention flop with lower leakage and higher performance / B.P. Rao, S.K. Gururajarao, and D.N. Shah. -2010.
72. A magnetic tunnel junction based zero standby leakage current retention flip-flop / K. Ryu, J. Kim, J. Jung, J.P. Kim, // IEEE Transactions on Very Large Scale Integration (VLSI) Systems. -2012. -20(11). -P. 2044-2053.
73. García, J.C., Montiel-Nelson, J.A. and Nooshabadi, S. High performance CMOS dual supply level shifter for a 0.5 V input and 1V output in standard 1.2 V 65nm technology process // IEEE Communications and Information Technology. ISCIT 2009: 9th International Symposium. -2009. -P. 963-966.
74. Lanuzza, M., Corsonello, P., and Perri, S. Low-power level shifter for multi-supply voltage designs // IEEE Transactions on Circuits and Systems II: Express Briefs. -2012. -59(12). -P. 922-926.
75. Khan, Q.A., Wadhwa, S.K., and Misri, K. A single supply level shifter for multi-voltage systems // IEEE VLSI Design, 2006, January: Held jointly with 5th International Conference on Embedded Systems and Design. -2006. -P. 4.

76. An, Ji Yeon, Hyoun Soo Park, and Young Hwan Kim. Level up/down converter with single power supply voltage for multi VDD Systems. // Journal of Semiconductor Technology and Science. -2010. -10.1. –P. 55-60.

77. OPEN RISC.- **https://openrisc.io/**

78. Simple RISC.- **https://www.ece.umd.edu/~blj/RiSC/RiSC-isa.pdf**

79. Clock Gating and Multi-VTH Low Power Design Methods Based on 32/28 nm ORCA Processor / V. Melikyan, D. Babayan, E. Babayan, P. Petrosyan, A. Melikyan, E. Mkrtchyan // IEEE East-West Design & Test Symposium, Sep. 26-29, 2015.- Batumi, Georgia, 2015.- P. 47-50.

80. Low Power OpenRISC Processor with Power Gating, Multi-VTH and Multi-Voltage Techniques / V. Melikyan, T. Hakhverdyan, S. Manukyan, A. Grigoryan, D. Babayan // IEEE East-West Design & Test Symposium, Oct. 14-17, 2016.- Yerevan, Armenia, 2016.- P. 33-36.

81. 1.9 GHz 1.05V 16-bit RISC Core for High Density and Low Power Operation in 28nm Technology / D. Babayan, E. Babayan, P. Petrosyan, A. Tumanyan, E. Kagramanyan, T. Hakhverdyan // IEEE East-West Design & Test Symposium, Oct. 14-17, 2016.- Yerevan, Armenia, 2016.- P. 459-462.

82. Silc, J., Robic, B. and Ungerer, T. Processor architecture: from dataflow to superscalar and beyond. -Springer Science & Business Media, 2012. –379p.

83. Write-Back Technique for Single-Ended 7T SRAM cell / V. Melikyan, A. Avetisyan, D. Babayan, K. Safaryan, T. Hakhverdyan // XXXVII International Scientific Conference on Electronics and Nanotechnology, Apr. 18 -20, 2017.- Kyiv, Ukraine, 2017.- P. 112-115.

84. A –1.8V to 0.9V body bias, 60 GOPS/W 4-core cluster in low-power 28nm UTBB FD-SOI technology / Rossi D., Pullini A., Gautschi M., Loi I., // SOI-3D-Subthreshold Microelectronics Technology Unified Conference (S3S). -2015. - P. 1-3.

85. Բաբայան Դ. Բարձր հաճախականային ցածր էներգասպառմամբ լարման մակարդակի փոխակերպիչ // Proceeding of Engineering Academy of Armenia (PEAA).- 2017.-V.14, N 1.- P. 148-153.

86. HSPICE. Simulation and Analysis User Guide. – Synopsys, 2013.

87. Custom Compiler.- https://www.synopsys.com/implementation-and-signoff/custom-implementation/custom-compiler.html

88. Babayan D. Power optimization approach of ORCA processor for 32/28nm technology node // IEEE Computer Science Information Technology Sep. 28 - Oct 2, 2015.- Yerevan, Armenia, 2015.- P. 231-233.

89. Բաբայան Դ. Համապիտանի հաջորդական դողի թվային միջուկի նախագծում՝ ցածր էներգասպառման մեթոդներով // ՀԱՊՀ Լրաբեր. Գիտական հոդվածների ժողովածու.- Երևան, 2016. –Մաս 1.- էջ 234-240:

90. Melikyan V., Krrikyan H., Hovhannisyan T., Babayan D. Synthesizable Assertions for Clock Domain Crossing Verification for the USB Prototype // Proceedings of the Republic of Armenia National Academy of Sciences and National Polytechnic University of Armenia. Series of Technical Sciences.- 2016.-V.69, N 2.- P. 138-150.

91. 32/28 NM LOW POWER ORCA PROCESSOR WITH MULTI-VOLTAGE SUPPLY / V. Melikyan, D. Babayan, E. Babayan, P. Petrosyan, V. Melkonyan, H. Manukyan, A. Manukyan // 10th International Conference On Semiconductor Micro- & Nanoelectronics, Sep. 11-13, 2015.-Yerevan, Armenia, 2015.- P. 123-127.

92. A new Approach of Multi Voltage and Adaptive Voltage Scaling Techniques for 16 nm FinFET RISC Processor / D. Babayan, E. Babayan, S. Antonyan, A. Salmasyan, E. Kagramanyan, A. Avetisyan // XXXVII International Scientific Conference on Electronics and Nanotechnology, Apr. 18 -20, 2017.- Kyiv, Ukraine, 2017.- P. 128-131.

93. Raha, P. A 0.6-4.2 V low-power configurable PLL architecture for 6 GHz-300 MHz applications in a 90 nm CMOS process // IEEE VLSI Circuits Symposium, 2004. Digest of Technical Papers: -2004. –P. 232-235.
94. Nebhrajan, V.A. Asynchronous FIFO Architectures. 2007-11-14]. http://www. gecoities. com/deepakgeorge2000/. vlsi-book/async-fifo. pdf.
95. Pettichord, B. Design for testability // Pacific Northwest Software Quality Conference. -2002, October. -P. 1-28.
96. Саммерфилд М. Qt. Профессиональное программирование. Разработка кроссплатформенных приложений на C++.- М.: Символ-Плюс, 2011.- 560 с.
97. Прата С. Язык программирования С: Лекции и упражнения.- М.: Вильямс, 2006.- 960с.
98. Van Rossum, G. and Drake, F.L. Python language reference manual. Network Theory. -2003. -144p.
99. Windows OS.- https://www.microsoft.com/en-us/windows/.
100. Design Compiler. -https://www.synopsys.com/support/training/rtl-synthesis/design-compiler.html
101. 1801- IEEE Standard for Design and Verification of Low Power Integrated Circuits. -2009.
102. Thomas D., Moorby P. The Verilog Hardware Description Language. -5th ed. – Springer, 2002. -386p.
103. Automation SD. VHDL Reference Manual. -Redmond, Washington, 1997.
104. LEDA. // Synopsys.com
105. VCS User Guide.- March, 2009.
106. Formality.-https://www.synopsys.com/verification/static-and-formal-verification/formality-and-formality-ultra.html

107. Tetramax.-https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/test-automation/tetramax-atpg.html

108. IC Compiler.-https://www.synopsys.com/implementation-and-signoff/physical-implementation/ic-compiler.html

109. IC Compiler II.-https://www.synopsys.com/implementation-and-signoff/physical-implementation/ic-compiler-ii.html

110. PrimeTime.-https://www.synopsys.com/apps/support/training/primetime1_fcd.html

ՀԱՎԵԼՎԱԾ 1

- **PWR Compiler ծրագրային միջոցի ներդրման ակտ <<Սինոփսիս Արմենիա>> ՓԲԸ-ում:**

ՀԱՎԵԼՎԱԾ 2

SYNOPSYS®

№ 353/1/17 «28» 04 2017

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

Դավիթ Ռոբերտի Բաբայանի «Պրոցեսորային ինտեգրալ սխեմաների էներգասպառման նվազեցման միջոցների մշակումը և հետազոտումը» թեմայով թեկնածուական ատենախոսության արդյունքների

ՆԵՐԴՐՄԱՆ ԱԿՏ

ՀԱՊՀ «Էլեկտրոնիկա, միկրո և նանոէլեկտրոնիկա» մասնագիտությամբ 3-րդ կուրսի ասպիրանտ Դ. Ռ. Բաբայանի կողմից մշակված PWR Compiler ծրագրային միջոցը ներդրված է «ՍԻՆՈՓՍԻՍ ԱՐՄԵՆԻԱ» ՓԲԸ-ում: Այն նախատեսված է պրոցեսորային ինտեգրալ սխեմաների էներգասպառման վերլուծության և նվազեցման համար: Մշակված PWR Compiler ծրագրային միջոցի կիրառումը զգալիորեն պարզեցրել է պրոցեսորային համակարգերի մշակման և ցածր էներգասպառմամբ նախագծման գործընթացը՝ կրճատելով նախագծի կատարման ընդհանուր ժամանակը: Ծրագրային միջոցը հանդիսանում է պրոցեսորային ինտեգրալ սխեմաների ավտոմատացված նախագծման գործընթացի բաղկացուցիչ մասը՝ ապահովելով նրա ցածր էներգասպառման բաղադրիչը: Մշակված ծրագրային միջոցը բավարարում է ժամանակակից էլեկտրոնային նախագծման բնագավառում կիրառվող ծրագրային գործիքային միջոցին ներկայացվող պահանջներին:

Անալոգային սխեմաների նախագծման
դեպարտամենտի տնօրեն՝ Ա. Մուրադյան

"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

- **RISC պրոցեսորային ԻՍ-ի կառուցվածքային նկարագիրը Verilog սխեմաների նկարագրման լեզվով**

```
module RISC4 ( pc0, clk, reset, done, x1 );

  input [7:0] pc0;
  input clk, reset, x1;
  output done;
  wire   x2, x3, x4, x5, x6, x7, y1, y2, y3, y4, y6, y7, y10, y11, y13, y14,
         tkn_ntk, c_out, n1, n2, n3;
  wire   [7:0] address;
  wire   [15:0] gout1;
  wire   [15:0] mem_out;
  wire   [3:0] A1;
  wire   [3:0] A2;
  wire   [3:0] A3;
  wire   [15:0] gout2;
  wire   [15:0] g_in;
  wire   [7:0] pc;
  wire   [15:0] ir;
  wire   [15:0] result;
  wire   [4:0] flags;

  FSM m1 ( .clk(clk), .x1(x1), .x2(x2), .x3(x3), .x4(x4), .x5(x5), .x6(x6),
         .x7(x7), .y1(y1), .y2(y2), .y3(y3), .y4(y4), .y6(y6), .y7(y7), .y10(
         y10), .y11(y11), .y12(done), .y13(y13), .y14(y14), .reset(reset) );
  SRAM m2 ( .clk(clk), .y6(y6), .address(address), .gout1(gout1), .mem_out(
         mem_out) );
  gprs1 m3 ( .y4(y4), .y11(y11), .A1(A1), .A2(A2), .A3(A3), .gout1(gout1),
         .gout2(gout2), .g_in(g_in) );
  program_counter m4 ( .clk(clk), .y1(y1), .y7(y7), .pc(pc), .pc0(pc0),
         .tkn_ntk(tkn_ntk), .ir({ir[15:4], n3, n2, n1, ir[0]}) );
  ireg m5 ( .clk(clk), .mem_out(mem_out), .y2(y2), .ir(ir) );
  decoder m6 ( .ir({ir[15:4], n3, n2, n1, ir[0]}), .x2(x2), .x3(x3), .x4(x4),
         .x5(x5), .x6(x6), .x7(x7) );
  ALU m7 ( .gout1(gout1), .gout2(gout2), .ir({ir[15:4], n3, n2, n1, ir[0]}),
         .y10(y10), .result(result), .c_out(c_out) );
  mux2 m8 ( .y11(y11), .mem_out(mem_out), .result(result), .g_in(g_in) );
```

```
  Load_Store m9 ( .pc(pc), .ir({ir[15:4], n3, n2, n1, ir[0]}), .y3(y3), .y13(
      y13), .y6(y6), .address(address) );
  operands_gpr m10 ( .A1(A1), .A2(A2), .A3(A3), .ir({ir[15:4], n3, n2, n1,
      ir[0]}) );
  branch m11 ( .clk(clk), .ir({ir[15:4], n3, n2, n1, ir[0]}), .y14(y14),
      .flags(flags), .tkn_ntk(tkn_ntk) );
  flags_set m12 ( .result(result), .flags(flags), .c_out(c_out) );
  BUF_U U1 ( .A(ir[3]), .X(n3) );
  BUF_U U2 ( .A(ir[2]), .X(n2) );
  BUF_U U3 ( .A(ir[1]), .X(n1) );
endmodule

module mux2 ( y11, mem_out, result, g_in );

  input [15:0] mem_out;
  input [15:0] result;
  output [15:0] g_in;
  input y11;
  wire   n1;
  INV U1 ( .A(y11), .X(n1) );
  AO22 U2 ( .A1(y11), .A2(result[9]), .B1(mem_out[9]), .B2(n1), .X(
      g_in[9]) );
  AO22 U3 ( .A1(result[0]), .A2(y11), .B1(mem_out[0]), .B2(n1), .X(
      g_in[0]) );
  AO22 U4 ( .A1(result[1]), .A2(y11), .B1(mem_out[1]), .B2(n1), .X(
      g_in[1]) );
  AO22 U5 ( .A1(result[2]), .A2(y11), .B1(mem_out[2]), .B2(n1), .X(
      g_in[2]) );
  AO22 U6 ( .A1(result[3]), .A2(y11), .B1(mem_out[3]), .B2(n1), .X(
      g_in[3]) );
  AO22 U7 ( .A1(result[4]), .A2(y11), .B1(mem_out[4]), .B2(n1), .X(
      g_in[4]) );
  AO22 U8 ( .A1(result[5]), .A2(y11), .B1(mem_out[5]), .B2(n1), .X(
      g_in[5]) );
  AO22 U9 ( .A1(result[6]), .A2(y11), .B1(mem_out[6]), .B2(n1), .X(
      g_in[6]) );
  AO22 U10 ( .A1(result[7]), .A2(y11), .B1(mem_out[7]), .B2(n1), .X(
      g_in[7]) );
  AO22 U11 ( .A1(result[8]), .A2(y11), .B1(mem_out[8]), .B2(n1), .X(
```

```
        g_in[8]) );
  AO22 U12 ( .A1(result[10]), .A2(y11), .B1(mem_out[10]), .B2(n1), .X(
        g_in[10]) );
  AO22 U13 ( .A1(result[11]), .A2(y11), .B1(mem_out[11]), .B2(n1), .X(
        g_in[11]) );
  AO22 U14 ( .A1(result[12]), .A2(y11), .B1(mem_out[12]), .B2(n1), .X(
        g_in[12]) );
  AO22 U15 ( .A1(result[13]), .A2(y11), .B1(mem_out[13]), .B2(n1), .X(
        g_in[13]) );
  AO22 U16 ( .A1(result[14]), .A2(y11), .B1(mem_out[14]), .B2(n1), .X(
        g_in[14]) );
  AO22 U17 ( .A1(result[15]), .A2(y11), .B1(mem_out[15]), .B2(n1), .X(
        g_in[15]) );

endmodule

module Load_Store ( pc, ir, y3, y13, y6, address );

  input [7:0] pc;
  input [15:0] ir;
  output [7:0] address;
  input y3, y13, y6;
  wire   N2, N3, N4, N5, N6, N7, N8, N9, N10, n3, n1, n2;
  LDPQ \address_reg[6]  ( .G(N2), .D(N9), .Q(address[6]) );
  LDPQ \address_reg[5]  ( .G(N2), .D(N8), .Q(address[5]) );
  LDPQ \address_reg[4]  ( .G(N2), .D(N7), .Q(address[4]) );
  LDPQ \address_reg[3]  ( .G(N2), .D(N6), .Q(address[3]) );
  LDPQ \address_reg[2]  ( .G(N2), .D(N5), .Q(address[2]) );
  LDPQ \address_reg[1]  ( .G(N2), .D(N4), .Q(address[1]) );
  LDPQ \address_reg[0]  ( .G(N2), .D(N3), .Q(address[0]) );
  LDPQ \address_reg[7]  ( .G(N2), .D(N10), .Q(address[7]) );
  INV U2 ( .A(y13), .X(n2) );
  ND2_S U3 ( .A1(n2), .A2(n3), .X(N2) );
  INV U4 ( .A(n3), .X(n1) );
  OAI21 U5 ( .A1(y3), .A2(y6), .B(n2), .X(n3) );
  AO22 U6 ( .A1(pc[1]), .A2(y13), .B1(ir[1]), .B2(n1), .X(N4) );
  AO22 U7 ( .A1(pc[0]), .A2(y13), .B1(ir[0]), .B2(n1), .X(N3) );
  AO22 U8 ( .A1(pc[2]), .A2(y13), .B1(ir[2]), .B2(n1), .X(N5) );
  AO22 U9 ( .A1(pc[3]), .A2(y13), .B1(ir[3]), .B2(n1), .X(N6) );
```

```
  AO22 U10 ( .A1(pc[4]), .A2(y13), .B1(ir[4]), .B2(n1), .X(N7) );
  AO22 U11 ( .A1(pc[5]), .A2(y13), .B1(ir[5]), .B2(n1), .X(N8) );
  AO22 U12 ( .A1(y13), .A2(pc[6]), .B1(ir[6]), .B2(n1), .X(N9) );
  AO22 U13 ( .A1(pc[7]), .A2(y13), .B1(ir[7]), .B2(n1), .X(N10) );

endmodule

module operands_gpr ( A1, A2, A3, ir );

  output [3:0] A1;
  output [3:0] A2;
  output [3:0] A3;
  input [15:0] ir;
  wire   \ir[3] , \ir[2] , \ir[1] , \ir[0] , \ir[11] , \ir[10] , \ir[9] ,
         \ir[8] , ir_7, ir_6, ir_5, ir_4, n2, n3, n1;
  assign A2[3] = \ir[3] ;
  assign \ir[3]  = ir[3];
  assign A2[2] = \ir[2] ;
  assign \ir[2]  = ir[2];
  assign A2[1] = \ir[1] ;
  assign \ir[1]  = ir[1];
  assign A2[0] = \ir[0] ;
  assign \ir[0]  = ir[0];
  assign A3[3] = \ir[11] ;
  assign \ir[11]  = ir[11];
  assign A3[2] = \ir[10] ;
  assign \ir[10]  = ir[10];
  assign A3[1] = \ir[9] ;
  assign \ir[9]  = ir[9];
  assign A3[0] = \ir[8] ;
  assign \ir[8]  = ir[8];
  assign ir_7 = ir[7];
  assign ir_6 = ir[6];
  assign ir_5 = ir[5];
  assign ir_4 = ir[4];
  INV U1 ( .A(n2), .X(n1) );
  AO22 U2 ( .A1(\ir[11] ), .A2(n1), .B1(ir_7), .B2(n2), .X(A1[3]) );
  ND3_S U3 ( .A1(ir[13]), .A2(ir[12]), .A3(n3), .X(n2) );
  NR2 U4 ( .A1(ir[15]), .A2(ir[14]), .X(n3) );
```

```
  AO22 U5 ( .A1(\ir[8] ), .A2(n1), .B1(ir_4), .B2(n2), .X(A1[0]) );
  AO22 U6 ( .A1(\ir[10] ), .A2(n1), .B1(ir_6), .B2(n2), .X(A1[2]) );
  AO22 U7 ( .A1(\ir[9] ), .A2(n1), .B1(ir_5), .B2(n2), .X(A1[1]) );
endmodule

module branch ( clk, ir, y14, flags, tkn_ntk );

  input [15:0] ir;
  input [4:0] flags;
  input clk, y14;
  output tkn_ntk;
  wire   N51, n6, n7, n8, n9, n10, n11, n12, n2, n3, n4, n5;
  FDPQ_V2 tkn_ntk_reg ( .D(N51), .CK(clk), .Q(tkn_ntk) );
  AOI32 U3 ( .A1(n3), .A2(n5), .A3(n2), .B1(n10), .B2(n4), .X(n9) );
  INV U4 ( .A(flags[0]), .X(n2) );
  AO21B U5 ( .A1(n5), .A2(flags[2]), .B(n11), .X(n10) );
  AOI32 U6 ( .A1(ir[10]), .A2(flags[3]), .A3(ir[8]), .B1(flags[0]), .B2(
      n3), .X(n11) );
  INV U7 ( .A(ir[10]), .X(n3) );
  INV U8 ( .A(ir[8]), .X(n5) );
  INV U9 ( .A(ir[9]), .X(n4) );
  OA32 U10 ( .A1(ir[8]), .A2(flags[3]), .A3(n3), .B1(n12), .B2(n5), .X(
      n8) );
  AOI22 U11 ( .A1(flags[1]), .A2(n3), .B1(flags[4]), .B2(ir[10]), .X(n12) );
  OA21 U12 ( .A1(n6), .A2(n7), .B(y14), .X(N51) );
  NR4 U13 ( .A1(ir[9]), .A2(ir[8]), .A3(ir[10]), .A4(flags[2]), .X(n6));
  OAOI211 U14 ( .A1(n8), .A2(n4), .B(n9), .C(ir[11]), .X(n7) );

endmodule

module flags_set ( result, flags, c_out );

  input [15:0] result;
  output [4:0] flags;
  input c_out;
  wire   c_out, n2, n3, n4, n5, n6, n7, n8, n9, n10, n11, n12, n13, n1;
  assign flags[1] = result[15];
  assign flags[3] = c_out;
  EO2_S U1 ( .A1(result[15]), .A2(c_out), .X(flags[4]) );
  NR4 U2 ( .A1(result[1]), .A2(result[14]), .A3(result[15]), .A4(
```

```
    result[13]), .X(n12) );
  AN4B U3 ( .B1(n10), .B2(n11), .B3(n12), .A(n13), .X(flags[0]) );
  OR4B U4 ( .B1(result[2]), .B2(result[3]), .B3(result[4]), .A(n1), .X(
    n13) );
  NR4 U5 ( .A1(result[0]), .A2(result[10]), .A3(result[11]), .A4(
    result[12]), .X(n10) );
  NR4 U6 ( .A1(result[6]), .A2(result[7]), .A3(result[8]), .A4(result[9]), .X(n11) );
  EO3 U7 ( .A1(n6), .A2(n7), .A3(result[1]), .X(n5) );
  EO3 U8 ( .A1(n8), .A2(result[11]), .A3(result[10]), .X(n7) );
  EO3 U9 ( .A1(n9), .A2(result[15]), .A3(result[14]), .X(n6) );
  EO2_S U10 ( .A1(result[13]), .A2(result[12]), .X(n8) );
  EO3 U11 ( .A1(n2), .A2(result[0]), .A3(n3), .X(flags[2]) );
  EO3 U12 ( .A1(n1), .A2(n4), .A3(result[4]), .X(n3) );
  EO3 U13 ( .A1(n5), .A2(result[3]), .A3(result[2]), .X(n2) );
  EO2_S U14 ( .A1(result[7]), .A2(result[6]), .X(n4) );
  INV U15 ( .A(result[5]), .X(n1) );
  EO2_S U16 ( .A1(result[9]), .A2(result[8]), .X(n9) );

endmodule

module FSM ( clk, x1, x2, x3, x4, x5, x6, x7, y1, y2, y3, y4, y6, y7, y10, y11,
    y12, y13, y14, reset );

  input clk, x1, x2, x3, x4, x5, x6, x7, reset;
  output y1, y2, y3, y4, y6, y7, y10, y11, y12, y13, y14;
  wire   n13, n14, n15, n16, n17, n18, n19, n20, n21, n22, n23, n24, n25, n26,
      n27, n28, n29, n1, n2, n3, n4, n5, n6, n7, n8, n9, n10, n12, n30, n31,
      n32, n33, n34, n35, n36;
  wire   [3:0] state;
  wire   [3:0] next_state;
  FDNRBQ \state_reg[2]  ( .D(next_state[2]), .CK(clk), .RD(reset), .Q(
    state[2]) );
  FDNRBQ_2 \state_reg[3]  ( .D(next_state[3]), .CK(clk), .RD(reset), .Q(
    state[3]) );
  FDNRBQ \state_reg[1]  ( .D(next_state[1]), .CK(clk), .RD(reset), .Q(
    state[1]) );
  FDNRBQ \state_reg[0]  ( .D(next_state[0]), .CK(clk), .RD(reset), .Q(
    state[0]) );
  NR2_S_8 U3 ( .A1(n12), .A2(state[2]), .X(n5) );
```

```
NR2_S_16 U4 ( .A1(n27), .A2(n6), .X(y6) );
INV_S_6 U5 ( .A(n5), .X(n6) );
INV_S_1 U6 ( .A(n20), .X(n1) );
INV_S_2 U7 ( .A(n1), .X(n2) );
INV_4 U8 ( .A(state[0]), .X(n12) );
ND2_S_0P5 U9 ( .A1(state[2]), .A2(n35), .X(n20) );
ND2_S_8 U10 ( .A1(state[3]), .A2(n34), .X(n27) );
INV_4 U11 ( .A(state[1]), .X(n34) );
OAI211_0P5 U12 ( .A1(x2), .A2(n10), .B1(n16), .B2(y2), .X(n18) );
OAI31_G_0P75 U13 ( .A1(n23), .A2(x7), .A3(x2), .B(y2), .X(n24) );
NR3_T_0P5 U14 ( .A1(state[2]), .A2(state[3]), .A3(n13), .X(y2) );
NR3_G_0P65 U15 ( .A1(n13), .A2(state[2]), .A3(n35), .X(y10) );
INV_1 U16 ( .A(n12), .X(n4) );
ND2_S_1 U17 ( .A1(state[1]), .A2(n4), .X(n13) );
OR2_1 U18 ( .A1(state[2]), .A2(state[3]), .X(n3) );
ND2_S_0P5 U19 ( .A1(n29), .A2(n34), .X(n28) );
ND4_S_0P5 U20 ( .A1(x1), .A2(n12), .A3(n32), .A4(n35), .X(n25) );
ND3_0P5 U21 ( .A1(n12), .A2(n35), .A3(n21), .X(n26) );
NR3_G_0P5 U22 ( .A1(n27), .A2(n12), .A3(n32), .X(y14) );
NR2_G_2 U23 ( .A1(n29), .A2(n3), .X(y1) );
AOI211_0P5 U24 ( .A1(state[2]), .A2(n33), .B1(n7), .B2(y4), .X(n19) );
OAI31_G_0P5 U25 ( .A1(n13), .A2(state[2]), .A3(n14), .B(n31), .X(
    next_state[3]) );
NR3_G_0P5 U26 ( .A1(n34), .A2(n4), .A3(n2), .X(y4) );
NR2_0P5 U27 ( .A1(n34), .A2(state[2]), .X(n21) );
ND2_S_0P5 U28 ( .A1(n4), .A2(n34), .X(n29) );
INV_S_0P5 U29 ( .A(state[2]), .X(n32) );
INV_1 U30 ( .A(state[3]), .X(n35) );
INV_1 U31 ( .A(n26), .X(y13) );
INV_1 U32 ( .A(y3), .X(n30) );
NR2_1 U33 ( .A1(n13), .A2(n2), .X(y7) );
INV_1 U34 ( .A(y10), .X(n31) );
INV_1 U35 ( .A(x2), .X(n8) );
ND2B_S_1 U36 ( .A(x6), .B(n10), .X(n23) );
NR2_1 U37 ( .A1(n2), .A2(n29), .X(y3) );
INV_1 U38 ( .A(x3), .X(n10) );
NR3_G_1 U39 ( .A1(n28), .A2(n35), .A3(n32), .X(y11) );
NR2_1 U40 ( .A1(n2), .A2(n28), .X(y12) );
```

```
  OAI311_1 U41 ( .A1(n12), .A2(state[3]), .A3(n21), .B1(n19), .B2(n22),
        .X(next_state[1]) );
  AOI31_1 U42 ( .A1(n23), .A2(n8), .A3(y2), .B(y13), .X(n22) );
  ND4_1 U43 ( .A1(n19), .A2(n24), .A3(n25), .A4(n26), .X(next_state[0]) );
  OAOI211_1 U44 ( .A1(n15), .A2(x3), .B(n8), .C(n9), .X(n14) );
  NR2B_V1DG_1 U45 ( .A(x7), .B(x6), .X(n15) );
  INV_1 U46 ( .A(n16), .X(n9) );
  ND2_S_1 U47 ( .A1(x5), .A2(x2), .X(n16) );
  ND3B_V1DG_1 U48 ( .A(n17), .B1(n18), .B2(n19), .X(next_state[2]) );
  OAI311_1 U49 ( .A1(n36), .A2(state[1]), .A3(n2), .B1(n31), .B2(n30), .X(n17) );
  INV_1 U50 ( .A(x1), .X(n36) );
  INV_S_0P5 U51 ( .A(n27), .X(n33) );
  NR3_G_0P5 U52 ( .A1(n12), .A2(state[2]), .A3(n27), .X(n7) );

Endmodule

module ALU_DW01_inc_0 ( A, SUM );

  input [15:0] A;
  output [15:0] SUM;
  wire   [15:2] carry;
  ADDH U1_1_14 ( .A(A[14]), .B(carry[14]), .CO(carry[15]), .S(SUM[14]));
  ADDH U1_1 ( .A(A[1]), .B(A[0]), .CO(carry[2]), .S(SUM[1]) );
  ADDH U1_1_2 ( .A(A[2]), .B(carry[2]), .CO(carry[3]), .S(SUM[2]) );
  ADDH U1_1_8 ( .A(A[8]), .B(carry[8]), .CO(carry[9]), .S(SUM[8]) );
  ADDH U1_1_9 ( .A(A[9]), .B(carry[9]), .CO(carry[10]), .S(SUM[9]) );
  ADDH U1_1_12 ( .A(A[12]), .B(carry[12]), .CO(carry[13]), .S(SUM[12]));
  ADDH U1_1_11 ( .A(A[11]), .B(carry[11]), .CO(carry[12]), .S(SUM[11]));
  ADDH U1_1_5 ( .A(A[5]), .B(carry[5]), .CO(carry[6]), .S(SUM[5]));
  ADDH U1_1_13 ( .A(A[13]), .B(carry[13]), .CO(carry[14]), .S(SUM[13]));
  ADDH U1_1_10 ( .A(A[10]), .B(carry[10]), .CO(carry[11]), .S(SUM[10]));
  ADDH U1_1_6 ( .A(A[6]), .B(carry[6]), .CO(carry[7]), .S(SUM[6]) );
  ADDH U1_1_4 ( .A(A[4]), .B(carry[4]), .CO(carry[5]), .S(SUM[4]) );
  ADDH U1_1_3 ( .A(A[3]), .B(carry[3]), .CO(carry[4]), .S(SUM[3]) );
  ADDH U1_1_7 ( .A(A[7]), .B(carry[7]), .CO(carry[8]), .S(SUM[7]) );
  INV U1 ( .A(A[0]), .X(SUM[0]) );
  EO2_0P5 U2 ( .A1(carry[15]), .A2(A[15]), .X(SUM[15]) );

Endmodule
```

```
module program_counter_DW01_inc_0 ( A, SUM );

  input [7:0] A;
  output [7:0] SUM;
  wire   [7:2] carry;
  ADDH_1 U1_1_6 ( .A(A[6]), .B(carry[6]), .CO(carry[7]), .S(SUM[6]) );
  ADDH_1 U1_1_5 ( .A(A[5]), .B(carry[5]), .CO(carry[6]), .S(SUM[5]) );
  ADDH_1 U1_1_4 ( .A(A[4]), .B(carry[4]), .CO(carry[5]), .S(SUM[4]) );
  ADDH_1 U1_1_3 ( .A(A[3]), .B(carry[3]), .CO(carry[4]), .S(SUM[3]) );
  ADDH_1 U1_1_2 ( .A(A[2]), .B(carry[2]), .CO(carry[3]), .S(SUM[2]) );
  ADDH_1 U1_1_1 ( .A(A[1]), .B(A[0]), .CO(carry[2]), .S(SUM[1]) );
  INV_1 U1 ( .A(A[0]), .X(SUM[0]) );
  EO2_0P5 U2 ( .A1(carry[7]), .A2(A[7]), .X(SUM[7]) );

endmodule

module program_counter ( clk, y1, y7, pc, pc0, tkn_ntk, ir );

  output [7:0] pc;
  input [7:0] pc0;
  input [15:0] ir;
  input clk, y1, y7, tkn_ntk;
  wire   N6, N7, N8, N9, N10, N11, N12, N13, n1, n2, n3, n4, n5, n6, n7, n8,
         n9, n10, n11, n12, n13, n14, n15, n16, n17, n18, n19, n20, n21, n22,
         n23, n24, n25, n26, n27, n28, n29, n30, n31, n32, n33, n34, n35, n36;

  program_counter_DW01_inc_0 add_449_S2 ( .A(pc), .SUM({N13, N12, N11, N10, N9,
       N8, N7, N6}) );
  FDPQB_V2 \pc_reg[0]  ( .D(n36), .CK(clk), .QN(n8) );
  FDPQB_V2 \pc_reg[1]  ( .D(n35), .CK(clk), .QN(n7) );
  FDPQB_V2 \pc_reg[2]  ( .D(n34), .CK(clk), .QN(n6) );
  FDPQB_V2 \pc_reg[3]  ( .D(n33), .CK(clk), .QN(n5) );
  FDPQB_V2 \pc_reg[4]  ( .D(n32), .CK(clk), .QN(n4) );
  FDPQB_V2 \pc_reg[5]  ( .D(n31), .CK(clk), .QN(n3) );
  FDPQB_V2 \pc_reg[6]  ( .D(n30), .CK(clk), .QN(n2) );
  FDPQB_V2 \pc_reg[7]  ( .D(n29), .CK(clk), .QN(n1) );
  NR2_S U3 ( .A1(n13), .A2(y1), .X(n28) );
  AOI22_T_0P5 U4 ( .A1(pc0[1]), .A2(y1), .B1(pc[1]), .B2(n13), .X(n24) );
  AOI22_0P5 U5 ( .A1(pc0[3]), .A2(y1), .B1(pc[3]), .B2(n13), .X(n20) );
  AOI22_0P5 U6 ( .A1(pc0[5]), .A2(y1), .B1(pc[5]), .B2(n13), .X(n16) );
```

```
AOI22_0P5 U7 ( .A1(pc0[7]), .A2(y1), .B1(pc[7]), .B2(n13), .X(n9) );
NR2_G_2 U8 ( .A1(y1), .A2(y7), .X(n13) );
AN2_MG U9 ( .A1(tkn_ntk), .A2(n28), .X(n12) );
NR2B_V1DG U10 ( .A(n28), .B(tkn_ntk), .X(n11) );
INV U11 ( .A(n8), .X(pc[0]) );
INV U12 ( .A(n2), .X(pc[6]) );
INV U13 ( .A(n7), .X(pc[1]) );
INV U14 ( .A(n6), .X(pc[2]) );
INV U15 ( .A(n5), .X(pc[3]) );
INV U16 ( .A(n4), .X(pc[4]) );
INV U17 ( .A(n3), .X(pc[5]) );
INV U18 ( .A(n1), .X(pc[7]) );
ND2_S U19 ( .A1(n9), .A2(n10), .X(n29) );
AOI22 U20 ( .A1(N13), .A2(n11), .B1(ir[7]), .B2(n12), .X(n10) );
ND2_S U21 ( .A1(n14), .A2(n15), .X(n30) );
AOI22 U22 ( .A1(pc0[6]), .A2(y1), .B1(pc[6]), .B2(n13), .X(n14) );
AOI22 U23 ( .A1(N12), .A2(n11), .B1(ir[6]), .B2(n12), .X(n15) );
ND2_S U24 ( .A1(n16), .A2(n17), .X(n31) );
AOI22 U25 ( .A1(N11), .A2(n11), .B1(ir[5]), .B2(n12), .X(n17) );
ND2_S U26 ( .A1(n18), .A2(n19), .X(n32) );
AOI22 U27 ( .A1(pc0[4]), .A2(y1), .B1(pc[4]), .B2(n13), .X(n18) );
AOI22 U28 ( .A1(N10), .A2(n11), .B1(ir[4]), .B2(n12), .X(n19) );
ND2_S U29 ( .A1(n20), .A2(n21), .X(n33) );
AOI22 U30 ( .A1(N9), .A2(n11), .B1(ir[3]), .B2(n12), .X(n21) );
ND2_S U31 ( .A1(n22), .A2(n23), .X(n34) );
AOI22 U32 ( .A1(pc0[2]), .A2(y1), .B1(pc[2]), .B2(n13), .X(n22) );
AOI22 U33 ( .A1(N8), .A2(n11), .B1(ir[2]), .B2(n12), .X(n23) );
ND2_S U36 ( .A1(n26), .A2(n27), .X(n36) );
AOI22 U37 ( .A1(pc0[0]), .A2(y1), .B1(pc[0]), .B2(n13), .X(n26) );
AOI22 U38 ( .A1(N6), .A2(n11), .B1(ir[0]), .B2(n12), .X(n27) );

endmodule
```

ՀԱՎԵԼՎԱԾ 3

- **PWR Compiler ծրագրային միջոցի նկարագրությունից հատվածներ**

```
  class Ui_MainWindow_1(object):
def setupUi(self, MainWindow):
    MainWindow.setObjectName("MainWindow")
    MainWindow.resize(916, 728)
```

```
MainWindow.setLayoutDirection(QtCore.Qt.LeftToRight)
self.centralWidget = QtWidgets.QWidget(MainWindow)
self.centralWidget.setObjectName("centralWidget")
self.groupBox = QtWidgets.QGroupBox(self.centralWidget)
self.groupBox.setGeometry(QtCore.QRect(20, 20, 781, 321))
font = QtGui.QFont()
font.setPointSize(10)
self.groupBox.setFont(font)
self.groupBox.setObjectName("groupBox")
self.label = QtWidgets.QLabel(self.groupBox)
self.label.setGeometry(QtCore.QRect(20, 30, 81, 16))
self.label.setText("")
self.label.setObjectName("label")
self.textEdit = QtWidgets.QTextEdit(self.groupBox)
self.textEdit.setGeometry(QtCore.QRect(280, 20, 431, 191))
self.textEdit.setObjectName("textEdit")
self.checkBox = QtWidgets.QCheckBox(self.groupBox)
self.checkBox.setGeometry(QtCore.QRect(20, 60, 241, 20))
font = QtGui.QFont()
font.setPointSize(12)
self.checkBox.setFont(font)
self.checkBox.setObjectName("checkBox")
self.checkBox_2 = QtWidgets.QCheckBox(self.groupBox)
self.checkBox_2.setGeometry(QtCore.QRect(20, 110, 241, 20))
font = QtGui.QFont()
font.setPointSize(12)
self.checkBox_2.setFont(font)
self.checkBox_2.setObjectName("checkBox_2")
self.pushButton = QtWidgets.QPushButton(self.groupBox)
self.pushButton.setGeometry(QtCore.QRect(20, 220, 161, 31))
font = QtGui.QFont()
font.setPointSize(9)
self.pushButton.setFont(font)
self.pushButton.setObjectName("pushButton")
self.checkBox_3 = QtWidgets.QCheckBox(self.groupBox)
self.checkBox_3.setGeometry(QtCore.QRect(20, 160, 241, 20))
font = QtGui.QFont()
font.setPointSize(12)
self.checkBox_3.setFont(font)
self.checkBox_3.setObjectName("checkBox_3")
self.groupBox_2 = QtWidgets.QGroupBox(self.groupBox)
self.groupBox_2.setGeometry(QtCore.QRect(480, 320, 781, 321))
font = QtGui.QFont()
font.setPointSize(10)
self.groupBox_2.setFont(font)
self.groupBox_2.setObjectName("groupBox_2")
self.label_2 = QtWidgets.QLabel(self.groupBox_2)
self.label_2.setGeometry(QtCore.QRect(20, 30, 81, 16))
self.label_2.setText("")
self.label_2.setObjectName("label_2")
self.textEdit_2 = QtWidgets.QTextEdit(self.groupBox_2)
self.textEdit_2.setGeometry(QtCore.QRect(280, 10, 431, 191))
self.textEdit_2.setObjectName("textEdit_2")
self.checkBox_4 = QtWidgets.QCheckBox(self.groupBox_2)
self.checkBox_4.setGeometry(QtCore.QRect(20, 60, 241, 20))
font = QtGui.QFont()
font.setPointSize(12)
```

```
        self.checkBox_4.setFont(font)
        self.checkBox_4.setObjectName("checkBox_4")
        self.checkBox_5 = QtWidgets.QCheckBox(self.groupBox_2)
        self.checkBox_5.setGeometry(QtCore.QRect(20, 110, 241, 20))
        font = QtGui.QFont()
        font.setPointSize(12)
        self.checkBox_5.setFont(font)
        self.checkBox_5.setObjectName("checkBox_5")
        self.pushButton_2 = QtWidgets.QPushButton(self.groupBox_2)
        self.pushButton_2.setGeometry(QtCore.QRect(20, 220, 141, 31))
        font = QtGui.QFont()
        font.setPointSize(9)
        self.pushButton_2.setFont(font)
        self.pushButton_2.setObjectName("pushButton_2")
        self.checkBox_6 = QtWidgets.QCheckBox(self.groupBox_2)
        self.checkBox_6.setGeometry(QtCore.QRect(20, 160, 241, 20))
        font = QtGui.QFont()
        font.setPointSize(12)
        self.checkBox_6.setFont(font)
        self.checkBox_6.setObjectName("checkBox_6")
        self.progressBar = QtWidgets.QProgressBar(self.centralWidget)
        self.progressBar.setGeometry(QtCore.QRect(60, 290, 161, 31))
        self.progressBar.setProperty("value", 57)
        self.progressBar.setObjectName("progressBar")
        self.groupBox_3 = QtWidgets.QGroupBox(self.centralWidget)
        self.groupBox_3.setGeometry(QtCore.QRect(20, 360, 781, 311))
        font = QtGui.QFont()
        font.setPointSize(10)
        self.groupBox_3.setFont(font)
        self.groupBox_3.setObjectName("groupBox_3")
        self.label_3 = QtWidgets.QLabel(self.groupBox_3)
        self.label_3.setGeometry(QtCore.QRect(20, 30, 81, 16))
        self.label_3.setText("")
        self.label_3.setObjectName("label_3")
        self.textEdit_3 = QtWidgets.QTextEdit(self.groupBox_3)
        self.textEdit_3.setGeometry(QtCore.QRect(280, 110, 431, 191))
        self.textEdit_3.setObjectName("textEdit_3")
        self.checkBox_7 = QtWidgets.QCheckBox(self.groupBox_3)
        self.checkBox_7.setGeometry(QtCore.QRect(20, 70, 241, 21))
        font = QtGui.QFont()
        font.setPointSize(12)
        self.checkBox_7.setFont(font)
        self.checkBox_7.setObjectName("checkBox_7")
        self.checkBox_8 = QtWidgets.QCheckBox(self.groupBox_3)
        self.checkBox_8.setGeometry(QtCore.QRect(20, 110, 241, 20))
        font = QtGui.QFont()
        font.setPointSize(12)
        self.checkBox_8.setFont(font)
        self.checkBox_8.setObjectName("checkBox_8")
        self.pushButton_3 = QtWidgets.QPushButton(self.groupBox_3)
        self.pushButton_3.setGeometry(QtCore.QRect(20, 220, 181, 31))
        font = QtGui.QFont()
        font.setPointSize(9)
        self.pushButton_3.setFont(font)
        self.pushButton_3.setObjectName("pushButton_3")
        self.checkBox_9 = QtWidgets.QCheckBox(self.groupBox_3)
        self.checkBox_9.setGeometry(QtCore.QRect(20, 150, 241, 20))
```

```
font = QtGui.QFont()
font.setPointSize(12)
self.checkBox_9.setFont(font)
self.checkBox_9.setObjectName("checkBox_9")
self.label_4 = QtWidgets.QLabel(self.groupBox_3)
self.label_4.setGeometry(QtCore.QRect(10, 30, 111, 16))
self.label_4.setObjectName("label_4")
self.checkBox_10 = QtWidgets.QCheckBox(self.groupBox_3)
self.checkBox_10.setGeometry(QtCore.QRect(20, 190, 241, 20))
font = QtGui.QFont()
font.setPointSize(12)
self.checkBox_10.setFont(font)
self.checkBox_10.setObjectName("checkBox_10")
self.checkBox_11 = QtWidgets.QCheckBox(self.groupBox_3)
self.checkBox_11.setGeometry(QtCore.QRect(190, 30, 241, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.checkBox_11.setFont(font)
self.checkBox_11.setObjectName("checkBox_11")
self.checkBox_12 = QtWidgets.QCheckBox(self.groupBox_3)
self.checkBox_12.setGeometry(QtCore.QRect(420, 30, 241, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.checkBox_12.setFont(font)
self.checkBox_12.setObjectName("checkBox_12")
self.checkBox_13 = QtWidgets.QCheckBox(self.groupBox_3)
self.checkBox_13.setGeometry(QtCore.QRect(340, 70, 241, 31))
font = QtGui.QFont()
font.setPointSize(12)
self.checkBox_13.setFont(font)
self.checkBox_13.setObjectName("checkBox_13")
self.progressBar_2 = QtWidgets.QProgressBar(self.centralWidget)
self.progressBar_2.setGeometry(QtCore.QRect(60, 630, 161, 31))
self.progressBar_2.setProperty("value", 37)
self.progressBar_2.setObjectName("progressBar_2")
MainWindow.setCentralWidget(self.centralWidget)
self.menuBar = QtWidgets.QMenuBar(MainWindow)
self.menuBar.setGeometry(QtCore.QRect(0, 0, 916, 26))
self.menuBar.setObjectName("menuBar")
self.menuFile = QtWidgets.QMenu(self.menuBar)
self.menuFile.setObjectName("menuFile")
self.menuEdit = QtWidgets.QMenu(self.menuBar)
self.menuEdit.setObjectName("menuEdit")
self.menuView = QtWidgets.QMenu(self.menuBar)
self.menuView.setObjectName("menuView")
self.menuhelp = QtWidgets.QMenu(self.menuBar)
self.menuhelp.setObjectName("menuhelp")
MainWindow.setMenuBar(self.menuBar)
self.mainToolBar = QtWidgets.QToolBar(MainWindow)
self.mainToolBar.setObjectName("mainToolBar")
MainWindow.addToolBar(QtCore.Qt.TopToolBarArea, self.mainToolBar)
self.statusBar = QtWidgets.QStatusBar(MainWindow)
self.statusBar.setObjectName("statusBar")
MainWindow.setStatusBar(self.statusBar)
self.menuBar.addAction(self.menuFile.menuAction())
self.menuBar.addAction(self.menuEdit.menuAction())
self.menuBar.addAction(self.menuView.menuAction())
```

```
        self.menuBar.addAction(self.menuhelp.menuAction())

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "PWR
Compiler"))
        self.groupBox.setTitle(_translate("MainWindow", "Low power
synthesis"))
        self.textEdit.setHtml(_translate("MainWindow", "<!DOCTYPE HTML
PUBLIC \"-//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-
html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style
type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'MS Shell Dlg 2\'; font-
size:10pt; font-weight:400; font-style:normal;\">\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-
indent:0px;\">================================</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">Analyzing results of
power estimation ...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">mapping mentioned
methods ...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">method 1 integration
...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">method 3, 5 integration
...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">method 7 integration
...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">checking results
....</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">archiving design
</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-
indent:0px;\">================================</p></body></html>"))
        self.checkBox.setText(_translate("MainWindow", "High effort"))
        self.checkBox_2.setText(_translate("MainWindow", "Middle
effort"))
        self.pushButton.setText(_translate("MainWindow", "low power
synthesis"))
        self.checkBox_3.setText(_translate("MainWindow", "Low effort"))
        self.groupBox_2.setTitle(_translate("MainWindow", "Virtual power
estimator"))
        self.textEdit_2.setHtml(_translate("MainWindow", "<!DOCTYPE HTML
PUBLIC \"-//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-
html40/strict.dtd\">\n"
```

```
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style
type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'MS Shell Dlg 2\'; font-
size:10pt; font-weight:400; font-style:normal;\">\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-
indent:0px;\">================================</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">Analyzing results of
power estimation ...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">mapping mentioned
methods ...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">method 1 integration
...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">method 3, 5 integration
...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">method 7 integration
...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">checking results
....</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">archiving design
</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-
indent:0px;\">================================</p></body></html>"))
        self.checkBox_4.setText(_translate("MainWindow", "High effort"))
        self.checkBox_5.setText(_translate("MainWindow", "Middle
effort"))
        self.pushButton_2.setText(_translate("MainWindow", "low power
synthesis"))
        self.checkBox_6.setText(_translate("MainWindow", "Low effort"))
        self.groupBox_3.setTitle(_translate("MainWindow", "Low power re-
synthesis"))
        self.textEdit_3.setHtml(_translate("MainWindow", "<!DOCTYPE HTML
PUBLIC \"-//W3C//DTD HTML 4.0//EN\" \"http://www.w3.org/TR/REC-
html40/strict.dtd\">\n"
"<html><head><meta name=\"qrichtext\" content=\"1\" /><style
type=\"text/css\">\n"
"p, li { white-space: pre-wrap; }\n"
"</style></head><body style=\" font-family:\'MS Shell Dlg 2\'; font-
size:10pt; font-weight:400; font-style:normal;\">\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-
indent:0px;\">================================</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">Re-analyzing results of
power estimation ...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">Re-mapping mentioned
methods ...</p>\n"
```

```
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">Re-method 1 integration
...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">Re-method 3, 5
integration ...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">Re-method 7 integration
...</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">Re-checking results
....</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-indent:0px;\">Re-archiving design
</p>\n"
"<p style=\" margin-top:0px; margin-bottom:0px; margin-left:0px; margin-
right:0px; -qt-block-indent:0; text-
indent:0px;\">================================</p></body></html>"))
        self.checkBox_7.setText(_translate("MainWindow", "Method 1"))
        self.checkBox_8.setText(_translate("MainWindow", "Method 2"))
        self.pushButton_3.setText(_translate("MainWindow", "low power re-
synthesis"))
        self.checkBox_9.setText(_translate("MainWindow", "Method 3"))
        self.label_4.setText(_translate("MainWindow", "excludes"))
        self.checkBox_10.setText(_translate("MainWindow", "Method 4"))
        self.checkBox_11.setText(_translate("MainWindow", "Suggested
method 5"))
        self.checkBox_12.setText(_translate("MainWindow", "Suggested
method 6"))
        self.checkBox_13.setText(_translate("MainWindow", "suggested
method 7"))
        self.menuFile.setTitle(_translate("MainWindow", "File"))
        self.menuEdit.setTitle(_translate("MainWindow", "Edit"))
        self.menuView.setTitle(_translate("MainWindow", "View"))
        self.menuhelp.setTitle(_translate("MainWindow", "Help"))
     endclass
```

ՀԱՎԵԼՎԱԾ 4

Նկարների ցանկ

Աղյուսակների ցանկ

Հապավումների ցանկ

ԿՄՕԿ - կոմպլեմենտար մետաղ-օքսիդ-կիսահաղորդիչ

ԻՍ – ինտեգրալ սխեմա

ԿԲ – կիսահաղորդչային բյուրեղ

ՀՀԴ- համապիտանի հաջորդական դող

ՄՕԿ- մետաղ-օքսիդ-կիսահաղորդիչ

ԹՏԳՀ- թվաբանական տրամաբանական գործողությունների հանգույց

ԴԼՀՄ – դինամիկ լարման և հաճախության մասշտաբավորում

ՄՀ – Մեկուսացնող հանգույց

ՀՇՄ – հզորության շրջափակման մեթոդ

ՀՇ – հզորության շրջափակում

ՏԱՇ – տակտային ազդանշանի շրջափակում

ՎՊՎ – վիճակը պահպանող և վերականգնող

ՏԱ- տակտային ազդանշան

ՄՀՁ – միասնական հզորության ձևաչափ

ԼՄՓ – լարման մակարդակի փոխակերպիչ

ԶԱՀԶՀ – զուգահեռ ազդանշանը հաջորդականի ձևափոխման հանգույց

ՀԱԶԶՀ – հաջորդական ազդանշանը զուգահեռի ձևափոխման հանգույց

ՕԾ - օպերացիոն ծանրաբեռնվածություն

ՋՄՀ – ջերմաստիճանի մասշտաբավորման հանգույց

ՑԷ – ցածր էներգասպառման

ՍՆԼ – Սխեմաներ նկարագրման լեզու