ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

ԱՆՆԱ ԿԱՐՊԻՍԻ ՍԱՂԱԹԵԼՅԱՆ

ՄՈԴՈՒԼՅԱՐ ԹՎԱԲԱՆՈՒԹՅԱՆ ԲԼՈԿՈՒՄ ԹՎԱԲԱՆԱԿԱՆ ՍԱՐՔԵՐԻ ՆԱԽԱԳԾՄԱՆ ՄԵԹՈԴՆԵՐԻ ՄՇԱԿՈՒՄ ԵՎ ՀԵՏԱԶՈՏՈՒՄ

Ե 13.03 - «Հաշվողական մեքենաներ, համալիրներ, համակարգեր, ցանցեր, դրանց տարրերը և սարքավորումները» մասնագիտությամբ տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման

ԱՏԵՆԱԽՈՍՈՒԹՅՈՒՆ

Գիտական ղեկավար՝ տ.գ.դ., պրոֆեսոր Գ.Տ. Կիրակոսյան

Երևան 2015

ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆ

ՆԵՐԱԾՈՒԹՅՈՒՆ

Ատենախոսության արդիականությունը

Թվային սարքերին միշտ ներկայացնում են արագագործության բարձրացման պահանջներ, և մոդուլյար թվաբանությունը դիտարկվում է որպես արագագործության բարձրացման և հաշվարկների ապահովության միջոց: Մոդուլյար թվաբանության օգտագործումը հնարավորություն է տալիս էականորեն բարձրացնել թվաբանական հաշվարկների արագությունը մնացորդների հետ գործողությունների զուգընթաց կատարման շնորհիվ: Չնայած մոդուլյար թվաբանության բնագավառում կատարվել են բավականին շատ տեսական հետազոտություններ այս ուղղությունը լայն զարգացում չի ստացել այսպիսի սարքերն իրականացնող տարրային բազայի բացակայության պատճառով: Ինտեգրալային սխեմատեխնիկայի զարգացմանը զուգընթաց ստեղծվեցին մոդուլային թվաբանության բլոկում նմանատիպ սարքերի իրագործման հնարավորություններ:

Ատենախոսությունը նվիրված է ինտեգրալային իրականացմամբ, բյուրեղի վրա համակարգերի (System-on-Chip, SoC) կառուցման համար բարդ ֆունկցիոնալ բլոկների (ԲՖ-բլոկներ) նախագծման մեթոդների մշակմանը և հետազոտմանը՝ մոդուլյար թվաբանության ապարատի կիրառմամբ:

Բարդ ինտեգրալային սխեմաների կառուցման արդյունավետ մոտեցումներից է SoC (մասնագիտացված բարդ բլոկների) մեթոդաբանությունը, որի նախագծման արդյունավետության բարձրացման հիմնական մեթոդը, նախապես մշակված և թեստավորում անցած ֆունկցիոնալ բլոկների բազմակի օգտագործումն է: Ֆունկցիոնալ բլոկների համար լայնորեն կիրառվում է նաև «Բարդ Ֆունկցիոնալ բլոկ» (ԲՖ բլոկ) տերմինը: Սակայն մոդուլյար թվաբանության ապարատում այսպիսի մասնագիտացված ԲՖ բլոկների կիրառումը սահմանափակվում է երկու պատճառով.

- Ժամանակակից ավտոմատ նախագծման համակարգերի արտադրողները աջակցում են տարբեր մակարդակների և բարդության դիրքային հաշվողական համակարգերի ստեղծմանը:

- Ժամանակակից ապարատային բազան հնարավորություն է տալիս հանրահաշվական գործողությունները փոխարինել մնացորդների հետ միատակտ աղյուսակային ընտրանքներով:

Կարևոր է նշել, որ մոդուլյար թվաբանության օգտագործմամբ մասնագիտացված ԲՖ բլոկների առկայությունն ընդլայնում է մոդուլյար թվաբանության կիրառման ոլորտները. օրինակ՝ ինֆորմացիայի թվային և թվաբանական հսկում և արատորոշում, ինֆորմացիայի գաղտնագրում և այլն: Այսպիսով, կարելի է պնդել, որ մոդուլյար մասնագիտացված ԲՖ բլոկների նախագծումը և կրկնակի օգտագործումը, ինչպես նաև ապարատային իրագործման նոր մեթոդների որոնումը արդիական գիտատեխնիկական խնդիր է:

Ատենախոսությունում մոդուլյար թվաբանությունը դիտարկվում է որպես հաշվարկների արագագործությունը և հուսալիությունը բարձրացնող գործիքամիջոց: Ունենալով զուգահեռացման հնարավորություն՝ արդյունավետության բարձրացման համար մոդուլյար հաշվարկների կատարման համար պահանջում է մոդուլյար կարգերում (ուղիներում) թվաբանական գործողությունների կատարման հատուկ սխեմաների մշակում: Դիտարկվում են մնացորդային դասերի համակարգում հաշվարկների հիմնական առանձնահատկություններն ու առավելությունները, ուսումնասիրվում են հիմնական արդյունքները, որոնք ստացվել են այս հեռանկարային ուղղությունում:

<u>Ատենախոսության նպատակը:</u> Ատենախոսության նպատակն է՝ հետազոտել և մշակել մոդուլյար թվաբանության բլոկում մասնագիտացված ԲՖ բլոկների միջոցով նախագծման նոր մեթոդներ և միավորել դրանք մեկ երկխոսային ավտոմատացված համակարգում, որի օգնությամբ հնարավոր է իրականացնել թվային սարքերի արդյունավետ կառուցվածքների մոդելավորում (սիմուլյացիա) և սինթեզում:

Այս նպատակով ուսումնասիրվել և վերլուծվել են՝

- մոդուլյար թվաբանության հիմնական օրենքները և նախագծման արդի մեթոդները,
- մոդուլյար հաշվարկների իրականացման եղանակները,

- մոդուլյար թվաբանության բլոկում թվաբանական գործողությունների կատարման ալգորիթմները և մեթոդները,
- թվային սարքերի ժամանակակից միջոցները և կիրառման առանձնահատկությունները:

Լուծվել են հետևյալ խնդիրները՝

- Հետազոտել թվային սարքերի նախագծման ժամանակակից մեթոդները և դրանց կիրառման առանձնահատկությունները:
- Մշակել տարբեր մեթոդներով մոդուլյար գումարիչների, բազմապատկիչների և մոդուլով աստիճան բարձրացնելու սարքերի բարդ ֆունկցիոնալ բլոկների միջոցով ապարատային իրականացումները, ինչպես նաև գնահատել դրանց արագագործությունը և ապարատային ծախսերը:
- Վերլուծել մոդուլյար ԲՖ բլոկների ապարատային կառուցվածքները՝ կախված մոդուլի արժեքից և կիրառման ոլորտից:
- Մշակել մոդուլյար թվաբանության բլոկում մասնագիտացված ԲՖ բլոկների միջոցով նախագծման մեթոդների միավորված ավտոմատացված համակարգ:

<u>Աշխատանքի գիտական նորույթը</u>՝ մոդուլյար համակարգերի ինտեգրալային իրագործման համար հիմնական հաշվողական հանգույցների, մասնագիտացված ԲՖ-բլոկների նախագծման մեթոդների և մոդուլյար թվաբանության բլոկի համար այնպիսի նոր մեթոդների մշակումն է, որոնք միավորված են թվային սարքերի արդյունավետ կառուցվածքների մոդելավորման և սինթեզման՝ առաջարկված ավտոմատացված համակարգում: Ստացվել են հետևյալ արդյունքները.

- Հիմնավորվել է թվային սարքերի սինթեզման Xilinx ընկերության ISE Design Suite փաթեթի ընտրությունը, որը պարունակում է ModelSim մոդելավորող և XST սինթեզող մակարդակները:
- Մշակվել են տարբեր մեթոդներով մոդուլյար գումարիչների, բազմապատկիչների և մոդուլով աստիճան բարձրացնելու սարքերի ԲՖ բլոկների միջոցով ապարատային իրացումները:

- Գնահատվել են նախագծված մոդուլյար սարքերի արագագործությունը և ապարատային ծախսերը՝ կախված մուտքային թվերի ու մոդուլի արժեքի կարգայնությունից (մեծությունից):
- Առաջարկվել են մոդուլյար թվաբանության բլոկում թվաբանական սարքերի նախագծման արդյունավետ մեթոդի ընտրության մոտեցումները:
- Մշակվել է մոդուլյար թվաբանության բլոկում մասնագիտացված ԲՖ բլոկների միջոցով նախագծման մեթոդների միավորված ավտոմատացված համակարգ:

<u>Մշակված մեթոդների հետազոտումները:</u> Ատենախոսությունում օգտագործվել են թվերի տեսության, դիսկրետ մաթեմատիկայի և բուլյան հանրահաշվի հիմունքները, թվային սարքերի կառուցման տեսական և գործնական միջոցները, ինչպես նաև ժամանակակից տրամաբանական սինթեզի և քոմփյութերային մոդելավորման միջոցները:

<u>Պաշտպանությանը ներկայացվում են հետևյալ արդյունքներ.</u>

1. Մոդուլյար գումարիչների կառուցման տարբեր մեթոդներ և այդ մեթոդների կիրառման արդյունավետությունը՝ կախված մուտքային թվերի և մոդուլի արժեքի կարգայնությունից:
2. Մոդուլյար բազմապատկիչների կառուցման տարբեր մեթոդներ և այդ մեթոդների կիրառման արդյունավետությունը՝ կախված մուտքային թվերի և մոդուլի արժեքի կարգայնությունից:
3. Մոդուլով աստիճան բարձրացումը իրականացնող սարքերի կառուցման մեթոդներ և այդ մեթոդների կիրառման արդյունավետությունը՝ կախված մուտքային թվերի և մոդուլի արժեքի կարգայնությունից:
4. Մոդուլյար թվաբանության բլոկում մասնագիտացված թվային սարքերի ԲՖ-բլոկների միջոցով նախագծման մեթոդների միավորված ավտոմատացված համակարգը:

<u>Ատենախոսության արդյունքում</u> հետազոտվել, վերլուծվել և մշակվել են մոդուլյար թվաբանության բլոկի (ապարատի) կիրառմամբ թվաբանական սարքերի նախագծման մեթոդները, նաև մոդուլյար բլոկում մասնագիտացված ԲՖ բլոկների (ինչպիսիք են՝

մոդուլյար գումարիչները, մոդուլների տարբեր արժեքների համար և տարբեր մեթոդներով իրագործված մոդուլյար բազմապատկիչները, մոդուլով աստիճան բարձրացման սարքերը) վրա կառուցման սկզբունքները: Առաջարկված մեթոդներով մշակվել են սարքերի նկարագրությունները Verilog HDL-ով, որոնք միավորվել են մոդուլյար թվաբանության բլոկում սարքերի նախագծման արդյունավետությունը բարձրացնող ավտոմատացված համակարգով:

Կիրառական նշանակությունը

Աշխատանքում ստացված արդյունքները կարող են կիրառվել մասնագիտացված բարդ սարքերի, ինչպիսիք են՝ ազդանշանների թվային մշակման, գաղտնագրման, կոդերի հսկման համակարգերի և մոդուլյար թվաբանության վրա հիմնված այլ սարքերի նախագծման համար: Մշակված մեթոդների կիրառումը հնարավորություն կընձեռնի բարձրացնել նշված համակարգերի նախագծման արդյունավետությունը և զգալիորեն կրճատել նախագծման ժամանակային ծախսերը:

RSA ալգորիթմի հիման վրա աշխատանքում մշակված գաղտնագրման սարքը ներդրվել և օգտագործվում է “Երևանի կապի միջոցների ԳՀԻ” ՓԲԸ–ում և “Ֆեմբոքս” ՍՊԸ-ում, իսկ “Ծրագրավորվող տրամաբանական սարքեր” մեթոդական ցուցումները և դրանց վրա հիմնված լաբորատոր աշխատանքներն ու թվաբանության բլոկում սարքերի նախագծման արդյունավետությունը բարձրացնող ավտոմատացված համակարգը ներդրվել են Հայաստանի ազգային պոլիտեխնիկական համալսարանի (ՀԱՊՀ) Քոմփյութերային համակարգեր և ցանցեր (ՔՀ և Ց) ամբիոնում:

Հրատարակված աշխատանքները

Ատենախոսության հիմնական արդյունքները զեկուցվել են ՀԱՊՀ տարեկան գիտաժողովներում (2010, 2011, 2013), Գերմանիայի Իլմենաու քաղաքի Տեխնոլոգիական համալսարանում՝ վերապատրաստման դասընթացների շրջանակներում, ինչպես նաև ՀԱՊՀ ՔՀևՑ ամբիոնի գիտական սեմինարներում: Ատենախոսության թեմայով հրատարակված է 7 հոդված:

Աշխատանքի կառուցվածքը և ծավալը

Թեկնածուական ատենախոսությունը բաղկացած է ներածությունից, չորս գլխից, եզրակացությունից, 107 անուն օգտագործված գրականության ցանկից և հավելվածից: Աշխատանքի ընդհանուր ծավալը 142 էջ է:

ԳԼՈՒԽ 1. ՄՈԴՈՒԼՅԱՐ ԹՎԱԲԱՆՈՒԹՅԱՆ ԲԼՈԿՈՒՄ ԺԱՄԱՆԱԿԱԿԻՑ ՆԱԽԱԳԾՄԱՆ ՄԵԹՈԴՆԵՐԻ ՀԵՏԱԶՈՏՈՒՄԸ ԵՎ ՎԵՐԼՈՒԾՈՒԹՅՈՒՆԸ

Մնացորդային դասերի համակարգը, որը մոդուլյար թվաբանության հիմքն է, ոչ դիրքային համակարգ է: Մոդուլյար թվաբանության օգտագործման արդյունավետությունը բացատրվում է մնացորդային դասերի հետևյալ երկու հատկությամբ.

1) գումարման և բազմապատկման գործողություններ իրականացնելիս բացակայում է կարգերի փոխանցումը,
2) մնացորդային դասերի համակարգում վեկտորի մեկ դիրքի սխալը չի ազդում վեկտորի մյուս դիրքերի հաշվարկների վրա, ինչի արդյունքում հնարավորություն է ընձեռվում նախագծել բարձր կայունությամբ թվային սարքեր:

Մոդուլյար թվաբանության բլոկում ժամանակակից նախագծման մեթոդները հետազոտելիս առանձին վերլուծվել են մոդուլյար թվաբանության արդի մեթոդները և մոդուլյար սարքերի նախագծման առկա մեթոդները, թվային սարքերի նախագծման ժամանակակից մեթոդները և կիրառման առանձնահատկությունները, ինչպես նաև հիմնավորվել է այդ սարքերի նախագծման համակարգի ընտրությունը և ձևավորվել աշխատանքի նպատակը և հետազոտության խնդիրները:

1.1. Մոդուլյար թվաբանության արդի մեթոդների վերլուծությունը

Շատ կիրառական խնդիրներում անհրաժեշտություն է առաջանում կատարել ամբողջ թվերով թվաբանական գործողություններ՝ ներկայացված որոշակի մոդուլով: Ներկայացման այս ձևը ենթադրում է, որ ամբողջ թիվը ներկայացված է մոդուլին բաժանումից ստացված մնացորդի միջոցով: a թվի մնացորդը ըստ մոդուլ m-ի, նշանակվում է a(mod m) ձևով: Ենթադրենք՝ տրված են m, a և b դրական ամբողջ թվերը:

$$a \equiv b(mod\ m), \tag{1.1}$$

որտեղ, m-ը կոչվում է համեմատման մոդուլ, a-ն և b-ն համեմատական են m մոդուլով, եթե (a-b) տարբերությունը բաժանվում է m-ի `(m|a-b):

Նկարագրածից երևում է, որ եթե a-ն բաժանվում է m-ի, ապա $a \equiv 0(mod\ m)$: a-ն համեմատական է b-ին մոդուլ m-ով, եթե a և b թվերը m մոդուլի վրա բաժանելիս ունեն միևնույն մնացորդները [8, 9, 13]:

Մոդուլյար թվաբանության հիմնական օրենքները և թեորեմները

Ցանկացած a, a1, b, b1, c Є Z ամբողջ թվերի համար տեղի ունեն հետևյալ առնչությունները.

1) a≡b (mod m) միայն այն դեպքում, եթե a և b թվերը բաժանելով m-ի տալիս են միևնույն մնացորդը:
2) Ռեֆլեքսիվություն. $a \equiv a(mod\ m),\ a < m$:
3) Սիմետրիկություն. եթե $a \equiv b(mod\ m)$, ապա $b \equiv a(mod\ m)$:
4) Տեղափոխություն. Եթե $a \equiv b(mod\ m)$ և $b \equiv c(mod\ m)$, ապա $a \equiv c(mod\ m)$:
5) Եթե $a \equiv a_1(mod\ m)$ և $b \equiv b_1(mod\ m)$, ապա $a + b \equiv a_1 + b_1(mod\ m)$ և $a \cdot b \equiv a_1 \cdot b_1(mod\ m)$:

Էլեկտրոնիկայի, սխեմատեխնիկայի, հաշվողական տեխնիկայի, էլեկտրատեխնիկայի և գիտության այլ ոլորտների շատ խնդիրներ պահանջում են կոմպլեքս թվերով հաշվարկներ: Կոմպլեքս թվերով հաշվարկների ժամանակ կարող է ճշտության կտրուկ կորուստ տեղի ունենալ հետևյալ պատճառով. համակարգիչներում կոմպլեքս թվաբանությունն իրականանում է սահող ստորակետով ավանդական թվաբանության հիման վրա, և թվաբանական գործողությունների իրականացման ժամանակի աճի՝ ճշտությունից խիստ կախվածության հետ կապված խնդիրն առավել բնութագրական է կոմպլեքս թվերով հաշվարկներին [14, 18, 62, 63]: Նշված խնդիրը պահանջում է նոր մեթոդների որոնում, որը կապված է թվերի ներկայացման և մշակման ոչ ավանդական հաշվարկային համակարգերի և թվաբանության կիրառման հետ [15, 16, 62, 85]: Այսպիսի թվաբանությունների շարքին է դասվում մոդուլյար թվաբանությունը:

Ներկա ժամանակներում մոդուլյար թվաբանությունը բազմաթիվ աշխատանքներում օգտագործվել է որպես արագագործության բարձրացման միջոց՝ կիրառվելով ազդանշանների թվային մշակման, գաղտնագրման մեջ և այլ բնագավառներում: Մոդուլյար թվաբանությունը համարվում է ժամանակակից գաղտնագրության հիմքը,

որի առավելությունն այն է, որ նախ ցանկացած արտահայտության համար, որտեղ օգտագործվում են միայն ամբողջ թվեր, պահանջվում է հաշվարկել ոչ թե այդ արտահայտությունը, այլ այն մնացորդը, որը կստացվի այդ արտահայտությունը մեկ այլ ամբողջ թվի բաժանումից, և երկրորդը՝ գաղտնագրման ալգորիթմի կայունությունը ապահովվում է շատ մեծ կարգայնությամբ թվերի (մի քանի հարյուր բիթ) կիրառումով[54, 86]: Մոդուլյար թվաբանությունը մեծ կիրառություն ունի համակարգիչներում ապարատային հսկում իրականացնելու գործընթացում: Գոյություն ունեն ըստ մոդուլի հսկման երկու եղանակ՝ *թվային* և *թվանշանային*: *Թվային հսկման* ժամանակ թվի հսկման կոդը մնացորդն է, որը ստացվում է թիվը մոդուլին բաժանելիս, իսկ *թվանշանային հսկման* դեպքում հսկման կոդը որոշվում է հսկվող թվի թվանշանների նկատմամբ իրականացվող գործողություններով, մասնավորապես, թվանշանների գումարով, և այս դեպքում հսկման կոդը թվի թվանշանների գումարը մոդուլի վրա բաժանելիս ստացված մնացորդն է: Թվանշանային հսկումը հատկապես իրականացվում է երկուական թվերի դեպքում: Ըստ մոդուլի հսկումը հնարավորություն է ընձեռում հսկել նաև թվաբանական գործողությունները: Այդպիսի հսկումը կարելի է իրականացնել՝ հիմնվելով բանաձևեր (1.2) և (1.3.)-ի վրա [3, 10, 11]:

$$\sum_{i=1}^{n} A_i = \sum_{i=1}^{n} r_{ai} \, mod \, m \qquad (1.2)$$

$$\prod_{i=1}^{n} A_i = \prod_{i=1}^{n} r_{ai} \, mod \, m \qquad (1.3)$$

Վերը շարադրվածը 2 օպերանդների դեպքում համարժեք է հետևյալ արտահայտությունների իրականացմանը.

գումարման դեպքում՝ $r_{a+b} = (r_a + r_b) mod \, m,$

հանման դեպքում՝ $r_{a-b} = (r_a - r_b) mod \, m,$

բազմապատկման դեպքում՝ $r_{a \cdot b} = (r_a \cdot r_b) mod \, m,$

բաժանման դեպքում՝ $r_a = (r_b \cdot r_c + r_d) mod \, m,$

որտեղ a – ն բաժանելին է, b – ն՝ բաժանարարը, c – նª քանորդը, d – նª մնացորդը,

r_a, r_b, r_c, r_d–ն՝ համապատասխանաբար a,b,c,d թվերի մնացորդներն են ըստ mod m-ի:

Մնացորդային դասերի համակարգերում հաշվարկման դիրքային համակարգերում ցանկացած բազմակարգ թիվ ներկայացվում է մի քանի ցածր կարգայնությամբ դիրքային թվերի տեսքով, որոնք հանդիսանում են տրված թվի մնացորդները՝ փոխադարձ պարզ հիմքերով թվերի բաժանումից: Սովորական դիրքային երկուական համակարգում գործողությունների իրականացումը (օրինակ՝ երկու թվերի բազմապատկում) կատարվում է հաջորդաբար՝ սկսած ցածր կարգից: Ընդ որում, կատարվում է փոխանցում հաջորդ բարձր կարգ: Մնացորդային դասերի համակարգը հնարավորություն է տալիս զուգահեռացնել այս պրոցեսը: Բոլոր գործողությունները իրականացվում են առանձին և անկախ, հետևաբար՝ հեշտ և արագ:

Արդյունքի որոշման համար կիրառում են մնացորդների մասին չինական թեորեմը, որի վրա հիմնվում է մոդուլյար թվաբանությունը [8, 9, 10] և որի կիրառմամբ կարելի է դիտարկել, օրինակ, $m_1 = 7, m_2 = 11, m_3 = 17$ մոդուլների համար M_i և $b_i (0 < i < 3)$ արժեքների հաշվարկումը:

$M = 17 \cdot 11 \cdot 7 = 1309, \quad m_1 = 11 \cdot 7 = 77, \quad m_2 = 17 \cdot 7 = 119, \quad m_3 = 17 \cdot 11 = 187:$

Որոշենք b_1, b_2, b_3՝ $(77 \cdot b_3) \bmod 17 = 1 \Rightarrow b_3 = 2$, $(17 \cdot 7 \cdot b_2) \bmod 11 = 1 \Rightarrow b_2 = 5$,

$(17 \cdot 11 \cdot b_1) mod\ 7 = 1 \Rightarrow b_1 = 3$, հետևաբար

$$C = \left(c_1(M_1 \cdot b_1) + c_2(M_2 \cdot b_2) + c_3(M_3 \cdot b_3)\right) \bmod M = (10 \cdot 119 \cdot 5 + 6 \cdot 187 \cdot 3) \bmod 1309 = 153:$$

Գումարումը, հանումը և բազմապատկումը հեշտ իրականացվում են, եթե արդյունքները 0-ից m-1 հատվածում են, այլ կերպ ասած, դրանք կարելի է դիտարկել որպես մոդուլ m-ով գործողություններ: Ենթադրենք ունենք u և v ամբողջ թվերը և դրանց մնացորդների բազմությունները՝ $u \leftrightarrow (u_0, u_1, \ldots, u_{N-1})$ և $v \leftrightarrow (v_0, v_1, \ldots, v_{N-1})$, այս դեպքում գումարման, հանման և բազմապատկման գործողությունները որոշվում են հետևյալ կերպ.

$$u + v \leftrightarrow (w_0, w_1, \ldots, w_i), \text{ որտեղ } \quad w_i = (u_i + v_i) \bmod m_i \qquad (1.4)$$

$$u - v \leftrightarrow (w_0, w_1, \ldots, w_i), \text{ որտեղ } \quad w_i = (u_i - v_i) \bmod m_i \qquad (1.5)$$

$$u \cdot v + \leftrightarrow (w_0, w_1, \ldots, w_i), \text{ որտեղ } \quad w_i = (u_i \cdot v_i) \bmod m_i \qquad (1.6)$$

Առանձին դիտարկենք բաժանման գործողությունը, թեև ատենախոսությունում մոդուլյար թվաբանության բլոկում բաժանման գործողություն իրականացնող սարքեր չեն նախագծվել, քանի որ գործնականում այդպիսի սարքեր չեն օգտագործվում: Բաժանման գործողությունը կատարվում է բազմապատկման նման, սակայն միայն այն դեպքում, երբ բաժանման արդյունքը ստացվում է ամբողջ թիվ [8, 10, 35, 71]:

Դիտարկենք օրինակի հիման վրա՝ $A/B = C$: Բաժանման գործողությունը կարելի է փոխարինել բազմապատկման գործողությամբ՝ $A \cdot (1/B) = C$: Մոդուլյար թվաբանության համար

$$(A \cdot (1/B) \bmod m_1 = A \bmod m_1 \cdot (1/B) \bmod m_1: \tag{1.7}$$

$A = 1377$ թիվը բաժանենք $B = 9$ թվին $m_1 = 7, m_2 = 11, m_3 = 17$ մոդուլների համար: Ներկայացնենք A և B թվերի մնացորդները՝ $A(a_1, a_2, a_3), B(b_1, b_2, b_3)$:

$a_1 = A \bmod m_1 = 1377 \bmod 7 = 5,\ a_2 = A \bmod m_2 = 1377 \bmod 11 = 2,$

$a_3 = A \bmod m_3 = 1377 \bmod 17 = 0,$

$b_1 = B \bmod m_1 = 9 \bmod 7 = 2,\ b_2 = B \bmod m_2 = 9 \bmod 11 = 9,$

$b_3 = B \bmod m_3 = 9 \bmod 17 = 0:$

Ստանում ենք A=(5, 2,0):B(2,9,9)=C(c1,c2,c3) արտահայտությունը, որի համար բաժանման գործողությունը փոխարինելով բազմապատկմամբ կարելի է որոշել յուրաքանչյուր c_i՝

$$c_i = (a_i \cdot (1/b_i)) \bmod m_i \tag{1.8}$$

$1/b_i$ դա $\bmod m_i$ դաշտում b_i տարրին հակառակ տարրն է, որը նշանակենք b_i^{-1}: Հետևաբար կարելի է պնդել, որ $(b_i \cdot b_i^{-1}) \bmod m_i = 1:$ Կիրառելով $b_i^{m_i - 1} - 1$ բանաձևը ստանում ենք՝

$$(b_i^{m_i - 1}) \bmod m_i = (b_i \cdot b_i^{-1}) \bmod m_i \tag{1.9}$$

Բանաձև (1.9)-ից ստանում ենք՝

$$(b_i^{m_i - 2}) \bmod m_i = b_i^{-1} \tag{1.10}$$

Տվյալ օրինակի համար ստանանք յուրաքանչյուր մոդուլով բաժանարարի հակառակ տարրը՝ $b_3^{-1} = (9^{17-2}) \bmod 17 = 15,\ b_2^{-1} = (9^{11-2}) \bmod 11 = 5, b_1^{-1} = (9^{7-2}) \bmod 7 = 4:$

Այստեղից՝

$c_3 = (a_1 \cdot b_1^{-1}) \mathrm{mod} m_1 = 0, \quad c_2 = (a_2 \cdot b_2^{-1}) \mathrm{mod} m_2 = (2 \cdot 5) \mathrm{mod} 11 = 10,$

$c_1 = (a_3 \cdot b_3^{-1}) \mathrm{mod} m_3 = (5 \cdot 4) \mathrm{mod} 7 = 6:$

Մոդուլյար համակարգում $m_1 = 7, m_2 = 11, m_3 = 17$ մոդուլների համար բաժանման արդյունքը՝ C=(6,10,0):

Էյլերի ֆունկցիայի օգնությամբ ըստ մոդուլի աստիճան բարձրացնելու գործողության իրականացման բացատրությունը և մաթեմատիկական հիմնավորումը բերված են աշխատություններ [8, 10, 35, 71]-ում, որտեղ ներկայացված է նաև Ֆերմայի թեորեմը պարզ թվերի վերաբերյալ:

Հարկ է նշել, որ p պարզ թվի համար Էյլերի ֆունկցիան՝ φ(p) = p-1: Նկատենք, որ p պարզ թվի համար p-1 արժեքը համընկնում է նրա Էյլերի ֆունկցիայի արժեքի հետ՝ φ(p)=p-1: Այստեղից Ֆերմայի թեորեմը կարող ենք գրել հետևյալ տեսքով $a^{p-1} \equiv a^{\varphi(m)} \equiv 1 \mathrm{mod}\ p$: Վերջին հավասարումը ճիշտ է ցանկացած m մոդուլի համար, որը փոխադարձ պարզ է a թվի հետ, a≥1: Այս փաստը հաստատվում է Էյլերի թեորեմայի օգնությամբ, որն էլ հանդիսանում է Էյլերի ֆունկցիայի ամենակարևոր հատկությունն է:

Ֆերմայի և Էյլերի թեորեմները թույլ են տալիս որոշել տրված թվի բարձր աստիճանի մնացորդը ըստ մոդուլի: Ենթադրենք, անհրաժեշտ է գտնել a_n թվի մնացորդը ըստ մոդուլ m-ի, որտեղ ԱԸԲ(a,m)=1 և φ(m)<n: n թիվը կարելի է ներկայացնել $n = q \cdot \varphi(m) + r$, որտեղ 0≤r<φ(m):

Մոդուլյար հաշվարկների կատարման գուգահեռականությունը

Մոդուլյար տեսքով ներկայացնելու առավելությունն այն է, որ թվաբանական գործողությունները կկատարվեն ավելի քիչ քայլերով, քան սովորական ձևով ներկայացնելիս:

Մնացորդային դասերի համակարգում դիրքային հաշվարկային համակարգում ներկայացված թվի յուրաքանչյուր ai-րդ կարգը իրենից ներկացնում է ամենափոքր դրական մնացորդ է ըստ mi-րդ մոդուլի բաժանումից, նոչ թե նախորդ քանորդի բաժանումից [24, 25, 28, 64, 65]: Աղյ. 1.1-ում ներկայացված են 0-ից 29 տասական թվերի (դիրքային հաշվարկային համակարգ-ԴՀՀ) համարժեքները՝ ներկայացված մնացորդային դասերի համակարգում (ՄԴՀ) 2, 3 և 5 փոխադարձ պարզ թվերի մոդուլներով:

Տարբեր մուդուլներով միաժամանակ հաշվարկումների եղանակները ստեղծում են զուգահեռացման ևս մեկ մակարդակ [10]:

Օրինակ, 26 թիվը աղ. 1.1-ից կարելի է ՄԴՀ-ում {(1,5);(2,3);(0,2)} բազմության միջոցով, որի հիմնական բնութագրիչը նրա տարրերի փոխադարձ անկախությունն է: Հենց սա է հիմնավորում այն զուգահեռացման լրացուցիչ մակարդակը, որն ապահովում է ՄԴՀ-ն, ինչը այլ կերպ անվանում են թվերի կարգերի մակարդակի զուգահեռացում:

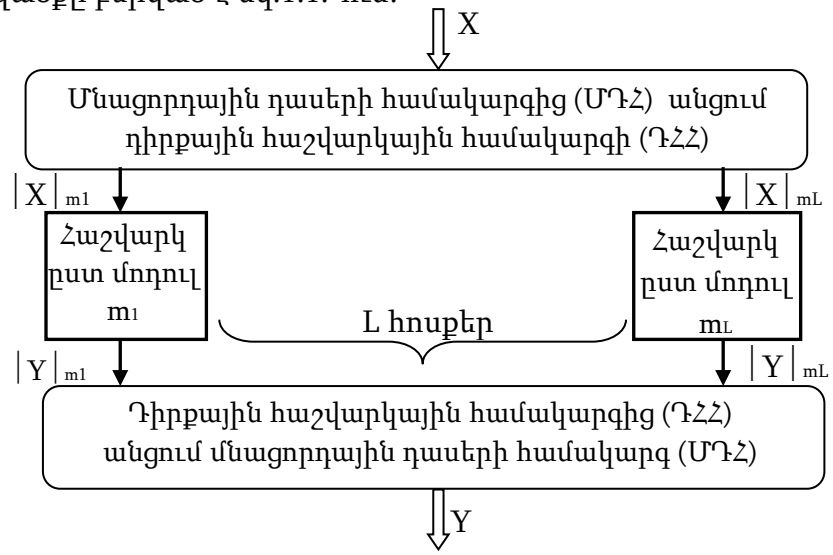
Աղյուսակ 1.1. Թվի ներկայացումը ԴՀՀ-ից ՄԴՀ-ում 2, 3 և 5 մոդուլներով

թիվը ԴՀՀ-ում	թիվը ՄԴՀ-ում		
	5	3	2
0	0	0	0
1	1	1	1
2	2	2	0
3	3	0	1
4	4	1	0
5	0	2	1
6	1	0	0
7	2	1	1
8	3	2	0
9	4	0	1

թիվը ԴՀՀ-ում	թիվը ՄԴՀ-ում		
	5	3	2
10	0	1	0
11	1	2	1
12	2	0	0
13	3	1	1
14	4	2	0
15	0	0	1
16	1	1	0
17	2	2	1
18	3	0	0
19	4	1	1

թիվը ԴՀՀ-ում	թիվը ՄԴՀ-ում		
	5	3	2
20	0	2	0
21	1	0	1
22	2	1	0
23	3	2	1
24	4	0	0
25	0	1	1
26	1	2	0
27	2	0	1
28	3	1	0
29	4	2	1

Գումարման և բազմապատկման գործողությունները ՄԴՀ-ում իրականացվում են զուգահեռ L հոսքերով: Մոդուլյար թվաբանությունում ազդանշանի թվային մշակման սարքերի կառուցվածքը բերված է նկ.1.1.-ում:



Նկ. 1.1. ՄԴՀ-ում ազդանշանի թվային մշակման սարքի կառուցվածքը

Մուտքային X թիվը ԴՀՀ-ից ձևափոխվում է ՄԴՀ-ի {m1,m2,…,mi} մոդուլների բազիսում, որից հետո կատարվում են միմյանցից անկախ հաշվարկները յուրաքանչյուր մոդուլ mi-ի համար: Ելքում իրականացվում է ՄԴՀ-ից ԴՀՀ հետ ձևափոխումը

Նկ. 1.1-ում պատկերված կառուցվածքը ունի մի շարք առավելություններ ինտեգրալային սխեմաների վրա նրա իրականացման դեպքում.

1. Առանձին մոդուլով յուրաքանչյուր կարգի (կանալի) անկախությունը բյուրեղի նախագծման և պլանավորման ժամանակ ապահովում է նկատելի ճկունություն:
2. Փոքր տարրային ռեսուրսներ ունեցող ծրագրավորվող ինտեգրալային սխեմաներով իրացումը հնարավորություն է տալիս տեղակայել սխեման տարբեր բյուրեղների վրա:
3. Ուղղորդող միջնակապերը տարածվում են միայն առանձին հաշվողական կանալի ներսում, ինչն ապահովում է ծախսվող հզորության նվազեցում և հապաղումների փոքրացում:
4. Յուրաքանչյուր առանձին կարգը (կանալը) սինխրոնացվում է անկախ մյուսներից, ինչը հանգեցնում է արագագործության զգալի բարձրացման:

1.2. Թվային սարքերի նախագծման ժամանակակից միջոցների հետազոտումը և կիրառման առանձնահատկությունները

Մշակվող ինտեգրալային սխեմաները պետք է համապատասխանեն պահանջներին (օրինակ կրկնակի օգտագործման հնարավորությունը, բարձր արագագործություն և այլն): Արագագործության աճը պետք է ուղղորդվի ինտեգրալ սխեմաների փոքր չափսերով և ցածր էներգասպառմամբ, սակայն այս մեծությունները սովորաբար հակառակ համեմատական են:

Մինչև վերջին ժամանակները տվյալ հակասությունների լուծման եղանակներից էր հատուկ մասնագիտացված պատվիրված ինտեգրալ սխեմաները [45, 67]: Դրանք հիմնականում կիրառվում էին այն սարքերում, որոնք ունեին մեծաքանակ

արտադրություն և, ինչը ավելի կարևոր էր, բացթողնված սխալի ցանկացած ուղղումը հնարավոր էր, նորից նախագծման նույն փուլերով անցնելու դեպքում:

Այս խնդիրների լուծումն էր հանդիսանում նոր մեթոդաբանության կիրառումը՝ բյուրեղի վրա համակարգերի նախագծումը, կամ System-on-Chip (SoC) [17, 18, 42, 43, 55, 59]: Այս մեթոդաբանությունը ծագեց հետևյալ գործոնների ազդեցությամբ.

- Կիսահաղորդչային տեխնոլոգիայի կատարելագործումը հնարավորություն էր ընձեռել մեկ բյուրեղի վրա տեղադրել տասնյակ միլիոնավոր տրանզիստորներ: Հնարավոր դարձավ մեկ բյուրեղի վրա միավորել թվային և անալոգային բլոկներ [45, 61, 82]:
- Մշակվող սարքերի անընդհատ բարդացման պայմաններում նախագծման ժամանակի կրճատման անհրաժեշտությամբ:

Այս և այլ տարբեր փաստերը ստիպեցին ավելի լայն կիրառել SoC նախագծման մեթոդաբանությունը, որի հիման վրա շատ նախագծող ընկերությունները կենտրոնացան մասնագիտացված ԲՖ բլոկների մշակման վրա: Մասնագիտացված ԲՖ բլոկների ստեղծումը իրականացվում էր համաձայն տվյալ ընկերության ներքին ստանդարտներին (հաճախ փակ, գաղտնի) և առաջանում էին համատեղության խնդիրներ: Կարևոր հարցերից էր նաև ԲՖ-բլոկների կրկնակի օգտագործման սկզբունքը, որտեղ կարելի է առանձնացնել SoC-ի առանձին բաղադրիչների կամ ամբողջ SoC-ի բազմակի օգտագործումը [68, 75]: Կարելի է առանձնացնել ԲՖ-բլոկների հետևյալ առավելությունները.

- SoC-ի հիման վրա կառուցված սարքերը որպես կանոն ունեն առավել փոքր մակերես,
- Մեկ բյուրեղի վրա ԲՖ բլոկների միավորումը թույլ է տալիս էականորեն կրճատել միջբլոկային կապերի քանակը և այդպիսով բարձրացնել համակարգի հուսալիությունը:

Հիմնվելով թվային սարքերի նախագծման ընդհանուր մեթոդաբանության վրա [83], ատենախոսական աշխատանքում օգտագործվում է սարքերի սինթեզման վրա հիմնված նախագծման մեթոդը, որն իրագործված է թվային սարքերի նկարագրման բարձր մակարդակի լեզուներից մեկով՝ Verilog-ով, այնուհետև ավտոմատ նախագծման

համակարգերի միջոցով (XST ISE սինթեզող փաթեթ, Xillinx), արտապատկերվում են տրամաբանական տարրերի հասանելի բազիսի վրա: Այս մեթոդով իրականացված նախագծերի առավել կարևոր բնութագրերից է դրանց բացարձակ անկախությունը տրամաբանական տարրերի բազիսից և նախագծման տեխնոլոգիայից: Իսկ թերությունների շարքին կարելի է դասել այն փաստը, որ մշակվող թվային սարքերի քանակի մեծացմանը զուգընթաց բարդանում և աճում է նաև տվյալ սարքերի նկարագրման ծավալը, ինչը հետագայում հանգեցնում է այդ սարքերի մշակման և թեստավորման համար նախատեսվող ժամանակային մեծ ծախսերի:

Կարևոր է նշել, որ մեծ համակարգերի նախագծման իրական գործընթացում օգտագործվում են, որպես կանոն, նախագծման տարբեր մեթոդաբանություններ, օրինակ՝ նախագծվող սարքերի առանձին բլոկներ կարող են նկարագրվել Verilog կամ VHDL լեզվով, իսկ մնացած բլոկների փոխարեն ընտրվել այլ համակարգերի՝ նախօրոք մշակված համապատասխան տարրեր [1, 21]:

Տվյալ աշխատանքում մոդուլյար թվաբանության բլոկում՝ որպես մասնագիտացված թվային սարքերի նախագծման տարբերակ ընտրված է վերը նշված մեթոդաբանությունն, երբ առանձին բլոկներ նկարագրված են Verilog նախագծման լեզվի միջոցով, և առանձին բլոկների աշխատանքը ծրագրավորված է FPGA (Field-Programmable Gate Array) ծրագրավորվող տրամաբանական սարքերի միջոցով:

Մնացորդների փոքր կարգայնությունն ապահովում է աղյուսակային թվաբանության իրականացում, որի դեպքում գործողության արդյունքը չի հաշվարկվում ամեն անգամ, այլ մեկ անգամ հաշվելուց հետո տեղադրվում է հիշող սարքում և անհրաժեշտության դեպքում վերցվում է հիշող սարքից, ասինքն՝ աղյուսակային թվաբանության դեպքում մնացորդային դասերի համակարգում գործողությունը և կոնվեյերացումը կատարվում են սինքրոնացման հաճախականության մեկ պարբերության ընթացքում (մեքենայական տակտ): Խնդիրներ առաջանում են թվերի ներկայացման միջավայրի գերլրացման և արդյունքների կլորացման դեպքում: Մնացորդային դասերի համակարգում աղյուսակային մեթոդով կարելի է իրականացնել ոչ միայն պարզագույն գործողություններ, այլև բարդ ֆունկցիաներ՝ նույնպես մեկ մեքենայական տակտով [22, 23]:

Թվային սարքի կարգայնությունը ոչ թե ալգորիթմի մուտքային տվյալների քանակությունն է, այլ բիթերի այն քանակությունը, որն անհրաժեշտ է թվի գրանցման համար: Ալգորիթմը, որի մուտքային տվյալներն են՝ $a_1, a_2, \dots, a_k$ կոչվում է պոլինոմիալ, եթե նրա աշխատանքի ժամանակը սահմանափակվում է $log_2 a_1, log_2 a_2, \dots, log_2 a_k$ բազմանդամով կամ մուտքային տվյալների բազմանդամով:

Ծրագրավորվող տրամաբանական սարքեր՝ PLD (Programmable Logic Devices)

Ծրագրավորվող տրամաբանության ժամանակակից ինտեգրալ սխեմաները բնութագրվում են ցածր արժեքով, բարձր արագագործությամբ, բազմակի վերածրագրավորման մեծ ֆունկցիոնալ հնարավորություններով և փոքր օգտագործվող հզորությամբ: Նախագծման ժամանակը այս սարքերի հիման վրա, նույնիսկ բարդ սխեմաների համար, զգալիորեն նվազում է՝ համեմատած նախագծման ավանդական մեթոդների հետ: Այսպիսի ծրագրավորվող տրամաբանական սարքերը (PLD-ները) լայն կիրառում և տարածում ստացան որպես թվային սարքերի համապիտանի տարրային բազա: Արտադրման տեխնոլոգիաների հետագա զարգացումը հնարավորություն ստեղծեց 1 բյուրեղի վրա, ծրագրավորվող կապերով միացված, բազմաթիվ PLD-ների իրականացման համար, որոնք կոչվեցին բարդ PLD-ներ (Complex PLD - CPLD) [2, 12, 45, 84]:

FPGA՝ կիրառողի կողմից ծրագրավորվող տարրերի մատրիցը տողերով և սյուներով տեղակայված նույն ձևով կոնֆիգուրացվող տրամաբանական բլոկներ են: FPGA-ի ներքին հատվածում իրականացվում է նախագծվող սարքի ֆունկցիոնալ սխեման, իսկ բյուրեղի եզրերում տեղակայված են մուտքի/ելքի բլոկներ, որոնք ապահովում են FPGA ինտերֆեյսը այլ միկրոսխեմաների հետ. ընդ որում, ժամանակակից FPGA-ում մուտքի/ելքի սխեմաները կարող են կատարել տվյալների փոխանցման ազդանշանների բազմաթիվ պահանջներ, որոնց քանակը հասնում է 20-ի:

Ըստ ճարտարապետության, FPGA միկրոսխեմաները լինում են դասական և համակցված: Համակցված FPGA-երը պարունակում են CPLD-երի տարրեր:
FPGA միկրոսխեմաները նախագծվել են Xilinx ընկերության կողմից, որի արտադրանքները PLD-ների համաշխարհային շուկայում մեծ տեղ են զբաղեցնում:

Կա FPGA-ների մի քանի տեսակ.

- Տրիգերային հիշողության վրա ծրագրավորված (Xilinx ընկերություն),
- Մեկանգամյա ծրագրավորվող FPGA՝ antifuse տիպի միջկապերով (Actel, QuickLogic ֆիրմաներ),
- Ֆլեշ հիշողության կոնֆիգուրացումով (Actel, ProASIC),
- Համակցված ճարտարապետությամբ, որոնց զարգացումն սկսվել է Altera ընկերություն FLEX8000 ընտանիքից:

FPGA-ները բաղկացած են հետևյալ հիմնական մասերից.

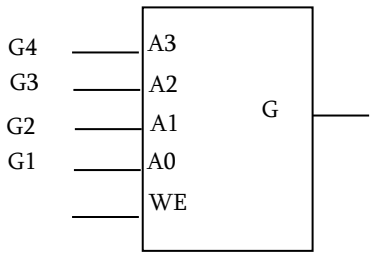
- ֆունկցիոնալ բլոկ,
- ծրագրավորվող կապերի համակարգ,
- մուտքի/ելքի բլոկներ:

FPGA-ների ֆունկցիոնալ բլոկի տիպիկ օրինակներ են համարվում.

- մուլտիպլեքսորների վրա հիմնված տրամաբանական մոդուլները,
- ծրագրավորվող հիշողությունների վրա հիմնված տրամաբանական բլոկները (LUT Look-up Table):

Դասական FPGA-ի օրինակ է համարվում Xilinx ընկերության FPGA XC4000 արտադրանքը: Ընդլայնված ընտանիքներում ամենամեծ ինտեգրալ սխեման պարունակում է 3136 CLB-ներ [2, 77, 81]:

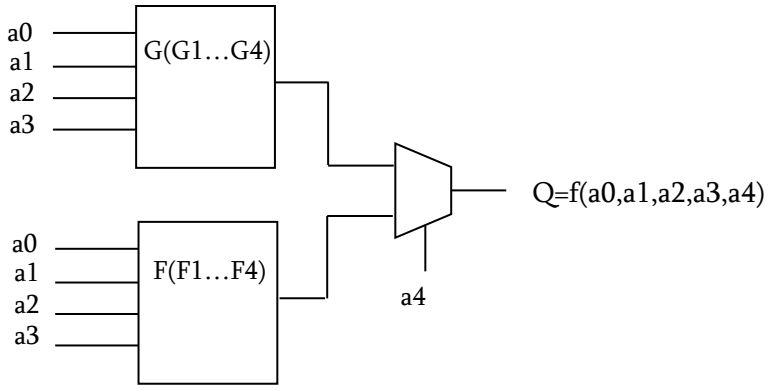
Հետագայում FPGA-ի ճարտարապետությունը բարդացավ: Նրա կազմում հայտնվում են հիշողության ներկառուցված բլոկներ և բազմապատկիչներ: FPGA-ների և CPLD-ների հիմնական տրամաբանական տարրի դերը կատարում է տրամաբանական LUT (look-up Table) աղյուսակը, որը միաբիթ օպերատիվ հիշող սարք է (ОЗУ, SRAM) 16 բջիջների համար (նկ.1.2.):



Նկ. 1.2. 16 x1 LUT-ի նշանակումը

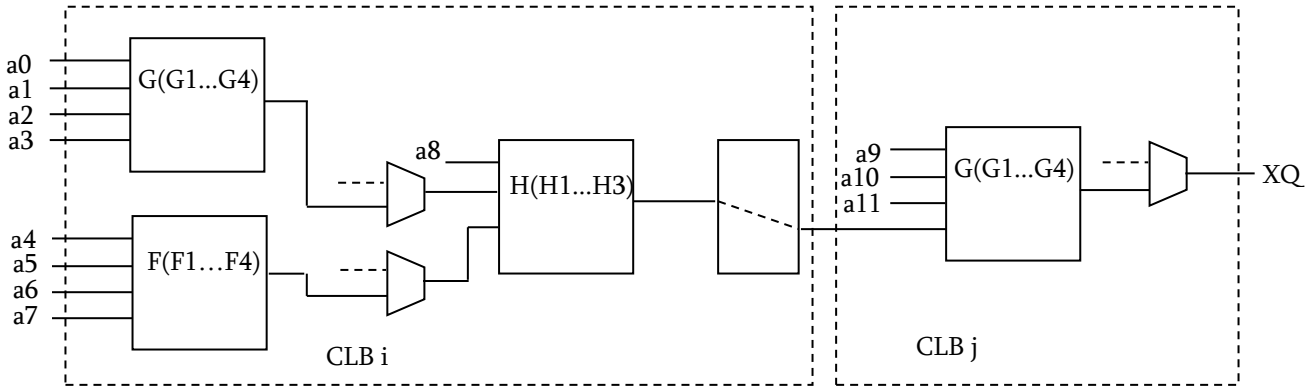
LUT-ում A3, A2, A1, A0 հասցեով գրանցվում է “1”, եթե ֆունկցիայի տվյալ հավաքածուում գրանցված է “1”: Օրինակ, եթե 1111 հասցեով գրանցված է “1”, իսկ մյուս հասցեներով ՙ0՚-ներ, ապա LUT-ը իրականացնում է քառամուտք “&” ֆունկցիա: Կոնֆիգուրացվող տրամաբանական բլոկի (CLB) զբաղեցրած մակերեսի փոքրացման և արագագործության բարձրացման նպատակով LUT-բլոկը սովորաբար կառուցվում է երկմուտք մուլտիպլեքսորների հիման վրա:

Հինգ փոփոխականից ֆունկցիայի իրականացումը 2 հատ քառամուտք LUT-երի միջոցով (տես՝նկ. 1.3.):



Նկ.1.3. Հինգ փոփոխականի ֆունկցիայի իրականացում

Տասներկու փոփոխականից ֆունկցիայի իրականացումը LUT-երի միջոցով կարելի է իրագործել 2 CLB-ի միջոցով (CLBi և CLBj): CLBi-ն իրականացնում է ֆունկցիան առաջին 8 փոփոխականից, իսկ CLBj-ն հաջորդ 4 փոփոխականից: CLBi-ի ելքը միացվում է CLBj-ի մուտքի հետ փոխանջատող մատրիցի (միջբլոկային կապ) միջոցով (տես՝նկ.1.4.):



Նկ. 1.4. Տասներկու փոփոխականից ֆունկցիայի իրականացում

Ուսումնասիրելով նախագծման ժամանակակից միջոցները՝ կարելի է առաջարկել մոդուլյար թվաբանության բլոկում թվային սարքրեի նախագծումը կատարել մասնագիտացված ԲՖ-բլոկերի միջոցով, ինչը կրճատում է նախագծման ժամանակը, այդ բլոկերի կրկնակի օգտագործման հնարավորություն է ընձեռնում և ամենակարևորը՝ այս ամենը զուգորդվում է համեմատական ցածր գնով և սարքերի փոքր հզորությամբ (LowPower):

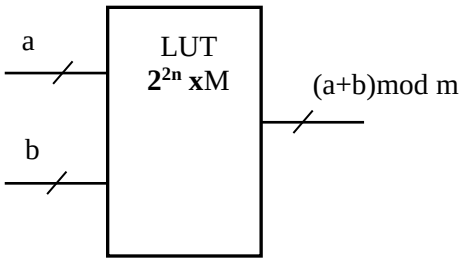
1.3. Մոդուլյար սարքերի նախագծման առկա մեթոդների վերլուծությունը

Նշված նախագծման ժամանակակից միջոցների առկայության պայմաններում մոդուլյար թվաբանության բլոկում այլ հեղինակների կողմից նույնպես իրականացվել են թվային սարքերի մշակումներ, որոնց մի մասը կատարվել է ծրագրավորվող տրամաբանական սարքերի օգտագործմամբ [16, 17, 18]:

Մոդուլյար գումարիչների նախագծման առկա մեթոդները

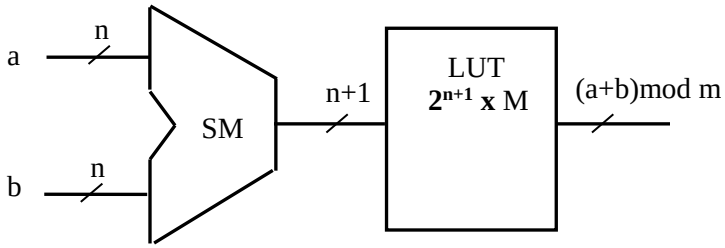
Մոդուլյար գումարիչների առկա մշակումներից կարելի է առանձնացնել *երկու մեթոդ*:

Մեթոդ 1. Նկ. 1.6.-ում ներկայացված մոդուլյար գումարիչի սխեման կատարում է մոդուլյար գումարում (a+b)mod m, մեկ հատ մեծ 2^{2n}xM չափով LUT աղյուսակի միջոցով, որտեղ M=$\log_2$m, իսկ n-ը a և b թվերի կարգայնությունն է: Այս լուծումը հարմար է ոչ մեծ կարգայնությամբ երկու թվերի համար: Երկու n կարգանի a և b թվերը տրվում են LUT աղյուսակի հասցեական մուտքերին, որից ընտրվում է ճիշտ պատասխանը: Լուծման այս տարբերակը նախընտրելի է, երբ a և b թվերի կարգայնությունը չորսից մեծ չէ (n≤4):



Նկ. 1.6. Մեծ LUT աղյուսակի միջոցով իրականացված մոդուլյար գումարիչի սխեման

Մեթոդ 2. Ավելի մեծ կարգայնությամբ թվերի համար LUT աղյուսակը կլիներ զգալի չափերի, և այս դեպքում առաջարկվում է մեկ այլ սխեմա՝ նախապես գումարում և գումարի արժեքի մուտքագրումը LUT աղյուսակ, ինչը կփոքրացնի LUT աղյուսակի չափերը մինչև 2^{n+1} x M: Առաջարկված տարբերակով մշակված սխեման բերված է նկ. 1.7.-ում:



Նկ. 1.7. Նախապես գումարման և LUT-ի միջոցով իրականացված մոդուլյար գումարիչի սխեման

Վերոհիշյալ առաջարկը հիմնված է a և b թվերի նախապես գումարման, և, (a+b)mod m արտահայտության բոլոր հնարավոր արժեքները պարունակող, մեկ հատ 2^{n+1}xM չափի LUT-աղյուսակի օգտագործման վրա: Աղյուսակի չափի կրճատումը 2^{2n}xM-ից մինչև 2^{n+1}xM չափը հնարավորություն կընձեռի ընդլայնել մոդուլների հավաքածուն [16, 17, 20, 31, 57]:

Մոդուլյար բազմապատկիչների նախագծման առկա մեթոդները

Մոդուլյար բազմապատկիչների առկա մշակումներից կարելի է առանձնացնել երեք մեթոդ:

Մեթոդ 1. Ինդեքսային (դիսկրետ-լոգարիթմական) մեթոդով մոդուլյար բազմապատկիչներ: Մոդուլյար բազմապատկիչների կառուցման տարածված մեթոդներից է ինդեքսային բազմապատկման մեթոդը (դիսկրետ-լոգարիթմական մեթոդ), երբ a և b բազմապատկիչները ձևափոխվում են համապատասխան i ինդեքսների, և մոդուլյար բազմապատկման գործողությունը փոխարինվում է մոդուլյար գումարման գործողությամբ [27, 32]: Թվի ինդեքսային ներկայացումը կապված է Գալուայի դաշտի մաթեմատիկական բնութագրերի հետ, հիմնված է m պարզ թվով մոդուլի պարզունակ արմատի գաղափարի վրա [30, 36, 39, 49]:

Եթե w-ն m մոդուլի պարզունակ արմատն է, ապա $w^0, w^1, w^2, \dots w^{\varphi(m)-1}$, շարքը m-ից փոքր և m-ի հետ փոխադարձ պարզ բոլոր թվերի համախումբ է: w պարզունակ արմատը

ամբողջ թիվ է, որի 0, 1, 2,…, (m-2) աստիճան բարձրացումով ձևավորվում են բոլոր չկրկնվող m մոդուլով մնացորդները: Ըստ Գաուսի, ցանկացած պարզ թվի համար գոյություն ունի պարզունակ արմատ [8, 9]: Այս դեպքում (0: m-1) հատվածից ցանկացած q ամբողջ թվին համապատասխանում է այնպիսի i թիվ, որ՝

$$q = w^i \bmod m \qquad (1.11)$$

Օրինակ, որոշենք m մոդուլով 2,3,... φ(m) թվերի պարզունակ արմատները:

$w^i \bmod 11$ $\varphi(11) = 10$ 4 պարզունակ արմատ

Աղյուսակ 1.2. 11 թվի w պարզունակ արմատների ներկայացումը

w\i	1	2	3	4	5	6	7	8	9	10	
1	1	1	1	1	1	1	1	1	1	1	
2	2	4	8	5	10	9	7	3	6	1	2-ը պարզունակ արմատ է
3	3	9	5	4	1	3	9	5	4	1	
4	4	5	9	3	1	4	5	9	3	1	
5	5	3	4	9	1	5	3	4	9	1	
6	6	3	7	9	10	5	8	4	2	1	6-ը պարզունակ արմատ է
7	7	5	2	3	10	4	6	9	8	1	7-ը պարզունակ արմատ է
8	8	9	6	4	10	3	2	5	7	1	8-ը պարզունակ արմատ է
9	9	4	3	5	1	9	4	3	5	1	
10	10	1	10	1	10	1	10	1	10	1	

Կարելի է պնդել, որ եթե p-ն պարզ կենտ թիվ է այսինքնէ պարզ թիվ է, որը հավասար չէ 2-ի, ապա բոլոր p^k և $2p^k$, որտեղ k=1,2,3,.., տեսքի մոդուլների համար, գոյություն ունեն պարզունակ արմատներ: Եթե վերադառնանք պարզունակ արմատի գաղափարի կիրառմանը, ապա կստանանք հետևյալ համապատասխանությունը՝

$$(a \cdot b) \bmod m = w^{((a+b) \bmod m(m-1)}, \qquad (1.15)$$

որտեղ ia-ն (index_a) և ib-ն (index_b) a և b թվերի m մոդուլով պարզունակ արմատի ինդեքսային արժեքներն են:

a և b թվերի ia (index_a) և ib (index_b) աստիճանները նախապես կարող են հաշվարկվել և տեղադրվել LUT աղյուսակներում, իսկ գումարման գործողությունը կատարել m-1 մոդուլով գումարիչի միջոցով: Հակադարձ ձևափոխումը նույնպես իրականացվում է LUT ի միջոցով [2, 53, 66, 84]:

Փաստորեն a և b թվերի բազմապատկումը կարելի է կատարել՝ հաշվարկելով ia (index_a) և ib (index_b) համապատասխան աստիճանների մոդուլյար գումարումը, այնուհետև կատարել հակադարձ ձևափոխում ինդեքսային ներկայացումից սկզբնական տեսքի: Հակադարձ ձևափոխումը նույնպես իրականացվում է LUT ի միջոցով:

Հետևաբար, եթե թվերի ինդեքսային ձևափոխման LUT1 և LUT2 աղյուսակները ներկայացնենք թվերի աճման կարգով, ապա mod11-ի համար կունենանք աղ.1.3-ում, և հակադարձ ձևափոխման LUT-ը կլինի աղ.1.4-ում ներկայացված տեսքի:

Աղյուսակ 1.3. mod11 թվի w=2 պարզունակ արմատի i ինդեքսային արժեքների ներկայացումը

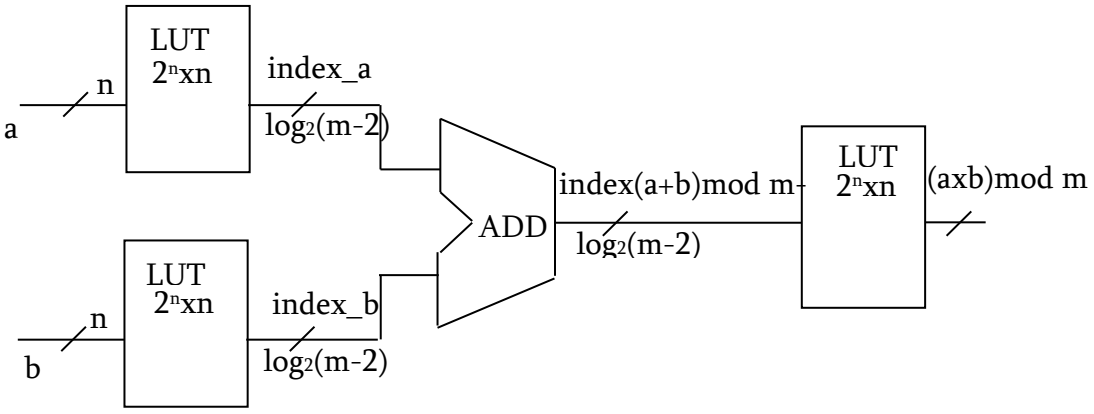
q	1	2	3	4	5	6	7	8	9	10
i	0	1	8	2	4	9	7	3	6	5

Աղյուսակ 1.4. mod11 թվի w=2 պարզունակ արմատի ինդեքսներին համապատասխանող q թվի ներկայացումը

i	0	1	2	3	4	5	6	7	8	9
q	1	2	4	8	5	10	9	7	3	6

Ինդեքսային մեթոդով մոդուլյար բազմապատկման սարքի կառուցվածքային սխեման, LUT աղյուսակների կիրառմամբ, ներկայացված է նկ. 1.8.-ում:

Բազմապատկումը կարելի է առավել արագացնել, m-1 մոդուլի արժեքը ներկայացնել փոխադարձ պարզ արտադրիչների միջոցով՝ հիմնվելով Էյլերի Φ(m) ֆունկցիայի առանձնահատկությունների վրա:



Նկ. 1.8. LUT աղյուսակների կիրառմամբ ինդեքսային մեթոդով մոդուլյար բազմապատկման սարքի կառուցվածքային սխեման

Շարունակելով ուսումնասիրել օրինակը, կարելի է ասել, որ եթե m=7 և m-1=6, իսկ այն ներկայացվում է փոխադարձ պարզ արտադրիչների միակ տեսքով՝ m-1=m1xm2, այսինքն՝ 6=2x3, ապա աղ.1.3-ը և, հետևաբար, LUT1-ի և LUT2-ի պարունակությունները կձևափոխվեն աղ.1.5-ում, իսկ հակադարձ ձևափոխումը, ինչպես նաև LUT3-ի պարունակությունը՝ աղ.1.6-ում ներկայացված տեսքի:

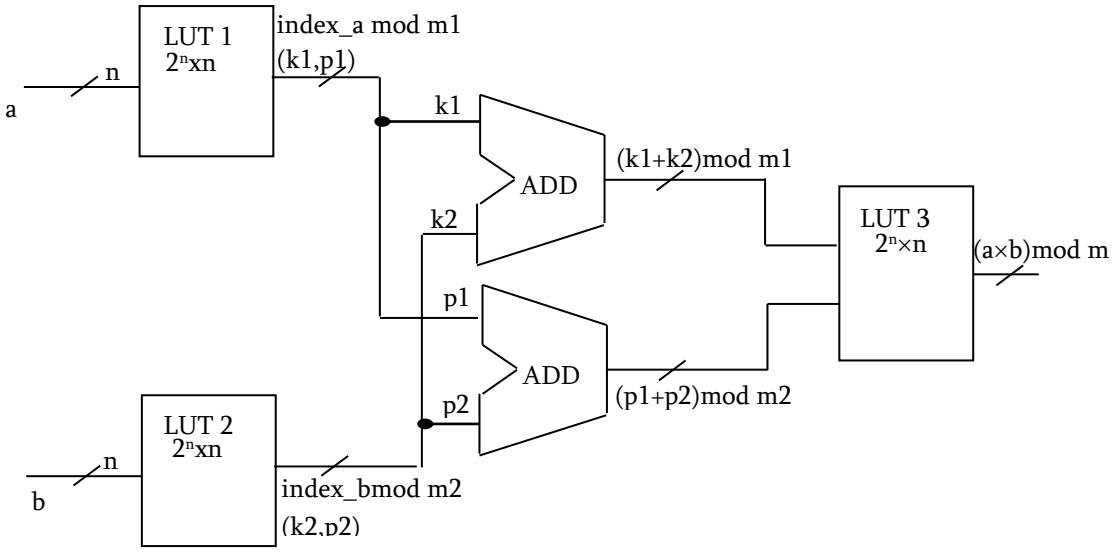
Աղյուսակ 1.5. mod 7 թվի mod2 և mod3 փոխադարձ պարզ արտադրիչների համար i ինդեքսային արժեքների ներկայացումը

q	1	2	3	4	5	6
i	0	2	1	4	5	3
imod2, imod3	(0,0)	(0,2)	(1,1)	(0,1)	(1,2)	(1,0)

Աղյուսակ 1.6. mod 7 թվի mod2 և mod3 փոխադարձ պարզ արտադրիչների ինդեքսներին համապատասխանող q թվի ներկայացումը

imod2, imod5	(0,0)	(0,1)	(0,2)	(1,0)	(1,1)	(1,2)
q	1	4	2	6	3	5

Զուգահեռ ինդեքսային մոդուլյար բազմապատկման գործողության կատարման դեպքում, այսինքն, երբ մոդուլի արժեքը կարելի է ներկայացնել փոխադարձ պարզ արտադրիչների միջոցով, ինդեքսները, որով կատարվելու է գումարման գործողությունը, ունեն ավելի փոքր կարգայնություն, համեմատած սովորական ինդեքսային մոդուլյար բազմապատկման ինդեքսների հետ (նկ. 1.9)[32, 49, 50]:



Նկ. 1.9. LUT աղյուսակների կիրառմամբ զուգահեռ ինդեքսային մոդուլյար բազմապատկման սարքի կառուցվածքային սխեման

Հատուկ անհրաժեշտ է նշել, որ ամբողջ թիվը կարելի է ներկայացնել փոխադարձ պարզ արտադրիչների միջոցով, միայն այն դեպքում, երբ այն չի պատկանում 2^n թվերի բազմությանը, որտեղ n=2, 3, ..., հետևաբար փոխադարձ պարզ արտադրիչների միջոցով թիվը ներկայացնելիս պետք է կատարվի $m-1 \neq 2^n$ պայմանը, որտեղ m-ը մոդուլի արժեքն է:

Մեթոդ 2. Քառակուսիների օրենքի վրա հիմնված մոդուլյար բազմապատկիչներ

Քառակուսիների օրենքի հիման վրա նախագծված բազմապատկիչները հաշվարկում են $|a \cdot b|_m$ մոդուլյար բազմապատկումը համաձայն հետևյալ արտահայտության.

$$ab = \left(\frac{a+b}{2}\right)^2 - \left(\frac{a-b}{2}\right)^2: \quad (1.12)$$

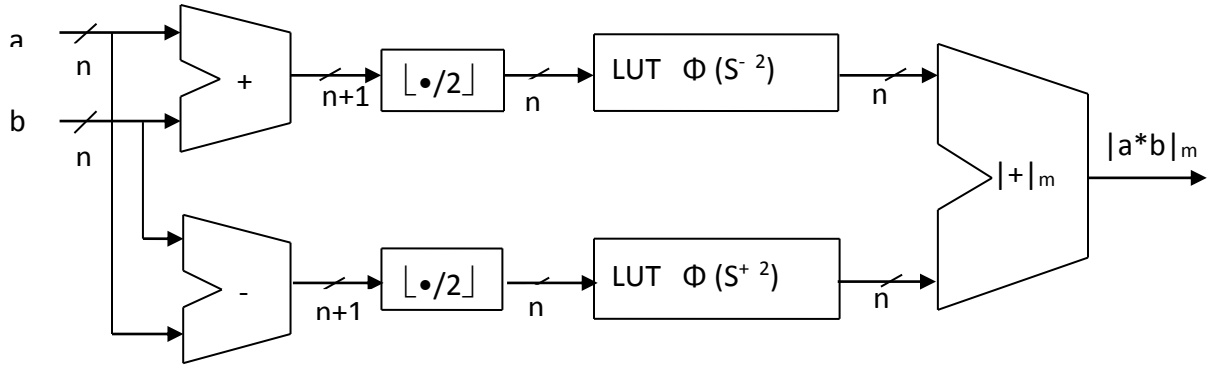
Բանաձև (1.12)- ի հիման վրա մոդուլյար բազմապատկումը կարելի է ձևափոխել.

$$\begin{aligned} |ab|_m &= |\Phi(S^+) - \Phi(S^-)|_m, \\ \Phi(S) &= |S^2|_m, \\ S^+ &= \frac{a+b}{2} \qquad S^- = \frac{a-b}{2} \end{aligned} \quad (1.13)$$

Արտադրյալը $|ab|_m$ կարելի է հաշվել հետևյալ բանաձևով՝

$$|ab|_m = \left| \left|\frac{1}{4}\right|_m |(a+b)^2 - (a-b)^2|_m \right|_m \quad (1.14)$$

Հիմնվելով բանաձև (1.14)-ի և LUT աղյուսակների կիրառման սկզբունքների վրա՝ առաջարկվում է քառակուսիների օրենքի վրա հիմնված մոդուլյար բազմապատկիչների հետևյալ կառուցվածքային սխեման (նկ. 1.10.) [32, 49]:



Նկ. 10. Բաժանող սարքերի կիրառմամբ մոդուլյար բազմապատկման սխեման ըստ քառակուսիների օրենքի

Մեթոդ 3. Բուտի ալգորիթմի կիրառմամբ մոդուլյար բազմապատկիչներ

Մեծ կարգայնություն ունեցող թվերի մոդուլյար բազմապատկումն իրականացնելու համար առաջարկվել է Բուտի ալգորիթմի կիրառումը:

Բարձր արտադրողականությամբ սարքերի մշակման ժամանակ ապարատային մեթոդներով բազմապատկման գործողության իրականացումը որոշակի բարդությունների հետ է կապված: Փոքր կարգայնությամբ թվերի բազմապատկման համար նպատակահարմար է կիրառել դիրքային մատրիցային /կոմբինացիոն/ բազմապատկիչ: Երկու՝ A և B օպերանդների բազմապատկման իրականացումը տեղաշարժի և գումարման գործողությունների օգնությամբ շատ ժամանակ է պահանջվում: Բուտի վերակոդավորման գործակիցները 2^k-1 մոդուլով բազմապատկումն իրականացնելու համար կիրառվել է աշխատանքներ [19, 26]-ում ներկայացված գործակիցների հաշվարկման լեմման, որի վրա հիմնվելով՝ Բուտի մոդիֆիկացված ալգորիթմով 2^k-1 մոդուլով բազմապատկման մասնակի արտադրյալների գործակիցները որոշվում են ըստ աղ. 1.7-ի:

Աղյուսակ 1.7. Բուտի մոդիֆիկացված ալգորիթմով 2^k-1 մոդուլով բազմապատկուման մասնակի արտադրյալների գործակիցները որոշվում

b_{2i+1}	b_{2i}	b_{2i-1}	PPi մասնակի արտադրյալներ	
0	0	0	0	0 . . . 00 . . . 0
0	0	1	+A, $\lvert A2^{2i}\rvert_{2k-1}$	$a_{n-2i-1}\ a_{n-2i-2}\ a_0\ a_{n-1}\ a_{n-2} \ldots a_{n-2i}$
0	1	0	+A, $\lvert A2^{2i}\rvert_{2k-1}$	$a_{n-2i-1}\ a_{n-2i-2}\ a_0\ a_{n-1}\ a_{n-2} \ldots a_{n-2i}$
0	1	1	+2A, $\lvert A2^{2i+1}\rvert_{2k-1}$	$a_{n-2i-2}\ a_{n-2i-3}\ a_0\ a_{n-1}\ a_{n-2} \ldots a_{n-2i-1}$
1	0	0	-2A, $\lvert -A2^{2i+1}\rvert_{2k-1}$	$\overline{a_{n-2i-2}}\ \overline{a_{n-2i-3}}\ \overline{a_0}\ \overline{a_{n-1}}\ \overline{a_{n-2}} \ldots \overline{a_{n-2i-1}}$
1	0	1	-A, $\lvert A2^{2i}\rvert_{2k-1}$	$\overline{a_{n-2i-1}}\ \overline{a_{n-2i-2}}\ \overline{a_0}\ \overline{a_{n-1}}\ \overline{a_{n-2}} \ldots \overline{a_{n-2i}}$
1	1	0	-A, $\lvert A2^{2i}\rvert_{2k-1}$	$\overline{a_{n-2i-1}}\ \overline{a_{n-2i-2}}\ \overline{a_0}\ \overline{a_{n-1}}\ \overline{a_{n-2}} \ldots \overline{a_{n-2i}}$
1	1	1	0	0 . . . 00 . . . 0

Այսպիսով մասնակի արտադրյալների ձևավորումն իրականացվում է բիթային ժխտումներով և ցիկլիկ տեղաշարժով, ինչի հետևանքով մասնակի արտադրյալների կարգայնությունը հավասար է բազմապատկիչների կարգայնությանը: Եթե Բուտի ալգորիթմի կիրառմամբ փոքր կարգայնության թվերի մոդուլյար բազմապատկման համար, մասնակի արտադրյալների գումարները ձևավորելիս նախընտրելի է մատրիցային բազմապատկիչների կիրառումը, ապա մեծ կարգայնության թվերի մոդուլյար բազմապատկման համար նախընտրելի է Ուոլլեսի ծառանման բազմապատկիչների օգտագործումը [44, 74]: Առավելությունը զուտ գումարման

գործողության մակարդակային կազմակերպումն է, իսկ թերությունը՝ սարքի բարդության կախվածությունը բազմապատկիչների կարգայնությունից:

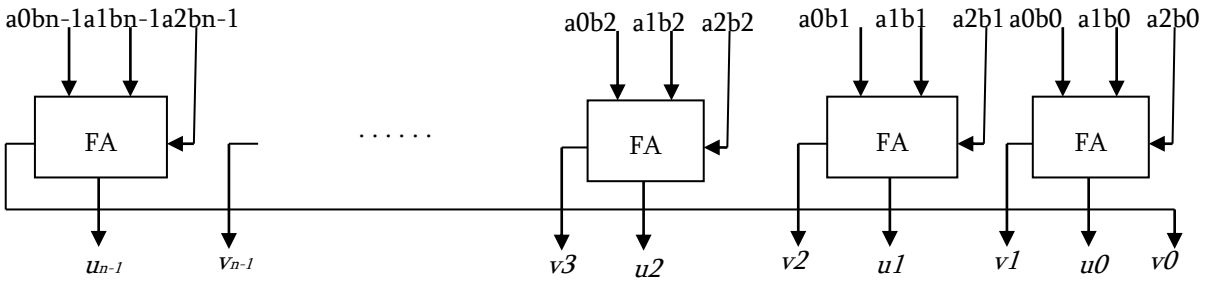
Քանի որ 2^k+1 մոդուլով բազմապատկման սարքում կիրառվում է 2^k-1 մոդուլով մոդուլյար հանող բլոկ, այստեղ անհրաժեշտ է կատարել լրացուցիչ ճշգրտում: Գործողությունները կատարվում են լրացուցիչ կոդում և լրացուցիչ կոնստանտը հավասար է մեկերի, և ոչ թե զրոների [80]: Բուտի մոդիֆիկացված ալգորիթմի վերակոդավորումը և գործակիցների արժեքները ներկայացված են աղ. 1.8-ում:

Աղյուսակ 1.8. Բուտի մոդիֆիկացված ալգորիթմով 2^k+1 մոդուլով բազմապատկման մասնակի արտադրյալների գործակիցների որոշում

b_{2i+1}	b_{2i}	b_{2i-1}	PPi մասնակի արտադրյալներ	
0	0	0	0	0 . . . 01 . . . 1
0	0	1	+A, $\lvert A2^{2i}\rvert_{2^k+1}$	$a_{n-2i-1}\ a_{n-2i-2}\ .\ .\ .\ a_0\ \overline{a_{n-1}}\ \overline{a_{n-2}}\ .\ .\ .\ \overline{a_{n-2i}}$
0	1	0	+A, $\lvert A2^{2i}\rvert_{2^k+1}$	$a_{n-2i-1}\ a_{n-2i-2}\ .\ .\ .\ a_0\ \overline{a_{n-1}}\ \overline{a_{n-2}}\ .\ .\ .\ \overline{a_{n-2i}}$
0	1	1	+2A, $\lvert A2^{2i+1}\rvert_{2^k+1}$	$a_{n-2i-2}\ a_{n-2i-3}\ .\ .\ .\ a_0\ \overline{a_{n-1}}\ \overline{a_{n-2}}\ .\ .\ .\ \overline{a_{n-2i-1}}$
1	0	0	-2A, $\lvert -A2^{2i+1}\rvert_{2^k+1}$	$\overline{a_{n-2i-2}}\ \overline{a_{n-2i-3}}\ .\ .\ .\ \overline{a_0}\ a_{n-1}\ a_{n-2}\ .\ .\ .\ a_{n-2i-1}$
1	0	1	-A, $\lvert A2^{2i}\rvert_{2^k+1}$	$\overline{a_{n-2i-1}}\ \overline{a_{n-2i-2}}\ .\ .\ .\ \overline{a_0}\ a_{n-1}\ a_{n-2}\ .\ .\ .\ a_{n-2i}$
1	1	0	-A, $\lvert A2^{2i}\rvert_{2^k+1}$	$\overline{a_{n-2i-1}}\ \overline{a_{n-2i-2}}\ .\ .\ .\ \overline{a_0}\ a_{n-1}\ a_{n-2}\ .\ .\ .\ a_{n-2i}$
1	1	1	-0	1 . . . 10 . . . 0

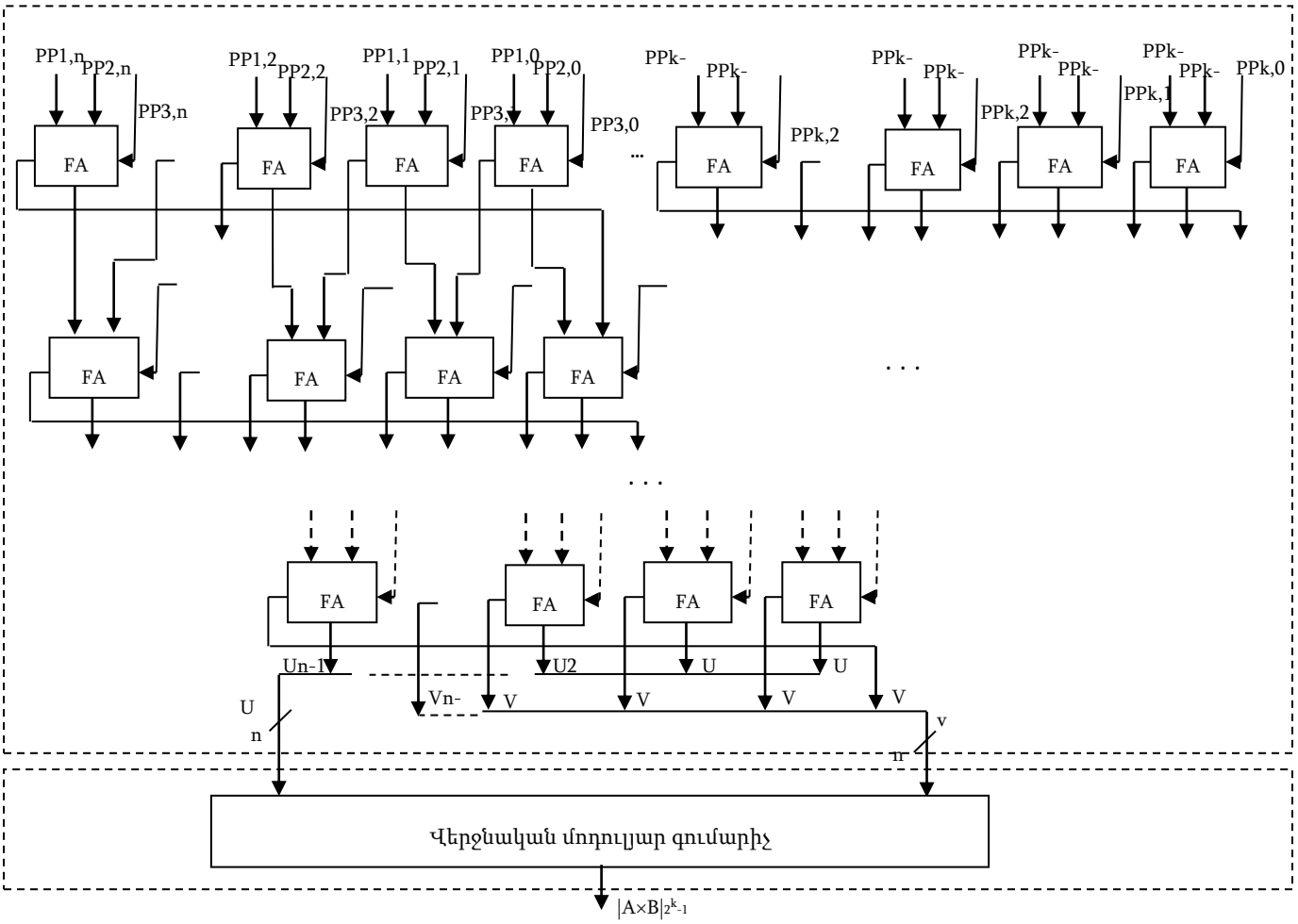
Օգտագործելով լրիվ գումարիչներ (FA-Full Adder) և կիսագումարիչներ (HA-Half Adder) կարելի է կառուցել Ուոլլեսի ծառանման բազմապատկիչ ցանկացած կարգայնության թվերի համար, և այս դեպքում գումարիչների քանակը մեծանում է $\log_2 n$ արտահայտությանը համեմատականորեն: Նույն հարաբերակցությամբ աճում է նաև բազմապատկման ընդհանուր ժամանակը:

Քանի որ Ուոլլեսի ծառը կառուցվում է փոխանցումը պահպանող գումարիչների (CSA-Carry Save Adder) հիման վրա՝ իսկ հաշվի առնելով 2^k-1 մոդուլով գումարիչների հատկությունը, դիտարկված տվյալ աշխատանքի մոդուլյար գումարիչների նախագծման բաժնում՝ կարելի է նույն սկզբունքով կառուցել 2^k-1 մոդուլով փոխանցումը պահպանող գումարիչ, բավական է, որպես հաջորդ մակարդակի մուտքային փոխանցում օգտագործել CSA-ի փոխանցման կարգը, ինչպես ներկայացված է նկ. 1.11.-ում:



Նկ. 1.11. 2^k-1 մոդուլով փոխանցման պահպանման գումարիչ

Գումարման բլոկի այս կառուցվածքի արագագործությունը կախված չէ օպերանդների կարգայնությունից և մոդուլի արժեքից: Նկ.1.11-ում պատկերված Բուտի ալգորիթմի և Ուոլլեսի ծառի կիրառմամբ 2^k-1 մոդուլով բազմապատկման սարքի կառուցվածքային սխեման կարելի է ներկայացնել CSA-ների միջոցով մասնակի արտադրյալների գումարի կազմակերպումով (նկ. 1.12):



Նկ. 1.12. Ուոլլեսի ծառի հիման վրա 2^k-1 մոդուլով մասնակի արտադրյալների գումարման իրականացումը

Դիտարկված մոդուլյար գումարիչների և բազմապատկիչների առկա մշակումները ունեն մեծ արագագործություն, քանի որ կատարում են աղյուսակային միատակտ ընտրություն: Սակայն ըստ օպերանդների կարգայնության և մոդուլի արժեքի մեծության, FPGA-ի վրա սինթեզելիս զբաղեցնում են նրա միայն LUT-ի ռեսուրսները: Պատճառը նախագծվող աղյուսակների մեծ չափերն են և, ինչն ավելի կարևոր է, դրանց ցիկլիկ կրկնությունները: Իսկ FPGA-ի տրիգերային հիշողությունը, որի միջոցով սինթեզվում է սարքի ղեկավարող ավտոմատը (FSM - Finite State Machine), և ներկառուցված բազմապատկիչները չեն օգտագործվում, քանի որ աղյուսակային ընտրանքի սկզբունքով նախագծված սարքերում FSM-ը նույնպես սինթեզվում է LUT-ի վրա [44, 51, 81]:

Հետազոտելով մոդուլյար թվաբանության բլոկում նախագծման առկա մեթոդները՝ կարելի է նշել, որ դրանց իրագործումները ԲՖ բլոկների օգտագործմամբ, սահմանափակվում են նաև այս պատճառով [22, 27, 29, 58]: Ատենախոսությունում հեղինակի կողմից առաջարկվում են նախագծման այնպիսի մեթոդներ, որոնք ինչպես կդիտարկվի աշխատանքի երկրորդ և երրորդ գլուխներում առկա մեթոդների համեմատ ունեն մի շարք առավելություններ, , FPGA-ի ռեսուրսների արդյունավետ օգտագործման տեսակետից: Նախագծման առկա մեթոդներից մի քանիսը մասամբ կիրառվել են ատենախոսությունում մշակված նոր մեթոդներում:

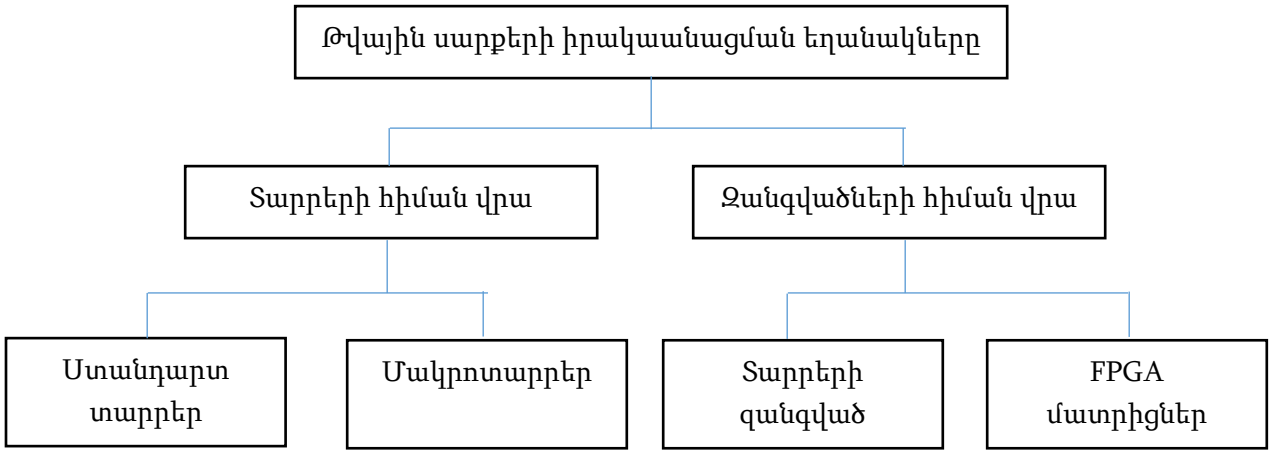
Բացի դրանից, կատարվել ենառկա և ատենախոսության աշխատանքում առաջարկված նախագծման մեթոդների համեմատական վերլուծություններ և սինթեզման արդյունքների հիման վրա ԲՖ բլոկների ռեսուրսների օգտագործման գնահատումներ:

1.4. Թվային սարքերի նախագծման արդի համակարգերի հետազոտումը և վերլուծությունը

Ժամանակակից ինտեգրալային սխեմաների կառուցվածքային բարդությունների աճի պատճառով դրանց միջոցով նախագծումը նույնպես բարդ խնդիր է: Առանց հստակ նախագծման համակարգերի մշակել սխեմաներ, որոնք պարունակում են մի քանի միլիոն տրանզիստոր գործնականում անհնար է: Պահանջները, որոնք ներկայացվում են նախագծվող սարքերին՝ արագագործությունը և արժեքն է: Շատ կարևոր է նաև նախա-

գծվող սարքերի ֆիզիկական չափերի փոքր լինելը, ինչին կարելի է հասնել ինտեգրացման աստիճանի բարձրացումով: Այս դեպքում մշակված սարքերի արժեքը կարելի է զգալիորեն փոքրացնել ավտոմատացված նախագծման համակարգերի տեխնոլոգիաների օգտագործմամբ, որոնք չնայած չեն ապահովում առավելագույն արագագործություն, սակայն նվազեցնում են սխեմաների մշակման համար անհրաժեշտ ժամանակը [45, 72, 83]:

Թվային սարքերի նախագծման արդի գործընթացում ներառված տարրերի և եղանակների միավորված հիերարխիկ մոտեցումը ներկայացված է նկ. 1.13-ում:



Նկ. 1.13. Թվային սարքերի նախագծման հիերարխիկ մոտեցումը

Ստանդարտ տարրերի օգտագործումը ենթադրում է մշակվող սարքերի ստանդարտավորումը տրամաբանական տարրերի մակարդակով: Այս դեպքում ավտոմատացված համակարգի գրադարանը պարունակում է տրամաբանական տարրերի մեծ քանակ: Բացի հիմնական տրամաբանական ֆունկցիաներից (օրինակ՝ ԵՎ-ՈՉ, ԿԱՄ-ՈՉ, ժխտում և այլն), տիպային գրադարանը պարունակում է նաև ավելի բարդ ֆունկցիաներ՝ մուլտիպլեքսոր, լրիվ գումարիչ, կոդավորիչ և այլն [2, 44]:

Այն սխեմաները, որոնք իրականացնում են ավելի բարդ ֆունկցիաներ, քան գրադարանում առկա ստանդարտ տարրերը կոչվում են ***մակրոտարրեր կամ մակրոբջիջներ***: Որպես մակրոտարրերի օրինակ կարելի է դիտարկել հիշող սարքերը և բազմապատկիչները:

Տարրերի զանգվածի նախագծման համակարգը ենթադրում է բազային սալիկներ, որոնք պարունակում են տարրական բջիջներ կամ տրանզիստորներ: Բազային բյուրեղներով այս սալիկների իրական կառուցվածքի ձևափոխելու համար անհրաժեշտ են միայն միջբջջային կապեր:

Եվ վերջապես, ծրագրավորվող բյուրեղներ (***FPGA մատրիցներ***), որոնց միջոցով կարելի է իրականացնել բուլյան ֆունկցիաների տրված հավաքածուները: Այս տարբերակի համար FPGA-ների արտադրության պրոցեսը լիովին առանձնացված է նրա ծրագրավորման գործընթացից, ինչը հնարավորություն է տալիս բաժանել նրա արժեքը բազմաթիվ նախագծվող սարքերի միջև [78]:

Քանի որ տվյալ աշխատանքում մոդուլյար թվաբանության բլոկում նախագծվող սարքերը իրագործված են FPGA-րի օգտագործմամբ,ուստի դիտարկենք այն խնդիրները, որոնք անհրաժեշտ է լուծել FPGA-երի վրա ծրագրավորման ժամանակ.

- ինչպես իրականացնել ծրագրավորվող տրամաբանությունը,
- որտեղ պահել այն ծրագիրը (կոնֆիգուրացիան), որը տրվում է ծրագրավորվող մատրիցին:

Առանձնացվում է FPGA-րի ծրագրավորման երեք տեխնոլոգիա.

- միանվագ գրանցումով FPGA:
- էներգաանկախ FPGA:
- էներգակախյալ FPGA:

Բյուրեղի վրա ժամանակակից համակարգերի նախագծողը կարող է կատարել ընտրություն բազմաթիվ իրացումների առումով, մասնավորապես՝

- արագագործության, հզորության և արժեքի,
- նախագծի բարդության,
- թեստավորման հնարավորության և այլ տեսակետներից:

Էներգակախյալ և էներգաանկախ FPGA-րի կիրառմամբ կարելի է նախագծել տարբեր բարդության և տրամաբանության սարքեր, և, որպես հնարավոր տարբերակներ, դիտարկենք երկու արդի ճարտարապետություն՝ Altera սերիայի MAX սարքերի ընտանիքը և Xilinx սերիայի XC40xx սարքերի ընտանիքը [2, 38, 43, 45]:

Նախագծման ժամանակակից մեթոդները և գործիքային միջոցները մեծ ազդեցություն ունեն արդի նախագծման գործընթացի կազմակերպման և իրականացման վրա, դրանք թույլ են տալիս ստեղծել համապատասխան մասնագիտացված թվային սարքեր և սարքավորումներ, որոնք նախատեսված են կոնկրետ կիրառման համար (մեր դեպքում մոդուլյար թվաբանության բլոկում թվային սարքրերի նախագծում և, օրինակ, կրիպտոպրոցեսորներ):

Թվային սարքերի նախագծման արդի համակարգերի հետազոտումները հիմնավորում են դրանց միջոցով մասնագիտացված սխեմաների իրագործման մեծ հնարավորությունները, հատկապես կարելի է նշել վերածրագրավորման և թեստավորման հնարավորությունները, ինտեգրացման բարձր աստիճանը և գնային տեսակետից լայն ընտրությունը: Համաշխարհային շուկայում FPGA-ի հիմնական արտադրողներ ճանաչվում են "Altera", "Xilinx", "Lattice semiconductor", "Actel" ընկերությունները [87, 99, 101, 102]:

Ատենախոսությունում մշակված մոդուլյար սարքերի սխեմաների նկարագրումները կատարվել են Verilog HDL-ի միջոցով, իսկ որպես սարքերի աշխատանքի մոդելավորման և սխեմաների սինթեզման միջոց՝ դիտարկվել են երկու՝ Xilinx ընկերության ISE DESIGN Suite և Altera ընկերության Quartus II փաթեթները: Նշված փաթեթներն ունեն մեծ առավելություն, քանի որ կիրառողի համար անվճար հասանելի են համացանցում և ունեն ինստալյացիոն պարզ հրահանգներ:

Հետազոտելով նշված ընկերությունների արտադրած FPGA-ը և վերլուծելով սինթեզող փաթեթների կիրառական հնարավորությունները առաջնությունը տրվել է Xilinx ընկերության ISE DESIGN Suite սինթեզող փաթեթին: Հետազոտությունների արդյունքները կարելի է ներկայացնել աղյ. 1.9-ի միջոցով:

Աղյուսակում ներկայացված Xilinx և Altera ընկերությունների արտադրած սարքերի տարբերություններից կարելի է առանձնացնել և կարևորել.

- բլոկների ներքին կառուցվածքները և սինթեզող փաթեթի ծրագրավորվող տրամաբանական սարքի (PLD) հիմնական կիրառական ուղղվածությունը: Xilinx ընկերության ISE DESIGN Suite փաթեթը սարքերի սինթեզումն իրականացնում է FPGA-ի վրա (Spartan-3, Spartan-3E, Spartan-6), իսկ Altera

ընկերության Quartus II փաթեթը՝ CPLD-ի վրա (Max II, Max V): FPGA-ի ճարտարապետությունն անհամեմատ ավելի ճկուն է տրիգերային հիշողության և ներկառուցված բազմապատկիչների շնորհիվ:

- Place & Route (սարքի ֆիզիկական տեղը FPGA-ի վրա և երթուղավորում) ֆունկցիայի առկայությունը: ISE DESIGN Suite փաթեթում այս ֆունկցիայի շնորհիվ նախագծողը ինքնուրույն կարող է որոշել սարքի FPGA-ում ծրագրավորման ֆիզիկական տեղը և անհրաժեշտության դեպքում երթուղավորել նրա կապերը նույն FPGA-ի վրա այլ սարքերի հետ:

Աղյուսակ. 1.9. ISE DESIGN Suite և Quartus II սինթեզող փաթեթների համեմատություն

	Xilinx ընկերություն և ISE DESIGN Suite սինթեզող փաթեթ	**Altera ընկերություն և Quartus II սինթեզող փաթեթ**
FPGA-ի արտադրությունը	Xilinx ընկերություն - 50%	Altera ընկերություն – 33%
FPGA-ի հիմնական ընտանիքները	Spartan, Artix, Kintex, Virtex	Cyclone, Arria, Stratix
Ծրագրավորվող սարքերի ներքին բլոկները	Կոնֆիգուրացվող տրամա-բանական բլոկներ (CLB)	Տրամաբանական բջիջներ (LC)
Բլոկների ներքին կառուցվածքները	LUT, տրիգերներ, մուլտիպլեքսորներ, ներկառուցված բազմապատկիչներ	մուլտիպլեքսորներ
Սինթեզող փաթեթի ծրագ-րավորվող տրամաբանական սարքի (PLD) հիմնական կիրառական ուղղվածությունը	FPGA, որոնք պարունակում են ծրագրավորվող LUT աղյուսակներ	CPLD, որոնք պարու-նակում են խոշորացված տրամաբ. բլոկներ տարրերի հիման վրա (2 ԵՎ-ՈՉ / 2 ԿԱՄ-ՈՉ)
FPGA-ի վրա նախագծվող սարքի իրականացման փուլերը	Place & Route (սարքի ֆիզիկական տեղը FPGA-ի վրա և երթուղավորում) ֆունկցիայի առկայությունը	Place & Route ֆունկցիայի բացակայությունը
Նախագծվող սարքի ֆայլի իրականացումը	Բիթ-ֆայլ	Ասեմբլեր

Ելնելով վերը նշվածից՝ որպես սարքերի աշխատանքի մոդելավորման և սխեմաների սինթեզման միջոց ընտրվել է Xilinx ընկերության ISE DESIGN Suite փաթեթը:

1.5. Աշխատանքի նպատակը և հետազոտության խնդիրները

Ուսումնասիրելով մոդուլյար թվաբանության հիմնական օրենքները և գործառության սկզբունքները, նախագծման արդի մեթոդները և մոդուլյար հաշվարկների իրականացման եղանակները, ինչպես նաև մոդուլյար թվաբանության բլոկում թվաբանական գործողությունների կատարման ալգորիթմները և կիրառման առանձնահատկությունները արդիական է մշակել մոդուլյար թվաբանության բլոկում մասնագիտացված ԲՖ բլոկների նախագծման մեթոդներ և միավորել դրանք մեկ երկխոսային ավտոմատացված բաց համակարգում, ինչը կօգնի իրականացնել թվային սարքերի կառուցվածքների մոդելավորում, սինթեզում, համեմատում և գնահատում, որն էլ տվյալ ատենախոսական թեզի նպատակն է:

Այս նպատակի իրականացման համար կատարվել են հետևյալ հետազոտությունները, մոդուլյար սարքերի մշակումներ և վերլուծություններ՝

1. Վերլուծվել և համակարգվել են ինտեգրալային սխեմաների նախագծման մոտեցումները և առկա մեթոդները:
2. Մշակվել են տարբեր մեթոդներով մոդուլյար գումարիչների ապարատային իրականացումներ, վերլուծվել են դրանց արագագործության և ապարատային ծախսերը:
3. Մշակվել են տարբեր մեթոդներով մոդուլյար բազմապատկիչների ապարատային իրականացումներ, վերլուծվել են դնրանց արագագործության և ապարատային ծախսերը:
4. Մշակվել են մոդուլով աստիճան բարձրացնելու սարքերի ապարատային իրականացումներ, վերլուծվել են դնրանց արագագործության և ապարատային ծախսերը:
5. Վերլուծվել են մոդուլյար Բ բլոկների ապարատային կառուցվածքները կախված մոդուլից և կիրառման ոլորտից:
6. Մշակվել է մոդուլյար թվաբանության բլոկի համար ավտոմատացված նախագծման համակարգը:

Գլուխ 1-ի եզրահանգումներ

Գլուխ 1-ում կատարված հետազոտությունների և վերլուծությունների հիման վրա կարելի ներկայացվում են հետևյալ եզրահանգումները.

1. Մոդուլյար թվաբանությունը դիտարկվում է որպես արագագործության բարձրացման և հաշվարկների ապահովության միջոց, որն ունի նաև կիրառական տարբեր ոլորտներ: Առանձին ներկայացված են մնացորդային դասերի համակարգում թվերի ներկայացման ավանդական և արագագործ մեթոդները:
2. Թվային սարքերի նախագծման այնպիսի մեթոդոլոգիաներ, ինչպիսիք են SoC նախագծումը, ԲՖ բլոկները, դրանց բաղադրիչները և կրկնակի օգտագործման մակարդակները նպատակահարմար է կիրառել մոդուլյար սարքերի իրագործման համար: Դիտարկված են SoC նախագծման մեթոդաբանության առավելությունները և նշված մեթոդաբանության ստեղծման նախապայմանները:
3. Մոդուլյար թվաբանության բլոկում նախագծման առկա մեթոդները, ինչպիսիք են մոդուլյար գումարիչների նախագծումները LUT աղյուսակների կիրառմամբ և մոդուլյար բազմապատկիչների իրագործումները, հիմնված ինդեքսային մեթոդի, քառակուսիների օրենքի և Բուտի ալգորիթմի վրա, SoC մեթոդոլոգիան կիրառելիս ունեն ԲՖ-բլոկների նախագծման առանձնահատկություններ: Որպես այդպիսիններ կարելի է դիտարկել ընտրված ԲՖ բլոկների տրիգերային հիշողության և ներկառուցված բազմապատկիչների ոչ արդյունավետ օգտագործումը:
4. Հիմնավորել է մոդուլյար թվաբանության բլոկում համակարգերի նախագծման և սինթեզման միջոցի՝ Xilinx ընկերության ISE DESIGN Suite փաթեթի նպատակահարմարությունը և կիրառումը:
5. Ձևավորվել են աշխատանքի նպատակը և հետազոտության հիմնական խնդիրները:

ԳԼՈՒԽ 2. ՄՈԴՈՒԼՅԱՐ ԹՎԱԲԱՆՈՒԹՅԱՆ ԲԼՈԿՈՒՄ ՍԱՐՔԵՐԻ ՆԱԽԱԳԾՄԱՆ ՄԵԹՈԴՆԵՐԻ ՄՇԱԿՈՒՄԸ

Ինչպես արդեն նշվել է, թվային սարքերին ներկայացվող արագագործության բարձրացման պահանջներ առաջադրելու պատճառով սկսեցին օգտագործել մնացորդային դասերի համակարգ և մոդուլյար հաշվարկներ, որը հնարավորություն է տալիս էականորեն բարձրացնել թվաբանական գործողությունների կատարման արագագործությունը, մնացորդների հետ կատարվող գործողությունների զուգահեռաբար կատարման շնորհիվ: Ունենալով զուգահեռականացման հնարավորություն մոդուլյար հաշվարկները պահանջում են մոդուլյար կապուղիներում թվաբանական գործողությունների կատարման մասնագիտացված հատուկ սարքերի մշակում:

Քանի որ մնացորդային դասերի համակարգը օգտագործում է մոդուլյար գործողություններ, ուստի արդյունավետության բարձրացման նպատակով թվաբանական սարքերում անհրաժեշտ է օգտագործել մնացորդային դասերի համակարգի համար հատուկ նախագծված մոդուլյար գումարիչներ, բազմապատկիչներ, մոդուլյով աստիճան բարձրացման գործողություն կատարող սարքեր:

2.1. Մոդուլյար գումարիչների նախագծման մշակված մեթոդները

Ժամանակակից ապարատային բազան հնարավորություն է տալիս հանրահաշվական գործողությունները փոխարինել մնացորդների հետ միատակտ աղյուսակային ընտրանքներով: Այսինքն, մշակվող մնացորդների փոքր կարգայնությամբ կարելի է արագագործության բարձրացման համար կիրառել աղյուսակային տեղադրման մեթոդները, սակայն մոդուլյար գումարիչներ կարելի է նախագծել և հիշող սարքերի կիրառմամբ , և առանց վերջինների [16, 17, 20, 31]:

Առանց LUT աղյուսակի օգտագործման մոդուլյար գումարիչների նախագծում

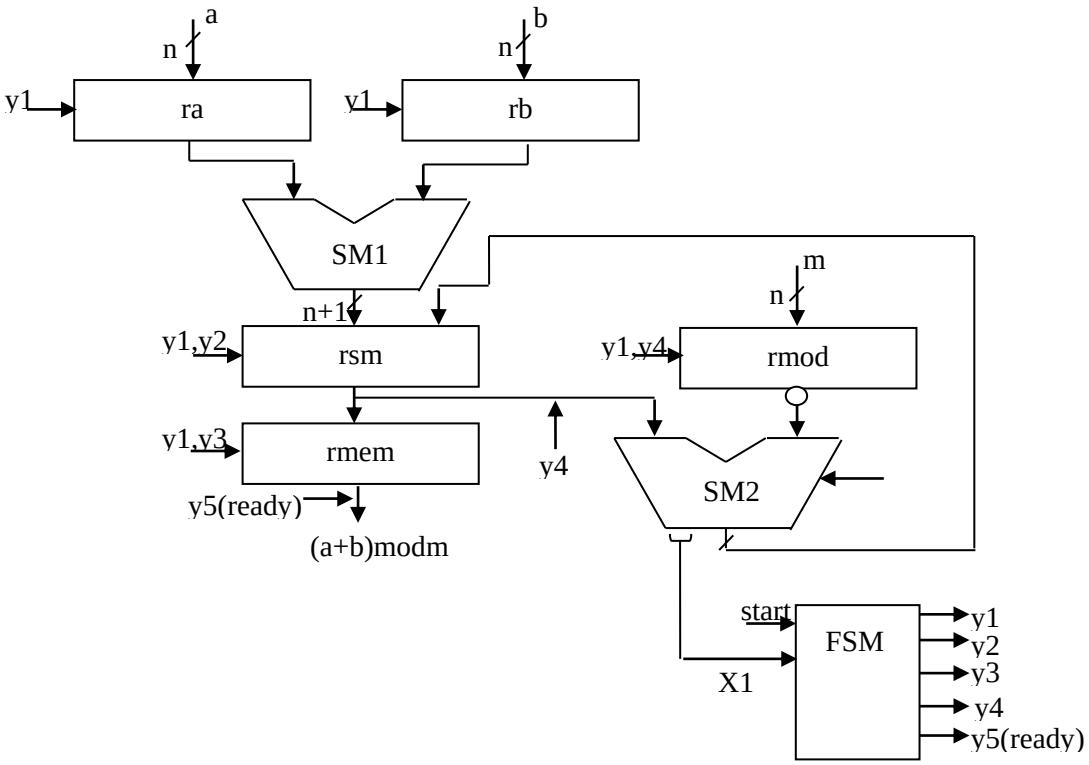
Տվյալ աշխատանքում մշակված մոդուլյար գումարիչի նախագծման մեթոդը գերադասելի է մոդուլի մեծ արժեքների դեպքում: Բացի դրանից, եթե սինթեզումն իրագործվում է FPGA-ների օգնությամբ, ապա վերջիններս ունեն սահմանափակ քանակի ներկառուցված տրամաբանություն և անհամեմատ ավելի շատ քանակով

տրիգերներ, հետևաբար կառավարող ավտոմատի առկայությունը նույնպես ունի իր առավելությունը [23, 40, 44]:

Տվյալ մեթոդով նախագծված մոդուլյար գումարիչը գործողությունը չի կատարում աղյուսակային ընտրանքի միջոցով: Հիմնվելով արտահայտություն (2.1)–ի վրա՝ մշակվել է մոդուլյար գումարիչի կառուցվածքային սխեմա, որը համալրված ՝ կառավարող ազդանշաններով, այսինքն՝ ենթադրվում է FSM-ի առկայությունը (նկ. 2.1)[46]:

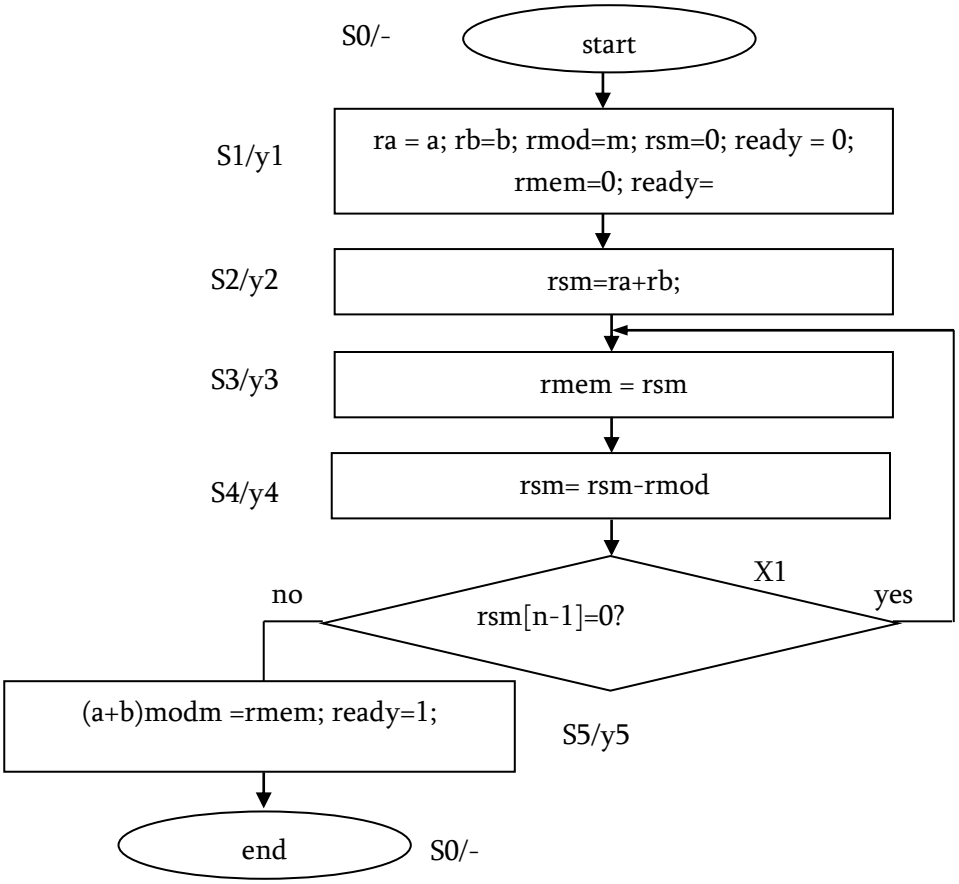
$$(a+b)\text{mod}m = \begin{cases} (a+b), & 0 \le a+b \le m \\ a+b - km, & a+b > m \end{cases} \qquad (2.1)$$

որտեղ $k = 1, 2, 3, \ldots (a+b)/m$:



Նկ.2.1. Մոդուլյար գումարման սարքի կառուցվածքային սխեման

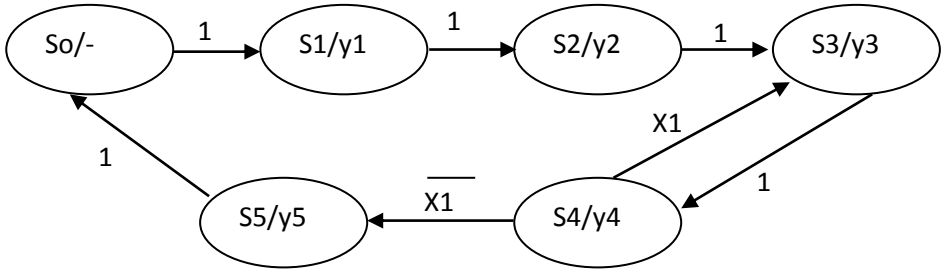
Կառավարող ազդանշանները նշված են նկ. 2.2-ում բերված ալգորիթմի բլոկ-սխեմայի օպերացիոն բլոկներին համապատասխան, և ձևավորվում են FSM (Finite State Machine) կառավարող ավտոմատի կողմից:



Նկ. 2.2. Առանց LUT-աղյուսակի օգտագործման մոդուլյար գումարիչի սխեման

Երկու n կարգանի a և b թվերը տեղակայվում են համապատասխանաբար ra և rb ռեգիստրներում: m մոդուլի լրացուցիչ կոդը տեղակայվում է rmod ռեգիստրում: SM1 գումարիչը կատարում է սովորական երկուական գումարում, որի արդյունքը տեղակայվում է rsm ռեգիստրում և կրկնօրինակվում rmem ռեգիստրում: SM2 գումարիչը հաշվարկում է a+b-m տարբերությունը, ինչից հետո վերլուծվում է այդ արժեքի նշանը և որոշվում. կատարել ցիկլի կրկնություն, թե rmem ռեգիստրի վերջին պարունակությունը արդեն (a+b)modm արտահայտության արժեք է: k գործակիցը ցույց է տալիս rsm ռեգիստրի նշանի ստուգման X1 ստուգող պայմանի ցիկլերի կատարման քանակը: (a+b)modm արտահայտության արժեքի որոշումից հետո կառավարող ավտոմատը հաղորդում է օպերացիոն սարքին ready ազդանշանը, որը դիտարկվում է որպես մոդուլյար գումարման գործողության կատարման ավարտ:

Կառավարող ավտոմատի համար կատարված է ալգորիթմի բլոկ-սխեմայի Մուրի մոդելի նշանադրում, որի անցումների գրաֆը ներկայացված է նկ. 2.3-ում:



Նկ. 2.3. Մոդուլյար գումարման սարքի կառավարող ավտոմատ Մուրի մոդելի անցումների գրաֆը

Այսպիսով, մոդուլյար գումարիչի նախագծման առաջարկված մեթոդը, հիմնված (2.1) արտահայտության վրա և մշակված կառուցվածքային սխեման (նկ. 2.1) համարվում են արագագործության տեսակետից առավել արդյունավետ, համեմատած ավանդական ըստ մոդուլի գումարման հետ: Նշվածը պայմանավորված է նրանով, որ ավանդական ըստ մոդուլ m գումարումը, այսինքն երկուական գումարում, ապա հաջորդող բաժանումը մոդուլ m-ի արժեքին: Քանի որ բաժանարարի (մոդուլ m-ի արժեքը) նորմալացումը, քանորդի յուրաքանչյուր կարգի որոշումը և, ի վերջո, վերջնական մնացորդի, որը և կհանդիսանա (a+b)modm արտահայտության արժեքը, հնարավոր նորմալացման անհրաժեշտությունը կդառնա արագագործության անկման պատճառ: Բացի դրանից, Verilog լեզվով RTL-նկարագրության համար առանձին բլոկներ նկարագրվում են always պրոցեդուրային օպերատորի միջոցով, իսկ եթե այն կոմբինացիոն սխեմա է, ապա կարող է նկարագրվել նաև assign օպերատորի կիրառմամբ: Արդյունքում կստացվի սարքի RTL-նկարագրում, որի հիման վրա կարելի է սինթեզել սխեման[4]:

<u>2^n+1 և 2^n-1 մոդուլներով արագագործ մոդուլյար գումարիչների նախագծումը BDD-տեխնոլոգիայի հիման վրա:</u>

Առաջարկվում է 2^n+1 և 2^n-1 մոդուլներով արագագործ մոդուլյար գումարիչներն նախագծել՝ կիրառելով երկուական լուծումների դիագրամների տեխնոլոգիան (Binary

Decision Diagram (BDD)): Տվյալ տեսակի մոդուլները լայնորեն կիրառվում են մոդուլյար թվաբանության բլոկում՝ սխեմաների կառուցման համար [16, 41, 79]:

2^n-1 մոդուլով մոդուլյար գումարիչները կարելի է նախագծել բարձր կարգից ցածր կարգ ցիկլիկ փոխանցմամբ գումարիչների հիման վրա: Դա պայմանավորված է նրանով, որ 2^n-1 մոդուլով մոդուլյար գումարումը կարող է ձևափոխվել 2^n մոդուլով գումարման՝ համաձայն բանաձև (2.2)-ի:

$$|a+b|_{2^n-1} = |a+b+c_{out}|_{2^n} : \quad (2.2)$$

Դասական BDD-ի յուրաքանչյուր գագաթ համապատասխանում է Շենոնի վերլուծությանը [44] և ապարատային իրագործումը կատարվում է մուլտիպլեքսորների հիման վրա: Սակայն փոխանցման ազդանշանը հաշվարկելիս, նպատակահարմար է օգտագործել գեներացման (g_i) և փոխանցման տարածման (p_i) ֆունկցիաները:

$$p_i = a_i \vee b_i, \quad (2.3)$$

$$g_i = a_i \oplus b_i, \quad (2.4)$$

$$c_{i+1} = g_i \vee p_i c_i, \quad (2.5)$$

Նմանատիպ գումարիչների արագագործությունը որոշվում է ելքային փոխանցման ֆունկցիայի հաշվարկման ամենակարճ ձևավորման ճանապարհով: Կառուցենք 2^n-1 մոդուլով գումարիչների փոխանցման ֆունկցիաների համակարգի ֆունկցիոնալ դիագրամը (Functional Decision Diagram (FDD)), համաձայն (2.2) բանաձևի, հիմնվելով ավանդական երկուական գումարիչի ֆունկցիոնալ դիագրամի վրա, համաձայն (2.3), (2.4) և (2.5) բանաձևերի:

Հարկ է նշել, որ մոդուլյար գումարիչի լիարժեք իրականացման համար անհրաժեշտ է կիրառել լրացուցիչ տրամաբանություն, որը չի մեծացնում հապաղումը, փաստորեն չի ազդում արագագործության վրա, քանի որ իրագործվում է հիմնական կառուցվածքին զուգահեռ:

2^n-1 մոդուլով գումարիչի կառուցումը ներառում է հետևյալ փուլերը.

- g_i գեներացման և p_i փոխանցման տարածման ֆունկցիաները իրականացնող բլոկի ձևավորում,

- c_i փոխանցման ֆունկցիան իրականացնող FDD-ի հիման վրա կառուցված բլոկի ձևավորում,
- մոդուլյար գումարի վերջնական արժեքը $S_i = a_i \oplus b_i \oplus c_i$ հաշվարկող բլոկի ձևավորում:

Այսպիսով, հիմնվելով բանաձևեր (2.3), (2.4) և (2.5)-ի և BDD տեխնոլոգիայի վրա [6], կարելի է փաստել, որ 2^n-1 մոդուլով գումարիչը չի զիջում արագագործությամբ նույն կարգայնությամբ երկուական գումարիչներին, իսկ կառուցման տեսանկյունից ավելի պարզ է:

Նույն սկզբունքով կարելի է կառուցել 2^n+1 մոդուլով գումարիչներ, ուր գումարման գործողությունը 2^n+1 մոդուլից բերվում է 2^n մոդուլի գումարմանը՝ ներկայացված բանաձև (2.6)-ի միջոցով:

$$|a+b|_{2^n+1} = |a+b+z|_{2^n} + c_{out}\,, \tag{2.6}$$

որտեղ $z = 2^{n+1} - (2^n + 1) = 2^n - 1$ մոդուլի արժեքի լրացումն է մինչև 2^n+1 աստիճան, իսկ cout-ը (a+b+z) գումարի ավագ բիթն է:

Բանաձև (2.6)-ից բխում է, որ |a+b|mod(2^n+1)-ն որոշելու համար նախ անհրաժեշտ է հաշվարկել (a+b+z) արտահայտությունը, այնուհետև դրան գումարել ավագ բիթի ժխտումը: Կոնստանտ z-ի ավելացումը նպատակահարմար է իրագործել CSA տիպի գումարիչների միջոցով, որը հնարավորություն է տալիս գումարել 3 թիվ միակարգ լրիվ գումարիչի հապաղումով: Հաջորդ քայլով կգումարենք ստացված կարգային արժեքները և համապատասխան փոխանցումները, որպեսզի ձևավորենք վերջնական փոխանցման ժխտումը:

2^n+1 մոդուլով գումարիչի կառուցումը ներառում է հետևյալ փուլերը.

- CSA գումարիչի միջոցով մուտքային օպերանդների գումարին կոնստանտ z-ի ավելացումը իրականացնող բլոկի ձևավորում,
- g_i գեներացման և p_i փոխանցման տարածման ֆունկցիաները իրականացնող բլոկի ձևավորում,
- c_{in} և c_{out} փոխանցման ֆունկցիաներն իրականացնող FDD-ի հիման վրա կառուցված բլոկի ձևավորում,

- մոդուլյար գումարի վերջնական արժեքը $S_i = a_i \oplus b_i \oplus c_i$ հաշվարկող բլոկի ձևավորում:

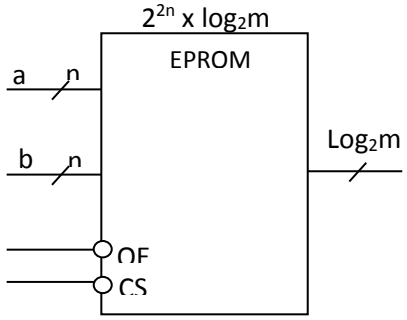
2.2. Մոդուլյար բազմապատկիչների նախագծման մեթոդները

Գլուխ 1-ում ներկայացված մոդուլյար բազմապատկիչների նախագծման առկա մեթոդներից բացի, ատենախոսությունում մշակվել են նախագծման նոր մեթոդներ, որոնցից մի քանիսը հիմնված են առկա մեթոդների վրա և ունեն որոշակի առավելություններ ԲՖ բլոկների (FPGA) օգտագործմամբ նախագծման տեսակետից, ինչպես արդեն նշվել է, այս FPGA-երն ունեն սահմանափակ քանակի ներկառուցված տրամաբանություն և անհամեմատ ավելի շատ քանակով տրիգերներ, հետևաբար կառավարող ավտոմատի առկայությունը նույնպես ունի իր առավելությունը:

Դիտարկված են մնացորդային դասերի համակարգում բազմապատկման սարքերի նախագծման սկզբունքները՝ հիշող սարքերի և ժամանակակից ինտեգրալ սխեմաների օգտագործմամբ: Այս խնդրի լուծումը հիմնված է մոդուլյար թվաբանության հայտնի օրենքների և բազմապատկման գործողության կատարման ալգորիթմների (մասնավորապես՝ ինդեքսային մեթոդով մոդուլյար բազմապատկման) վրա:

<u>Հիշող սարքի միջոցով մոդուլյար բազմապատկիչների նախագծում</u>

Նկ. 2.4.-ում պատկերված սխեման հնարավորություն է տալիս ստանալ (axb) mod m արտահայտության արժեքը PROM (Programmable Read-Only-Memory) տիպի հիշող սարքի միջոցով:



Նկ. 2.4. EPROM-ի միջոցով մոդուլյար բազմապատկիչի իրագործումը

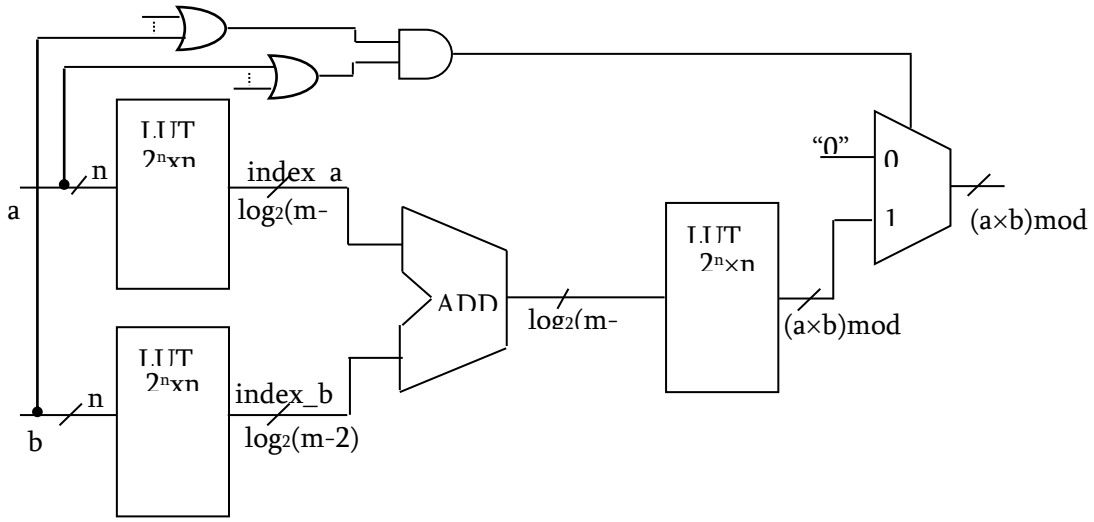
Երկու a և b թվերը՝ որպես հասցեական մուտք, ներմուծվում են PROM, որտեղից և տվյալ հասցեով ընտրվում է նախապես գրանցված (axb)modm արժեքը: Ելնելով տվյալ

սկզբունքից՝ կարելի է օգտագործել նաև LUT (Look-Up-Table) – աղյուսակ՝ $2^{2^n} x \log_2 m$ չափերով: Այսպիսի լուծումը շատ հարմար է փոքր կարգայնությամբ a և b թվերի համար, քանի որ n-ի և m-ի արժեքների աճի դեպքում աճում են նաև հիշող սարքի կամ LUT աղյուսակի չափերը: Օրինակ, ենթադրենք առկա է PROM, որի միջոցով իրագործված է m=7 մոդուլով a=11 և b=23 երկու n=8 կարգանի թվերի բամապատկում: PROM հիշող սարքի հասցեական մուտքերին կձևավորվի 00001011 00010111 երկուական կոդը, որտեղ հասցեի առաջին ութ բիթերը a=11 թվի երկուական ներկայացումն է, իսկ հաջորդ ութ բիթերը՝ b = 23 թվի երկուական ներկայացումը: Հետևաբար PROM հիշող սարքի 2839-րդ բջջում պահված կլինի (axb) mod m բանաձևին համաձայն՝ (11x23)mod 7= 253mod7=1 տասական թվի $(001)_2$ եռակարգ երկուական ներկայացումը, քանի որ $\log_2 7 < 3$:

Եթե a կամ b օպերանդներից (արտադրիչներից) որևէ մեկը հավասար է զրոյի, ապա “ԵՎ” սխեմայի ելքում կձևավորվի տրամաբանական “0”, ինչը կոմպարատորի ելքում կձևավորի տրամաբանական “1” արժեքը, որի միջոցով էլ մուլտիպլեքսորի ելք փոխանցվելու է “0” արժեք, հակառակ դեպքում մուլտիպլեքսորի ելք փոխանցվելու է LUT3-ից հակադարձ ձևափոխման արժեքը[7]:

Ինչպես ինդեքսային մոդուլյար բազմապատկիչների, այնպես էլ զուգահեռ ինդեքսային մոդուլյար բազմապատկիչների ապարատային իրագործման ժամանակ անհրաժեշտ է դիտարկել նաև այն դեպքը, երբ օպերանդներից (արտադրիչներից) մեկը (կամ երկու օպերանդը միաժամանակ) հավասար է զրոյի: Ինչպես երևում է աղ.1.5-ից, ցանկացած m պարզ մոդուլով բազմապատկման համար q ամբողջ թվերի բազմությանը $\{q\} = \{1,2,3,\dots,m-1\}$ համապատասխանում է i ինդեքսների բազմություն՝ $\{i\} = \{1,2,3,\dots,m-2\}$, այսինքն օպերանդի զրոյի հավասար լինելը LUT1 և LUT2 աղյուսակներում արտապատկերված չէ: Այս դեպքը պահանջում է լրացուցիչ սխեմա մուտքային օպերանդները “0”-ի հետ համեմատելու համար:

Ինդեքսային մոդուլյար բազմապատկման սարքի կառուցվածքային սխեմային ավելացված է համեմատման սխեմայից (կոմպարատորից) և մուլտիպլեքսորից բաղկացած բլոկ (նկ. 2.5):



Նկ. 2.5. Մուտքային օպերանդների զրոյի ստուգման իրականացմամբ ինդեքսային մեթոդով մոդուլյար բազմապատկման սարքի կառուցվածքային սխեման

Զուգահեռ ինդեքսային մոդուլյար բազմապատկիչներում զրոյական օպերանդների ստուգումը կատարվում է նույն սկզբունքով:

Ելնելով վերը նշվածից՝ կարելի է եզրակացնել, որ մոդուլյար բազմապատկիչների նախագծման առաջարկված մեթոդներն ավելի արդյունավետ են արագագործության տեսանկյունից (3 տակտ՝ 1-ական տակտ LUT-երից ընթերցելու և 1 տակտ գումարման համար)՝ համեմատած ավանդական ըստ mod m-ի բազմապատկման մեթոդի, այսինքն՝ սովորական բազմապատկում և հաջորդող բաժանում m-ի մոդուլի:

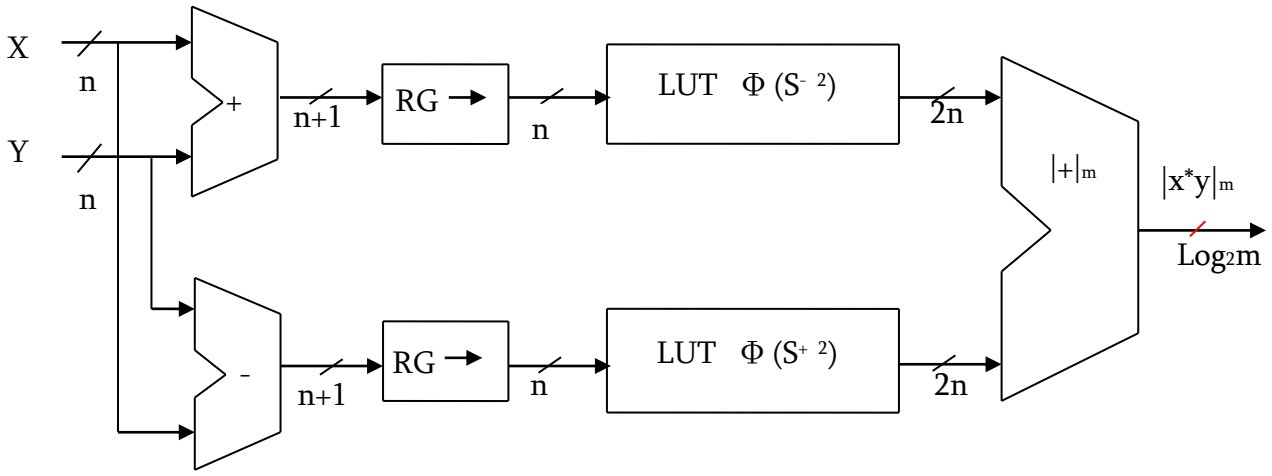
Քանի որ բաժանարարի (մոդուլ m-ի արժեքի) նորմալացման, քանորդի յուրաքանչյուր բիթի հաշվարկման և, ի վերջո, վերջնական մնացորդի, որը և կհանդիսանա $(a \cdot b)\ mod m$ արդյունքը, հնարավոր նորմալացման անհրաժեշտ գործընթացները կարող են արագագործության նվազեցման պատճառ դառնալ: Նշված գործողությունների համար ծախսվում է՝ $n + 2n - \log_2 m + 3$ տակտ՝ n տակտ բազմապատկման, $2n - \log_2 m + 1$տակտ քանորդի կարգերի որոշման և 2 տակտ բաժանարարի և վերջնական մնացորդի նորմալացման համար:

Հարկ է նշել, որ նմանատիպ բազմապատկիչները կարող են կիրառվել մասնագիտացված սարքերում՝ կոդերի հսկման, գաղտնագրման, ազդանշանների մշակման և այլ ոլորտներում: Անհրաժեշտ է նաև նշել, որ ինդեքսային մեթոդով մոդուլյար բազմապատկիչների նախագծման առաջարկված տարբերակները աշխատունակ են

միայն այն դեպքում, երբ m-ը պարզ թիվ է: Այլ տարբերակների համար կարելի է առաջարկել տարբեր (օրինակ՝ քառակուսիների օրենքի վրա հիմնված, Բուտի ալգորիթմով գործող և այլն) մեթոդներ:

Քառակուսիների օրենքի հիման վրա մոդուլյար բազմապատկիչների նախագծում

Գլուխ 1-ում դիտարկված քառակուսիների օրենքի հիման վրա մոդուլյար բազմապատկման սխեման կարելի ձևափոխել, փոխարինելով, մինչև LUT աղյուսակներում համապատասխան էյլերի ֆունկցիայով որոշվող արժեքների գրանցումը 2-ի բաժանող սարքերը մեկ բիթ աջ տեղաշարժող ռեգիստրներով (նկ. 2.6):



Նկ. 2.6. Տեղաշարժող ռեգիստրների կիրառմամբ մոդուլյար բազմապատկման սխեման ըստ քառակուսու օրենքի

Երկուսի բաժանման գործողությունը վտանգում է գումարման և հանման գործողություններից հետո ձևավորվող միջանկյալ արժեքների ամբողջ թիվ մնալը, և, հետագայում, LUT-աղյուսակից ճիշտ արդյունքի ընտրումը, ինչը, իր հերթին կվտանգի վերջնական արդյունքի ճշտությունը: Բացի դրանից m մոդուլով 2 թվի բաժանումը՝ $|2|^{-1}m$ երաշխավորվում է միայն, եթե m-ը կենտ թիվ է, այս դեպքում բաժանելիս ստացված ½ արժեքները հետագա գումարելիս կոնպենսացվում են և արդյունքում ստացվում է ճիշտ պատասխան:

Օրինակ . Հաշվենք (axb)mod m, a=3, b=7, m=5: (m-ը չի կարող լինել զույգ):

$$(3 \cdot 5) mod\ 5 = 21\ mod\ 5 = 1$$

Տեղադրելով a և b թվերի արժեքները բանաձև (5)-ում կստանանք նույն արժեքը:

$$\left(\frac{1}{4}\right)|(3+7)^2-(3-7)^2| \; mod \; 5 = \left(\frac{1}{4}\right)|10^2-4^2| \; mod \; 5 = \left(\frac{1}{4}\right)84 \; mod \; 5 = \mathbf{1}$$

Ի տարբերություն ինդեքսային մեթոդի, քառակուսիների օրենքի վրա հիմնված մոդուլյար բազմապատկման մեթոդը չի զիջում արագագործության և ապարատային ծախսերի տեսակետից և կիրառվում է բոլոր տեսակի թվերի համար:

Ատենախոսությունում մշակված EPROM-ի միջոցով (նկ.2.4) և քառակուսիների օրենքի հիման վրա (նկ.2.6) մոդուլյար բազմապատկիչները կարելի է իրականացնել, երբ մոդուլի արժեքը ցանկացած տեսակի թիվ է, ինդեքսային մեթոդով մոդուլյար բազմա-պատկիչներն (նկ.2.5) իրացվում են, երբ մոդուլի արժեքը պարզ թիվ է: Այժմ դիտարկենք մոդուլյար բազմապատկիչների այնպիսի տարբերակ, որ մոդուլի արժեքը է 2^k է:

<u>$m=2^k$ մոդուլով մոդուլյար բազմապատկիչների նախագծման մեթոդները</u>

$m=2^k$ մոդուլով մոդուլյար բազմապատկիչների կառուցման համար կարող է օգտագործվել մեթոդ, որը հիմնված է մոդուլյար թվաբանության և դասական բուլյան հանրահաշվի հատկությունների միավորման վրա[8, 13]:

Ենթադրենք տրված են a և b դրական թվերը՝ a>0 և b>0, և դրանց P=a×b արտադրյալը: Եթե երկու թվերը փոքր են մոդուլի 2^k արժեքից, և a, և b թվերը կարող են ներկայացվել n բիթերով (երկու թվերը ունեն միևնույն կարգայնությունը): Այս դեպքում թվերի P արտադրյալը կարող է ներկայացվել հետևյալ բանաձևով՝

$$P = \sum_{i=0}^{2n-1} r_i \cdot 2^i, \qquad (2.7)$$

որտեղ r_i –ն a և b թվերի բազմապատկման i -րդ բիթն է [8, 9]:

(2.7)-ը ընդլայնված տեսքով գրելու դեպքում այն կներակայացնել (2.8) օգնությամբ:

$$P = \sum_{i=0}^{n-1} r_i \cdot 2^i + 2^n \sum_{i=n}^{2n-1} r_i \cdot 2^{i-n} \qquad (2.8)$$

Հաշվարկենք (2.8) արտահայտության արժեքը մոդուլի համար: Հարմարության համար 2^k մոդուլի տեսքը փոխարինենք m-ով: Համապատասխանաբար (2.8) արտահայտությունը կարելի է ներկայացնել հետևյալ տեսքով՝

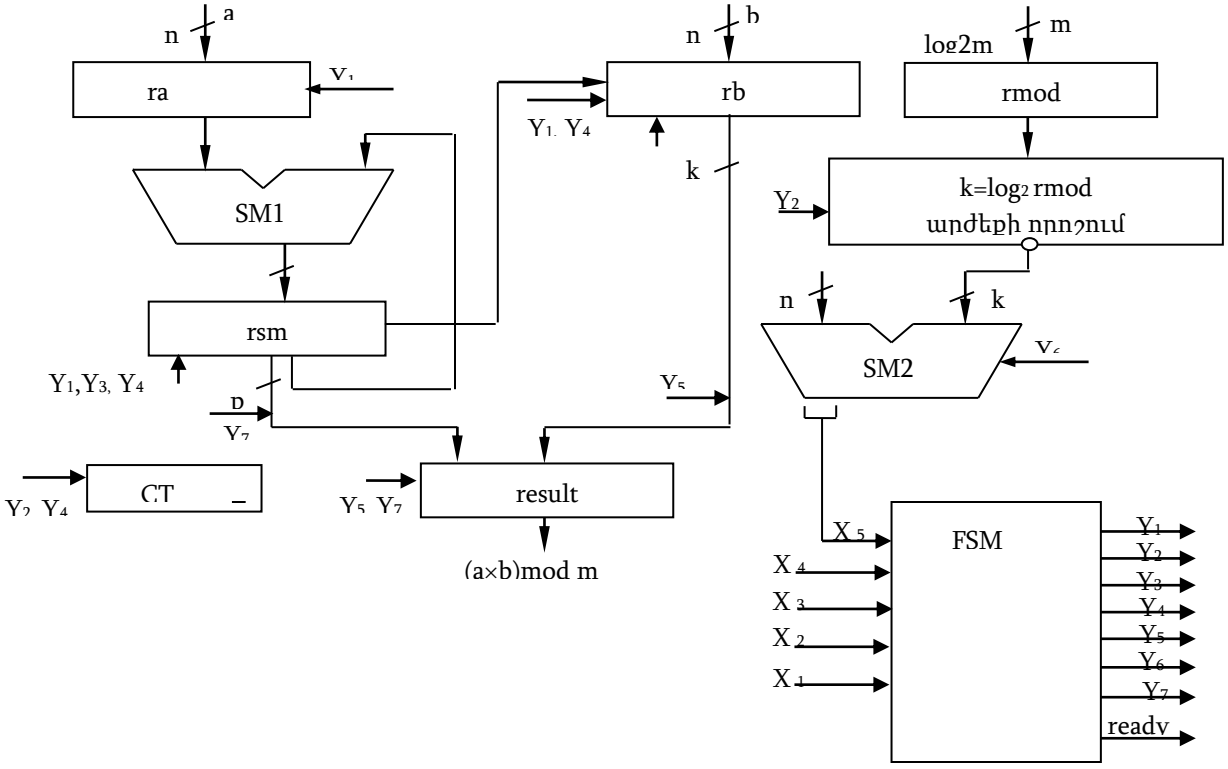
$$|P|_m = \sum_{i=0}^{n-1} r_i \cdot 2^i + |2^n|m \cdot \sum_{i=n}^{2n-1} r_i \cdot 2^{i-n} \mid m \qquad (2.9)$$

Հաշվի առնելով այն փաստը, որ $|2^k|m = 0$ արտահայտություն (2.9)-ը կարելի է ներկայացնել (2.10)-ի տեսքով:

$$|P|_m = \left|\sum_{i=0}^{n-1} r_i \cdot 2^i\right| m \qquad (2.10)$$

Ելնելով այս ամենից՝ m= 2^k մոդուլով մոդուլյար բազմապատկիչը a և b դրական թվերի սովորական երկուական բազմապատկիչ է, որից որպես վերջնական արժեք վերցված են ընդհանուր $P = a \cdot b$ արտադրյալի ցածր k բիթերը:

Այս տարբերակով մոդուլյար բազմապատկման սարքի կառուցվածքային սխեման կունենա նկ. 2.7-ում բերված տեսքը:



Նկ. 2.7. 2^k մոդուլով մոդուլյար բազմապատկիչի կառուցվածքային սխեման

a և b դրական թվերի սովորական երկուական բազմապատկման գործողության ամենատարածված ձևն է բազմապատկիչի փոքր կարգերից սկսած բազմապատկումը, բազմապատկիչի և մասնակի արդյունքների գումարի տեղաշարժումով դեպի աջ: Բազմապատկումը կատարվում է n տակտերում, որտեղ n–ը բազմապատկիչի կարգերի քանակն է: Յուրաքանչյուր տակտում վերլուծվում է բազմապատկիչի փոքր կարգը: Եթե այն հավասար է “1”, ապա մասնակի արտադրյալի գումարին գումարվում է բազմա-

պատկելին: Հակառակ դեպքում գումարվում է “0”, հետո բազմապատկիչը և մասնակի արտադրյալի գումարը տեղաշարժվում է 1 կարգ դեպի աջ, այնուհետև վերլուծվում է բազմապատկիչի փոքր կարգը և այլն: Այս գործողությունը կատարվում է n անգամ:

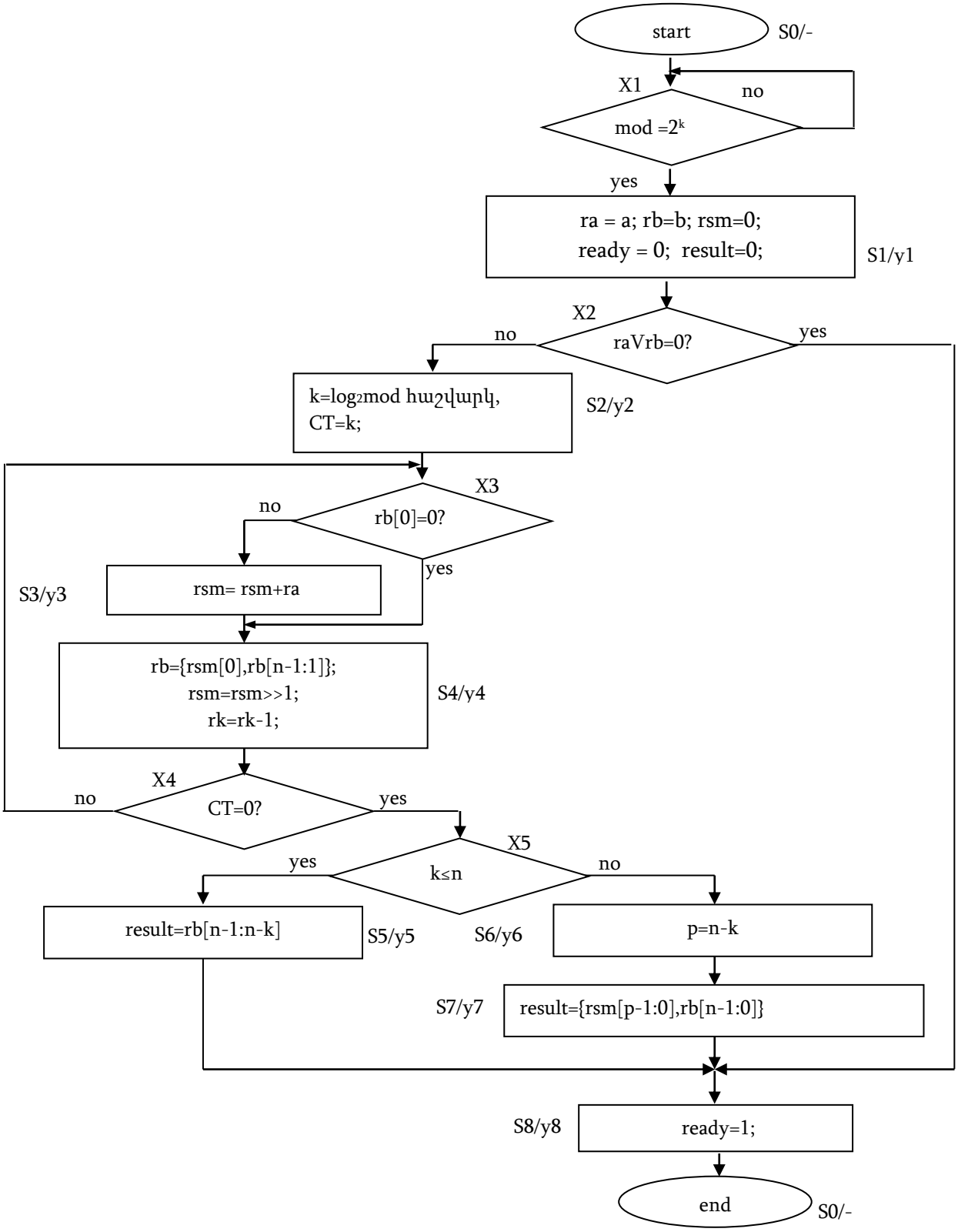
Երկու n կարգանի a և b թվերը տեղակայվում են համապատասխանաբար ra և rb ռեգիստրներում, իսկ մոդուլի արժեքը տեղակայվում է rmod ռեգիստրում:

Երկուական բազմապատկման գործողության իրականացման համար առանձնացվում է նաև rsm ռեգիստրը, որտեղ կգրանցվեն բազմապատկման ընթացքում ձևավորվող միջանկյալ գումարման արդյունքները, ինչպես նաև result-ը վերջնական արդյունքի ռեգիստրն է, որտեղ կգրանցվի $P = (a \cdot b)\ modm$ մոդուլյար բազմապատկման արդյունքը:

Ինչպես երևում է, կառուցվածքային սխեման (նկ. 2.7) համալրված է ղեկավարող ազդանշաններով, որոնք ձևավորվում են կառավարող ավտոմատի կողմից: Այս սարքի ալգորիթմի բլոկ-սխեման ներկայացված է նկ. 2.8-ում:

a կամ b թվերի զրոյին հավասար լինելը ստուգելուց հետո (X2 պայմանը), առանձին ալգորիթմով կատարվում է k արժեքի հաշվարկը, որը տեղակայվում է CT հաշվիչում, քանի որ բազմապատկման գործողությունը կատարվելու է ոչ թե n, այլ k տակտերի ընթացքում՝ քանի որ արտադրյալի վերջին k կարգերը որոշելու համար ըստ բանաձև (2.10-ի):

Եթե $k \leq n$ ապա result ռեգիստրում տեղակայվում են rb ռեգիստրի բարձր k բիթերը (result=rb[n-1:n-k]), հակառակ դեպքում՝ result ռեգիստրում տեղակայվում են rsm ռեգիստրի ցածր $P = n - k$ բիթերը՝ միավորված rb ռեգիստրի n բիթերի հետ (result={rsm[p-1:0],rb[n-1:0]}): p արժեքի հաշվարկը կարելի է կատարել նաև k-ն որոշելուց անմիջապես հետո, սակայն այս տարբերակում p-ի արժեքը որոշվում է անմիջապես result վերջնական արդյունքի ձևավորումից առաջ, և, հնարավոր է p -ի արժեքի որոշման կարիք ընդհանրապես չլինի[6]:



Նկ. 2.8. 2^kմոդուլով մոդուլյար բազմապատկիչի ալգորիթմի բլոկ-սխեման

Համաձայն դասական երկուական առանց նշանի թվերի բազմապատկման ալգորիթմի, արտադրյալի բարձր կարգերը ձևավորվում են rsm ռեգիստրում, իսկ ցածր կարգերը՝ rb ռեգիստրում: Ելնելով այս փաստից՝ $P = (a \cdot b)\ modm$ մոդուլյար բազմապատկման արդյունքի վերջնական արժեքի ձևավորման համար անհրաժեշտ է քննարկել երկու դեպք՝ երբ $k \leq n$, և, երբ $k > n$ (X5 պայմանը):

Դիտարկենք 2^k մոդուլով մոդուլյար բազմապատկման օրինակ, կիրառելով ավանդական երկուական բազմապատկման ալգորիթմը, միակ բացառությամբ, որ եթե k≤n, ապա բազմապատկումը կարելի է կատարել ոչ թե n, այլ k տակտերի ընթացքում:

Օրինակ՝ $a = 67, b = 51, m = 32, k = 5, n = 8: (a \cdot b) mod\ m = (67 \cdot 51) mod\ 32 = 25$:

rsm	rb	CT
000000000	0011001	k= 5
+ 01000011		
001000011		
000100001	1001100	k= 4
+ 01000011		
001100100		
000110010	0100110	k= 3
000011001	0010011	k= 2
000001100	1001001	k= 1
+ 01000011		
001001111		
000100111	1100100	k= 0

$(67 \cdot 51) mod\ 32 = (11001)_2 = 25_{10}$

Համաձայն ալգորիթմի՝ ra 8 բիթանի ռեգիստրում տեղակայվում է 01000011, rb ռեգիստրում՝ 00110011, rmod ռեգիստրում՝ 32, իսկ rsm-ում՝ 000000000: Քանի որ k=5, m=32 մոդուլով երկուական բամապատկումը կարելի է կատարել 5 տակտերի ընթացքում, հետևաբար բազմապատկման տակտերի հաշվիչում գրանցվում է 5:

Կառավարող ավտոմատի կառուցման համար կատարվում է ալգորիթմի բլոկ-սխեմայի Մուրի մոդելի նշանադրում այն նույն սկզբունքով, ինչ ներկայացված է 2.1. բաժնում, ինչի հիման վրա ձևավորում ենք անցումային գրաֆը:

Կարելի է կատարել նաև կառավարող ավտոմատի համար ալգորիթմի բլոկ-սխեմայի Միլի մոդելի նշանադրում: Այս դեպքում, ավտոմատի վիճակների քանակը կնվազի երկուսով, ինչը պայմանավորված է Միլի և Մուրի մոդելների համար բլոկ-սխեմայի նշա-

նադրման առանձնահատկություններով, և այս դեպքում վիճակների քանակի նվազեցումը կրճատում է նաև վիճակների կոդավորման կարգայնությունը:

Կառավարող ազդանշանները նշվում են նկ. 2.7-ում բերված ալգորիթմի բլոկ-սխեմայի օպերացիոն բլոկներին համապատասխան, և ձևավորվում են FSM (Finite State Machine) կառավարող ավտոմատի կողմից: Նկ. 2.7.-ի կառուցվածքային սխեմայում SM1 գումարիչը կատարում է սովորական երկուական գումարում և երկուական բազմապատկման գործողության կատարման ֆունկցիոնալ հանգույցն է, իսկ SM2 գումարիչի միջոցով որոշվում է n և k արժեքների տարբերությունը և հաշվարկվում է p արժեքը (p=n-k): Այնուհետև վերլուծվում է SM2 գումարիչի բարձր բիթը, եթե այն հավասար է զրոյի, ապա կարելի է եզրակացնել, որ k≤n, հակառակ դեպքում՝ k>n: Առանձին ուշադրության է արժանի k արժեքի որոշումը: Քանի որ այն հաշվարկվում է ըստ $k=\log_2$ mod բանաձևի ուստի անհրաժեշտ է հաշվարկել m մոդուլի արժեքի լոգարիթմը: Այս հաշվարկը իրականացնելու համար բավական է rmod տեղաշարժող ռեգիստրի աջ տեղաշարժերի միջոցով որոշել միակ “1” արժեքի դիրքը (կարգը), ինչը և կհանդիսանա $\log_2$ mod արտահայտության արժեքը:

2^k մոդուլով մոդուլյար բազմապատկիչների կառուցման համար կարելի է առաջարկել նաև փոխանցումը պահպանող գումարիչների հիման վրա կառուցված $\frac{k}{2}$ - կարգանի կոմբինացիոն բազմապատկիչների կիրառումը, սակայն, ինչպես նշվել է, որպես ապարատային ծախսեր պահանջվում են $\frac{5k}{2}$ լրիվ գումարիչներ, իսկ փոխանցման ազդանշանի տարածման ժամանակը $\frac{3k}{2}+2(\frac{k}{2}-3)$ է [47] :

Ելնելով վերը նշվածից՝ կարելի է եզրակացնել, որ $m=2^k$ մոդուլով մոդուլյար բազմապատկիչների նախագծման մշակված մեթոդը ավելի արդյունավետ է արագագործության և օգտագործվող գումարիչների կարգայնության տեսանկյունից (k տակտ)՝ ավանդական ըստ mod m-ի բազմապատկման մեթոդի համեմատ, այսինքն՝ սովորական բազմապատկում և հաջորդող բաժանում m-ի մոդուլի: Քանի որ բաժանարարի (մոդուլ m-ի արժեքի) նորմալացման, քանորդի յուրաքանչյուր բիթի հաշվարկման և, ի վերջո, վերջնական մնացորդի, որը և կհանդիսանա (axb) modm արդյունքը, հնարավոր նորմալացման անհրաժեշտ գործընթացները կարող են

արագագործության նվազեցման պատճառ դառնալ: Նշված գործողությունների համար ծախսվում է՝(n+2n-log2m+3) տակտ՝n տակտ բազմապատկման, 2n-log2m+1 տակտ քանորդի կարգերի որոշման և 2 տակտ բաժանարարի և վերջնական մնացորդի նորմալացման համար:

Եթե, նույնիսկ, բաժանման գործողությունն իրականացվիլ արդյունքի ձևավորման փոփոխական ժամանակի եղանակով ապա կրճատվում է միայն բաժանարարի նորմալացման ժամանակը ($n = 32$կարգանի բազմապատկիչների համար մինչև 24, $n = 64$՝ մինչև 56 անգամ) [4]:

Աղ. 2.1-ում ներկայացված են երկուական բազմապատկման և հաջորդող՝ m-ի մոդուլի բաժանման եղանակով (*եղանակ 1*) և առաջարկվող՝ $m=2^k$ մոդուլով մոդուլյար բազմապատկիչների եղանակով (*եղանակ 2*) կառուցված սարքերի աշխատանքի ժամանակի և օգտագործվող գումարիչների վերլուծությունները n=8 և n=16 կարգայնություն ունեցող թվերի դեպքում: Ինչպես երևում է աղյուսակից, $m=2^k$ մոդուլով մոդուլյար բազմապատկիչների եղանակով (*եղանակ 2*) կառուցված սարքերի արագագործությունը զգալիորեն բարձրանում է, իսկ ապարատային ծախսերը (մասնավորապես՝ գումարիչների քանակը և կարգայնությունը) կրճատվում են:

Աղյուսակ 2.1. Երկուական բազմապատկման և հաջորդող՝ m-ի մոդուլի վրա բաժանման ու $m=2^k$ մոդուլով մոդուլյար բազմապատկիչների կիրառման եղանակների համեմատում

Մոդուլի արժեք	n=8				n=16			
	եղանակ1		եղանակ 2		եղանակ 1		եղանակ 2	
	Տակտ քանակ	Գումարիչ քանակ	Տակտ. քանակ	Գումարիչ. քանակ	Տակտ. քանակ	Գումարիչ քանակ	Տակտ. քանակ	Գումարիչ քանակ
m=16 k=4	23	1x8 և 1x16 կարգ. SM	4	2x8 կարգ. SM	47	1x16 և 1x32 կարգ. SM	4	2x16 կարգ. SM
m=64 k=6	18	1x8 և 1x16 կարգ. SM	6	2x8 կարգ. SM	45	1x16 և 1x32 կարգ. SM	6	2x16 կարգ. SM
m=1024 k=10	13	1x8 և 1x16 կարգ. SM	8	2x8 կարգ. SM	41	1x16 և 1x32 կարգ. SM	10	2x16 կարգ. SM

Նշենք, որ $m = 2^k$ մոդուլով մոդուլյար բազմապատկիչները կարող են կիրառվել նաև $m = 2^k + 1$ և $m = 2^k - 1$ մոդուլներով կոդերի հսկում իրականացնող մասնագիտացված սարքերում, իսկ այն մասնավոր դեպքերում, երբ $m = 2^k + 1$ և $m = 2^k - 1$ թվերը պարզ թվեր են (օրինակ՝ $m = 2^4 + 1$=17, $m = 2^8 + 1 = 257, m = 2^5 - 1 = 31, m = 2^7 - 1 = 127$ և

այլն), ապա նմանատիպ բազմապատկիչները կարող են կիրառվել նաև որոշ ալգորիթմներով (RSA գաղտնագրման ալգորիթմ, Դիֆֆի-Հելլմանի գաղտնագրման բանալիների գեներացման ալգորիթմ) գաղտնագրման սարքերի կառուցման դեպքում[5, 97]:

Ինչպես արդեն նշվել է, մոդուլյար բազմապատկիչների իրականացումը ինդեքսային մեթոդով հնարավոր է, եթե մոդուլի արժեքը պարզ թիվ է, քանի որ պարզունակ արմատի որոշումը բանաձև (1.8)-ի հիման վրա անհնար է: Սակայն 2^k մոդուլով մոդուլյար բազմապատկման սարքի իրականացումը նույնպես հնարավոր է ինդեքսային մեթոդով, եթե $k < 3$: Դիտարկենք 2^k մոդուլով մոդուլյար բազմապատկման համար թվի մոդուլյար ներկայացումից ինդեքսային ներկայացման իրականացումը [8, 9, 13, 35]:

Ցանկացած k>3 պայմանին բավարարող k թվի համար $\pm5^1$, $\pm5^2$, $\pm5^2$,...., $\pm5^{k-2}$ ամբողջ թվերը հանդիսանում են 2^k մոդուլով բոլոր մնացորդները: Հետևաբար, ցանկացած a կենտ թիվ $A \in [1,\ 2^k - 1]$ միջակայքից կարելի է ներկայացնել մոդուլյար ինդեքսային ձևով, որպես (β,γ) ինդեքսների զույգ (2.11)-ի տեսքով:

$$a = |5^{\beta}\ (-1^{\gamma})|_{2^k}\ , \tag{2.11}$$

որտեղ՝ $\beta \in \{0,1, \dots, (2^{k-2} - 1)\}$ և $\gamma \in \{0,1\}$ միջակայքերին:

Այսպիսով, A թիվը կարելի է ներկայացնել (2.11)-ի տեսքով:

$$a = 2^{\alpha}|5^{\beta}\ (-1^{\gamma})|_{2^k} \tag{2.12}$$

որտեղ α գործակիցը $\alpha \in \{0,1, \dots, k-1\}$ միջակայքից է:

Փաստորեն յուրաքանչյուր ամբողջ a թիվը $\alpha \in [1, 2^{k-1}]$ միջակայքից կարելի է ներկայացնել(α, β, γ) ինդեքսների եռյակի միջոցով: Երկու թվի մոդուլյար բազմապատկումը կարելի է իրականացնել հետևյալ օրենքի հիման վրա: Եթե A և B թվերը բավարարում են a≠0, b≠0, և $a \in [1,2, \dots, 2^k - 1]$, ապա յուրաքանչյուր թիվ ներկայացվում է (α, β, γ) ինդեքսների եռյակի միջոցով, որոնք ներկայացված են (2.13) և (2.14) տեսքով:

$$a = 2^{\alpha_1}|5^{\beta_1}\ (-1^{\gamma_1})|_{2^k} \tag{2.13}$$

$$b = 2^{\alpha_2}|5^{\beta_2}\ (-1^{\gamma_2})|_{2^k} \tag{2.14}$$

Հիմնվելով (2.13) և (2.14)-ի վրա երկու թվի բազմապատկումը կարելի ներկայացնել բանաձև (2.15)-ի միջոցով՝

$$|a \cdot b|_{2^k} = 2^{\alpha_1+\alpha_2} |5^{\beta_1+\beta_2} (-1^{\gamma_1+\gamma_2})|_{2^k} : \quad (2.15)$$

Գործողությունը իրականացվում է համաձայն հետևյալ օրենքի՝

- α_1 և α_2 ինդեքսների գումարումը սովորական երկուական գումարում է: Եթե α_1 և α_2 ինդեքսների գումարը հավասար է k-1, համապատասխան β և γ ինդեքսները հավասարեցվում են զրոյի: Եթե α_1 և α_2 ինդեքսների գումարը մեծ է k-1 արժեքից, բազմապատկման արդյունքը հավասար է զրոյի:
- β1 և β2 ինդեքսները գումարվում են 2^{k-2} մոդուլով:
- γ_1 և γ_2 գումարվում են 2-ի մոդուլով գումարիչի միջոցով:

Այս մոտեցման հիմնական թերությունն այն է, որ $[1, 2^k - 1]$միջակայքի թվերը կարելի է ներկայացնել ինդեքսների մի քանի եռյակների միջոցով [1, 2, 7]: Օրինակ, եթե $k = 3$, ապա բոլոր յոթ թվերի համար պահանջվում է ինդեքսների 23 տարբեր եռյակներ (3 թվի ներկայացման համար կարելի է օգտագործել 4 եռյակ՝ (2,0,0),(2,0,1),(2,1,0),(2,1,1)): Եթե մոդուլի արժեքը՝ $m = 2^4$, ապա [1, 2, 3...15] միջակայքից թվերի համար պահանջվում է ինդեքսների 32 եռյակ: Մոդուլի արժեքի աճին զուգընթաց այս եռյակների ավելցուկները հանգեցնում են մեծածավալ LUT աղյուսակների կիրառման անհրաժեշտությանը, այսինքն՝ աճում են ապարատային ծախսերը: Ինդեքսների եռյակների քանակի նվազեցմանը կարելի է հասնել, 2.12 արտահայտությունը ձևափոխելու դեպքում, որից հետո յուրաքանչյուր թիվ կարելի է ներկայացնել (2.16.)-ի տեսքով՝

$$a = |2^{\alpha} \cdot 5^{\beta} (-1^{\gamma})|_{2^k} \quad (2.16)$$

Այս դեպքում ինդեքսների փոփոխման միջակայքը հաշվարկվում է համաձայն հետևյալ օրենքների.

- եթե $\alpha_1 + \alpha_2 \geq k$, ապա բազմապատկման արդյունքը հավասար է զրոյի,

  եթե $\alpha_1 + \alpha_2 = k$, ապա բազմապատկման արդյունքը $2^{k-!}$ է:
- β1 և β2 ինդեքսները գումարվում են $2^{k-2-\alpha}$ մոդուլով:
- γ_1 և γ_2 գումարվում են 2-ի մոդուլով գումարիչի միջոցով:

Աղ. 2.2-ում ներկայացված են $m=16=2^4$ մոդուլով (2.12)՝ համաձայն արտահայտության, բոլոր չկրկնվող ինդեքսների եռյակները, որոնք դիտարկվում են որպես մոդիֆիկացված ինդեքսային ձևափոխում $m=16$ մոդուլով բազմապատկման համար:

Աղյուսակ 2.2. m=16 մոդուլի բոլոր չկրկնվող ինդեքսների ներկայացումը

$a=\|2^{\alpha}5^{\beta}(-1^{\gamma})\|_{16}$	α	β	γ
1	0	0	0
2	1	0	0
3	2	1	0
4	2	0	0
5	0	1	0
6	1	1	1
7	0	2	1
8	3	0	0
9	0	2	0
10	1	1	0
11	0	1	1
12	2	0	1
13	2	1	1
14	1	0	1
15	0	0	1

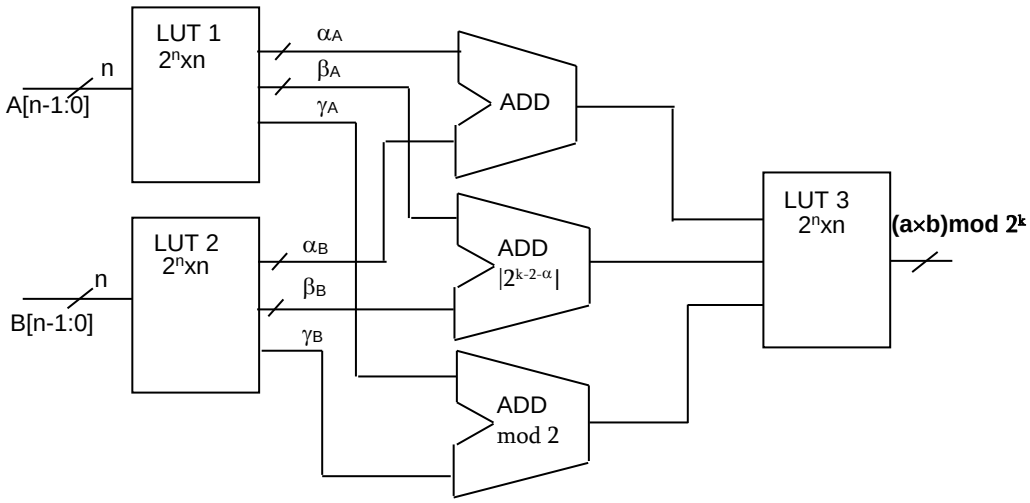
Քանի որ աղ. 2.2-ը պարունակում է բոլոր չկրկնվող ինդեքսների եռյակները, նրանց ավելցուկների խնդիրը վերանում է: Մի քանի տարբերակի հիման վրա հիմնավորենք աղյուսակում ներկայացված ինդեքսային եռյակները:

$(2^0 5^0(-1^0))\bmod 16=1 \bmod 16=1$, $(2^0 5^1(-1^0))\bmod 16=5 \bmod 16=5$,

$(2^1 5^1(-1^1))\bmod 16=-10 \bmod 16=6$, $(2^3 5^0(-1^0))\bmod 16=8 \bmod 16=8$ և այլն:

LUT աղյուսակների կրառմամբ 2^k մոդուլով մոդուլյար բազմապատկման սարքի կառուցվածքային սխեմանբերված է նկ. 2.9-ում:

Բեռնելով A և B թվերի ինդեքսների արժեքները համապատասխանաբար եռամուտք LUT1 և LUT2 աղյուսակներում՝ կատարում ենք ինդեքսների գումարում երեք գումարիչների միջոցով, որոնցից մեկը դիրքային է (α_1 և α_2 ինդեքսների գումարիչը), իսկ մյուս երկուսը՝ մոդուլյար ($\beta 1$ և $\beta 2$, γ_1 և γ_2 ինդեքսների գումարման համար): Հակադարձ ինդեքսային ձևափոխման LUT աղյուսակն ունի երեք մուտք:



Նկ. 2.9. 2^k մոդուլով ինդեքսային մեթոդով մոդուլյար բազմապատկիչի կառուցվածքային սխեման

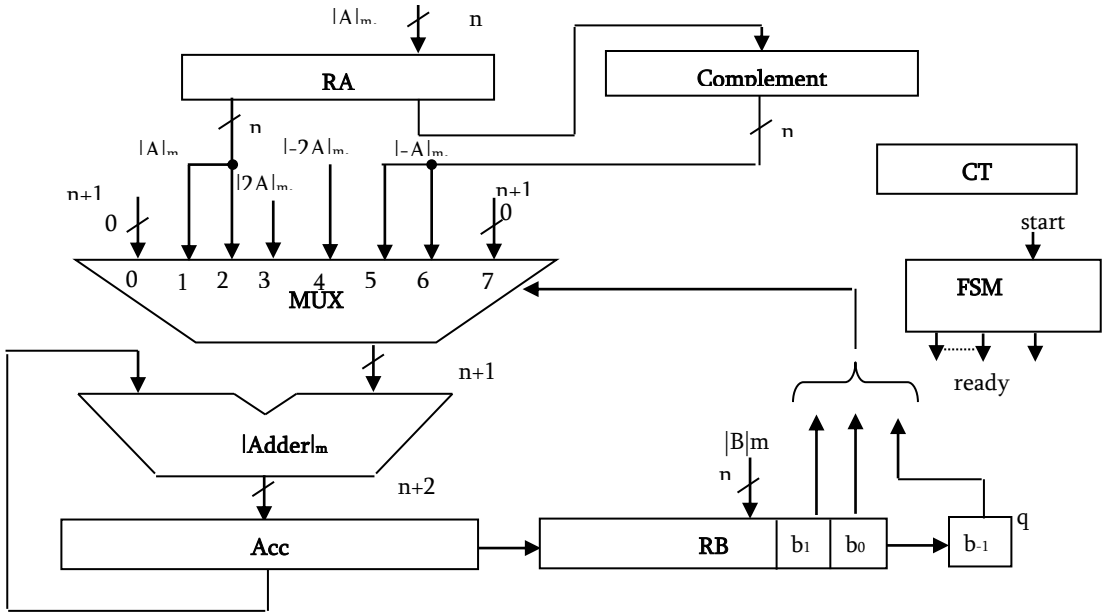
2^k մոդուլով մոդուլյար բազմապատկման սարքի ինդեքսային մոդուլյար բազմապատկիչների ապարատային իրագործման ժամանակ անհրաժեշտ է դիտարկել նաև այն դեպքը, երբ օպերանդներից (արտադրիչներից) մեկը (կամ երկու օպերանդները միաժամանակ) հավասար է զրոյի: Ինչպես երևում է աղյ. 2.2-ից, օպերանդի զրոյի հավասար լինելը LUT1 և LUT2 աղյուսակներում արտապատկերված չէ: Այս դեպքը պահանջում է լրացուցիչ սխեմա մուտքային օպերանդները “0”-ի հետ համեմատելու համար:

Առաջարկված մոդուլյար բազմապատկիչի նախագծման այս մեթոդը մասամբ հիմնված է գլուխ 1-ում դիտարկված ինդեքսային մեթոդով մոդուլյար բազմապատկման մեթոդի վրա, սակայն այս դեպքում մոդուլի արժեքը ոչ թե պարզ թիվ է, այլ՝ 2^k թիվ:

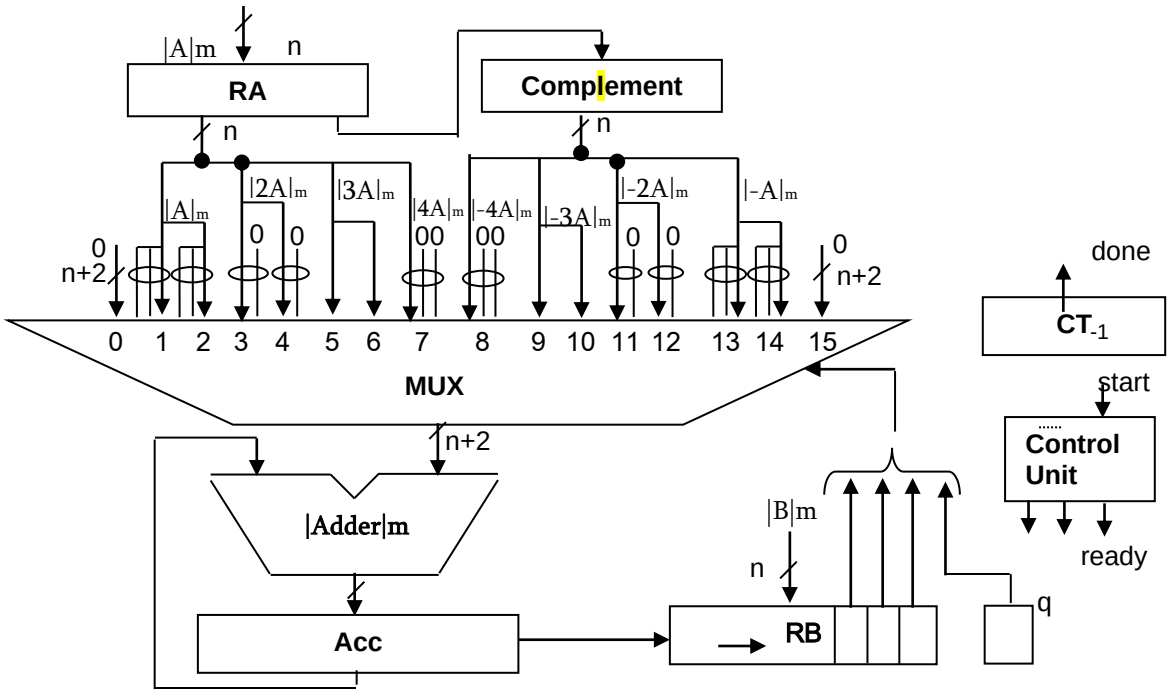
<u>Բուտի ալգորիթմի հիման վրա մոդուլյար բազմապատկիչների նախագծում</u>

Գլուխ 1-ում հետազոտվել է Բուտի ալգորիթմի հիման վրա մոդուլյար բազմապատկիչների կառուցման առկա մեթոդը: Ատենախոսությունում առաջարկվում է Բուտի ալգորիթմի հիման վրա մոդուլյար բազմապատկիչների նախագծման մեկ այլ մեթոդ, երբ մասնակի արտադրյալները որոշվում են ոչ թե կոմբինացիոն կամ ծառանման բազմապատկիչների (կախված թվերի կարգայնությունից) միջոցով, այլ Բուտի վերակոդավորման գործողություններից հետո մասնակի արտադրյալների գումարը իրացվում է մոդուլյար գումարիչների միջոցով: Բազմապատկումը կատարվում է n տակտերի ընթացքում, որտեղ n-ը բազմապատկիչի կարգերի քանակն է [44, 74,91]: Բուտի մոդիֆիկացված ալգորիթմով (երեք և չորս բիթերի վերլուծությամբ) մոդուլյար բազմա-

պատկման սարքերի կառուցվածքային սխեմաները պատկերված են նկ. 2.10-ում և նկ. 2.11-ում:



Նկ. 2.10. Բուտի մոդիֆիկացված ալգորիթմով (երեք բիթի վերլուծությամբ) մոդուլյար բազմապատկման սարքի կառուցվածքային սխեմա



Նկ. 2.11. Բուտի մոդիֆիկացված ալգորիթմով (երեք բիթի վերլուծությամբ) մոդուլյար բազմապատկման սարքի կառուցվածքային սխեմա

Բազմապատկման յուրաքանչյուր տակտում վերլուծվում է բազմապատկիչի երկուական բիթ՝ ընթացիկը և նախորդը: Կատարվում է վերակոդավորում (Booth

recoding) երկուական կոդից 3-ի հիմքով կոդի՝ (0,1) = (-1, 0,1): Ելնելով վերը նշվածից, կարելի է պնդել, որ մոդուլյար բազմապատկիչների իրականացման համար կարելի է կիրառել Բուտի ալգորիթմը, որի համար կպահանջվի հաշվարկել բազմապատկելին իր գործակիցներով ըստ մոդուլ m-ի, օրինակ՝

$$|-A|_m,\ |-2A|_m,\ |\pm 3A|_m,\ |\pm 5A|_m,\ |\pm 7A|_m \text{ և այլն:}$$

Այսպիսի մասնակի արտադրյալների ձևավորումը զգալիորեն բարդացնում է բազմապատկման սարքի կառուցվածքը, ինչը ստիպում է ընտրել վերակոդավորման այն տարբերակը, երբ մասնակի արտադրյալների հաշվարկման ապարատային ծախսերը կարդարացվեն: Եթե դիտարկենք Բուտի տարբեր մոդիֆիկացումներով նախագծված սարքերը՝ Radix4, Radix 8, Radix 16 և Radix 32 արագագործության և ապարատային ծախսերի տեսակետից, կստանանք հետևյալ արդյունքները (աղ. 2.3).

Աղյուսակ 2.3. Բուտի մոդիֆիկացումներով նախագծված սարքերի արագագործությունը և ապարատային ծախսերը

	Արագագործություն	Ապարատային ծախսեր		
Radix 4	n տակտ	1 հատ 4_1MX	n կարգ. հաշվիչ	n+1 կարգ. գումարիչ
Radix 8	n/2 տակտ	1 հատ 8_1MX	n/2 կարգ. հաշվիչ	n+2 կարգ. գումարիչ
Radix 16	n/3 տակտ	1 հատ 16_1MX	n/3 կարգ. հաշվիչ	n+3 կարգ. գումարիչ
Radix 32	n/4 տակտ	1 հատ 32_1MX	n/4 կարգ. հաշվիչ	n+4 կարգ. գումարիչ

Եթե բազմապատկման գործողությունը իրականացնենք նույն ալգորիթմներով հաջորդող բաժանման գործողությամբ, և ոչ թե մնացորդային դասերի համակարգով, ապա բազմապատկումից (2n տակտ միայն բազմապատկման համար) հետո բաժանման գործողությունը կատարվում է.

- 2n-m տակտ՝ նորմալիզացման համար,
- 2n-m+1 տակտ՝քանորդի կարգերի որոշման համար,
- 2n-m տակտ՝ վերջնական մնացորդի տեղաշարժի համար,

հետևաբար զուտ բաժանման գործողության վրա ծախսվում է 3n-3m +1 ժամանակ: Արդյունքները ներկայացված են աղ. 2.4-ում՝

Աղյուսակ 2.4. Բուտի մոդիֆիկացված ալգորիթմների մոդուլյար և երկուական թվաբանություններում սարքերի արագագործությունը

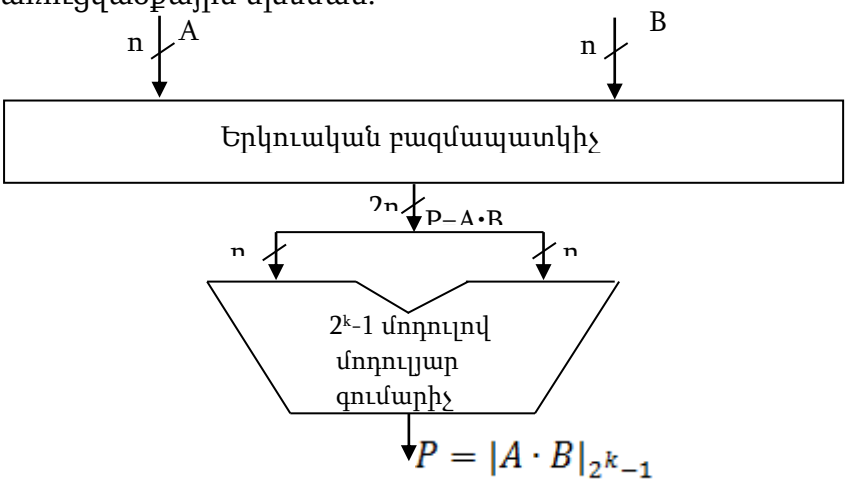
	Բուտի ալգորիթմի մոդիֆիկացումները			
	Radix 4	Radix 8	Radix 16	Radix 32
Մոդուլյար բլոկում Բուտի ալգորիթմով նախագծված բազմապատկիչներ	n տակտ	n/2 տակտ	n/3 տակտ	n/4 տակտ
Բուտի ալգորիթմով բազմապատկում և հաջորդող բաժանում	2n +6n-3m +1	n/2+6n-3m +1	n/3+6n-3m +1	n/4+6n-3m +1

Դիտարկենք 2^k-1 մոդուլով բազմապատկիչների իրացման առանձնահատկությունները: Այս մոդուլով բազմապատկման հնարավոր տարբերակներից է ինդեքսային մոդուլյար բազմապատկումը: Այդպիսի մոդուլյար բազմապատկիչների կառուցման սկզբունքները ներկայացված են տվյալ աշխատանքի ինդեքսային մոդուլյար բազմապատկիչների նախագծման բաժնում: Սակայն, ինչպես արդեն նշվել է, ինդեքսային մեթոդով մոդուլային բազմապատկիչների կառուցումը հնարավոր է, միայն, եթե մոդուլի արժեքը պարզ թիվ է (օրինակ՝ $m=2^3-1=7$, $m=2^5-1=31$, $m=2^7-1=127$), իսկ 2^k-1տիպի ոչ բոլոր մոդուլներն են պարզ թվեր (օրինակ՝ $m=2^4-1=15$, $m=2^6-1=63$, $m=2^8-1=255$): Բացի դրանից բազմապատկելիների կարգայնության մեծացմամբ զգալիորեն ավելանում են ապարատային ծախսերը:

2^k-1 մոդուլով բազմապատկիչները կարելի է կառուցել նաև սովորական դիրքային բազմապատկիչի և մոդուլյար գումարիչի հիման վրա: Դիտարկենք այս սկզբունքով կառուցված մոդուլյար բազմապատկիչները: Օրինակ, ունենք երկու դրական ամբողջ թիվ. $A\geq 0$ և $B\geq 0$, դրանց արտադրյալը՝ $P=AxB$: Ենթադրենք՝ A և B թվերը գտնվում են $[0, 2^k-1]$ միջակայքում, (այսինքն դրանց արժեքները փոքր են 2^k-ից) և կարող են ներկայացվել n=k բիթերի միջոցով: P արտադրյալը կարող է ներկայացվել (2.17) արտահայտության միջոցով և բայնուհետ ձևափոխվել (2.18) բանաձևի: Վերահաշվարկենք այդ արտահայտությունը $m=2^k-1$ մոդուլի համար և հաշվի առնելով, որ $|2^k|_{2^k-1} = 2^k - (2^k - 1) = 1$ կստանանք բանաձև (2.17.)-ը [33, 80]:

$$|P|_{2^k-1} = |\sum_{i=0}^{n-1} r_i \cdot 2^i + \sum_{i=n}^{2n-1} r_i \cdot 2^{i-1}|_{2^k-1} \qquad (2.17)$$

Հիմնվելով բանաձև (2.14)-ի վրա և կիրառելով դիրքային բազմապատկիչի և 2^k-1 մոդուլով մոդուլյար գումարիչի սկզբունքը՝ նկ. 2.12-ում ներկայացվում է 2^k-1 մոդուլով բազմապատկիչի կառուցվածքային սխեման:



Նկ. 2.12. Դիրքային բազմապատկիչի և մոդուլյար գումարիչի հիման վրա 2^k-1 մոդուլով մոդուլյար բազմապատկման սարքի կառուցվածքային սխեմա

Այս մշակված մեթոդի առավելությունը իրականացման պարզությունն է, սակայն բազմապատկիչի հիմնական բնութագրերը (արագագործություն, զբաղեցնող մակերես և այլն) կախված են դիրքային բազմապատկիչի և մոդուլյար գումարիչի որոշակի իրացումից: Նշված բնութագրերի լավարկումը կարելի է կազմակերպել, որպես դիրքային բազմապատկիչ կիրառելով Բուտի ալգորիթմի մոդիֆիկացումներից մեկը, ինչպես նաև բազմապատկիչներում օգտագործել Ուոլլեսի ծառանման գումարիչներ, և որպես մուդուլային գումարիչ կիրառել BDD տեխնոլոգիաների վրա հիմնված արագագործ մոդուլյար բազմապատկիչներ (այս մոդուլյար գումարիչների կառուցման սկզբունքները ներկայացված են տվյալ աշխատանքի 2.1. բաժնում):

Ուոլլեսի ծառանման բազմապատկիչը [44] հնարավորություն է տալիս արդյունավետ իրականացնել 2^k-1 մոդուլով մասնակի արտադրյալների ձևավորումը: Այս դեպքում $PP_1, \dots, PP_k$ մասնակի արտադրյալները և P վերջնական արտադրյալը կարելի է հաշվարկել 2^k-1 մոդուլով: PP_i-ի արժեքը հաշվարկվում է (2.16) օգնությամբ [8, 44]

$$PP_i = a_i \, . b_{n-i-1} \dots . b_0 b_{n-1} \dots . b_{n-i} \qquad (2.16)$$

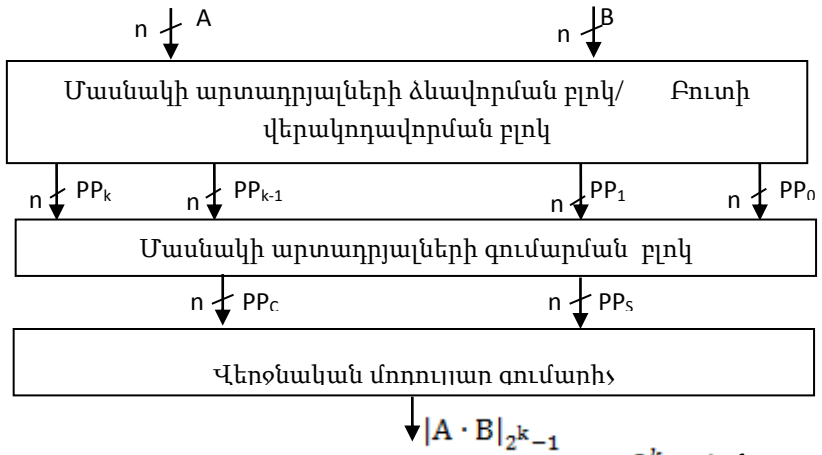
և 2^k-1մոդուլով i -րդ մասնակի արտադրյալն է: Այսպիսով 2^k-1 մոդուլով վերջնական արտադրյալը կարելի է դիտարկել որպես PP_i մասնակի արտադրյալների գումար և ներկայացնել (2.17) բանաձևով [77, 80]:

$$|A \cdot B|_{2^k-1} = |\sum_{i=0}^{n-1} PP_i|_{2^k-1} \qquad (2.17)$$

Կարևոր է նշել, որ ի տարբերություն սովորական դիրքային երկուական գումարիչի, որտեղ մասնակի արտադրյալների կարգայնությունը հավասար է $n+1$ մոդուլյար բազմապատկիչի մասնակի արտադրյալների կարգայնությունը հավասար է n-ի: Ընդ որում, եթե այս նույն մոտեցման համար կիրառենք բազմապատկման Բուտի մոդիֆիկացված ալգորիթմը (երեք բիթ վերլուծությամբ), ապա (2.17) բանաձևը կարելի ներկայացնել հետևյալ կերպ՝

$$|A \cdot B|_{2^k-1} = |\sum_{i=0}^{n/2} PP_i|_{2^k-1} \qquad (2.18)$$

Ելնելով (2.18)-ից՝ կարելի է առաջարկել Բուտի ալգորիթմի կիրառմամբ $m = 2^k - 1$մոդուլով բազմապատկման սարքի կառուցվածքը, որը ներկայացված է նկ. 2.13-ում և բաղկացած է մասնակի արտադրյալների ձևավորման բլոկից (գեներատորից), որն այլ կերպ կարելի է անվանել Բուտի վերակոդավորման բլոկ, և այդ մասնակի արտադրյալների գումարող բլոկից, որը իր մեջ ներառում է Ուոլլեսի ծառը և վերջնական մոդուլյար գումարիչ:



Նկ. 2.13. Բուտի ալգորիթմի և Ուոլլեսի ծառի կիրառմամբ $m = 2^k - 1$ մոդուլով բազմապատկման սարքի կառուցվածքային սխեմա

Այստեղ PP_c-ը փոխանցումների մասնակի արտադրյալներն են, և PP_s-ը արտադրյալի տվյալ կարգը ձևավորող մասնակի արտադրյալները: $m = 2^k + 1$մոդուլով բազմապատկիչները նույնպես ունեն իրականացման առանձնահատկություններ և կիրառվում են հատուկ ալգորիթմներում՝ թվային հսկում, գաղտնագրում և այլն:

$m = 2^k + 1$ մոդուլով բազմապատկիչների իրականացման եղանակներից են՝

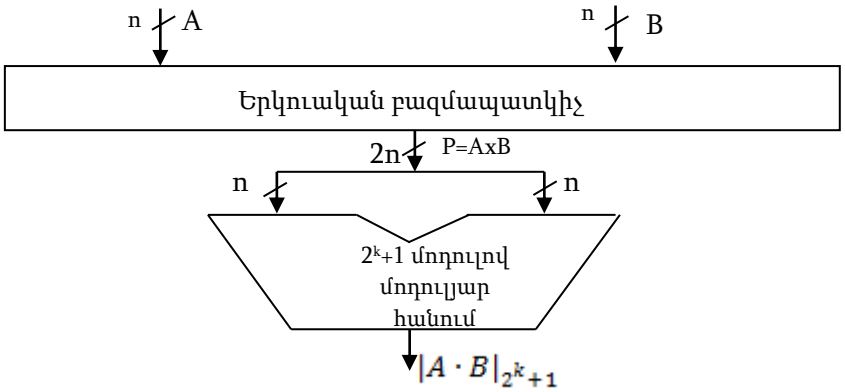
- LUT աղյուսակների կիրառմամբ:
- Դիրքային բազմապատկիչների կիրառմամբ:

LUT աղյուսակների կիրառման եղանակը հիմնված է ինդեքսային մոդուլյար բազմապատկման վրա, սակայն այն գործում է , եթե մոդուլի արժեքը պարզ թիվ է, իսկ 2^k+1 մոդուլին պատկանում են միայն երկու թիվ՝ $m=2^4+1=17$ և $m=2^8+1=257$:

Դիտարկենք դիրքային բազմապատկիչների կիրառման եղանակը՝ հիմնված (2.19)-ի վրա, հաշվի առնելով այն փաստը, որ $|2^k|_{2^k} + 2^k - (2^k + 1) = -1$ [74, 77]:

$$|P|_{2^k+1} = \left| \sum_{i=0}^{n-1} r_i \cdot 2^i - \sum_{i=n}^{2n-1} r_i \cdot 2^{i-n} \right|_{2^k+1} \qquad (2.19)$$

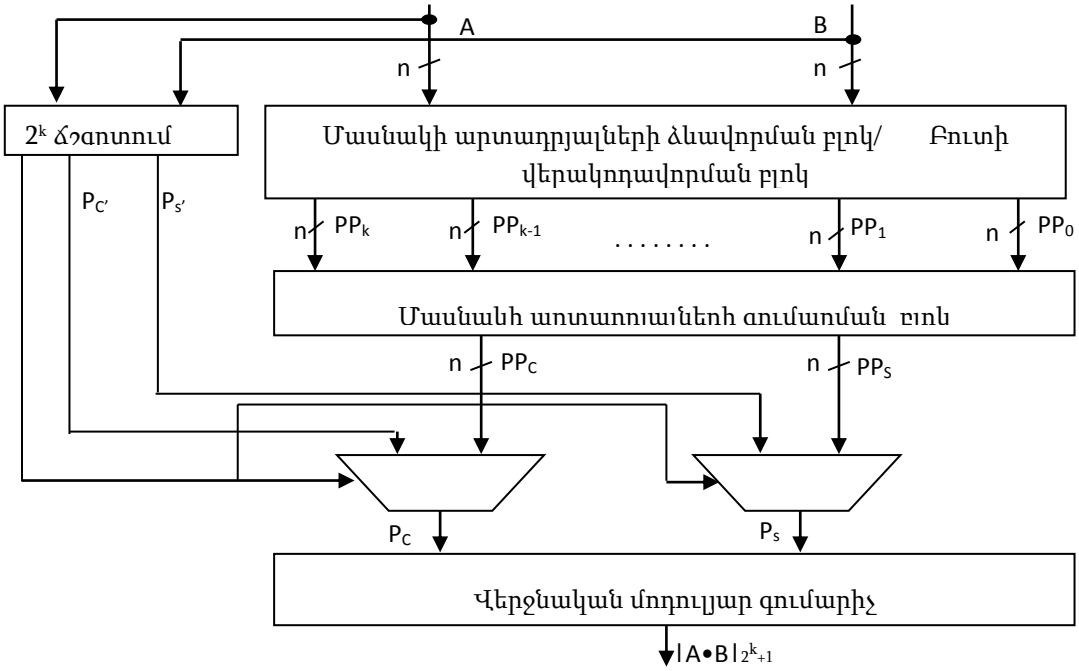
Բազմապատկիչի կառուցվածքը (նկ. 2.14) կազմված է երկու սխեմայից՝ դիրքային բազմապատկիչից և $m = 2^k + 1$մոդուլով մոդուլյար հանող սարքից:



Նկ. 2.14. Դիրքային բազմապատկիչի և մոդուլյար հանող սարքի հիման վրա $m = 2^k + 1$ մոդուլով մոդուլյար բազմապատկման սարքի կառուցվածքային սխեմա

Հետևաբար, Բուտի մոդիֆիկացված ալգորիթմով մոդուլյար բազմապատկիչը (նկ. 2.15.) կազմված կլինի մասնակի արտադրյալների ձևավորման բլոկից (Բուտի վերակոդավորման բլոկ), մասնակի արտադրյալների գումարիչից, արդյունքի ճշգրտման բլոկից, այն դեպքի համար, երբ բազմապատկիչները կամ դրանցից մեկը հավասար լինի

զրոյի, երկու մուլտիպլեքսորներից, որոնք կստուգեն կա արդյոք ճշգրտման անհրաժեշտություն և վերջնական մոդուլյար գումարիչից:



Նկ. 2.15. Բուտի ալգորիթմի և Ուոլլեսի ծառի կիրառմամբ $m = 2^k + 1$մոդուլով բազմապատկման սարքի կառուցվածքային սխեմա

Հիմնվելով աղյուսակներ 2.1-ի և 2.4-ի տվյալների, բանաձևեր (2.10), (2.15), (2.20) և (2.121) վրա կիրառման արդյունքում մշակված կառուցվածքային սխեմաների (նկ. 2.7, 2.9, 2.10, 2.11, 2.13, 2.15) մոդուլյար բազմապատկիչների նախագծման առաջարկված որոշ մեթոդներում, ի տարբերություն առկա նախագծման մեթոդների, կառավարող ավտոմատը ներկայացված է FSM-ի միջոցով, ինչը զգալիորեն նվազեցնում է FPGA ռեսուրսների օգտագործումը (հետազոտման արդյունքները ներկայացված են գլուխ 3-ում): Բացի դրանից հատուկ նախագծված են $m = 2^k$ մոդուլով մոդուլյար բազմապատկիչներ FSM-ի օգտագործմամբ և, ինչը կարևոր է, $m = 2^k$ մոդուլով մոդուլյար բազմապատկիչներ՝ կառուցված առկա նախագծման ինդեքսային մեթոդի հիման վրա:

2.3. Մոդուլով աստիճան բարձրացնելու սարքի նախագծման մեթոդները

Ինչպես արդեն նշվել է, թվային սարքերին ներկայացվող արագագործության բարձրացնելու անհրաժեշտությունը հարկադրում է օգտագործել մնացորդային դասերի համակարգը և մոդուլյար հաշվարկները, ինչը հնարավորություն է տալիս էականորեն բար-

ձրացնել թվաբանական գործողությունների կատարման արագագործությունը, մնացորդների հետ կատարվող գործողությունների զուգահեռաբար կատարման շնորհիվ:

m մոդուլով աստիճան բարձրացնելու գործողության իրականացման սարքի նախագծման անհրաժեշտությունը թելադրված է բազմաթիվ կիրառական ալգորիթմներում այս գործողության կատարմամբ:

Եթե $a^b mod\ m$ արտահայտությունը ավանդական սկզբունքով հաշվարկելու դեպքում հետազոտական խնդիրներում b -ի արժեքը կարող է այնքան մեծ լինել, որ a^b արժեքը պահելու համար պահանջվի շատ մեծ հիշողություն: Հաշվարկել $a^b mod\ m$ արտահայտությունը կարելի է ավելի արագ, եթե այն ներկայացվի (2.20)-ի տեսքով [8, 9, 13]:

$$a^b \bmod m = a_1^b \bmod m \qquad (2.20)$$

որտեղ a_1–ը a -ն m-ի բաժանման մնացորդն է, այսինքն՝ մի թիվ, որն ավելի փոքր է, քան m մոդուլի արժեքը: Օրինակ.

$$a^{769} = a^{7x10^2+6x10^1+9} = a^9 \cdot (a^6)^{10} \cdot ((a^7)^{10})^{10} = a^9(a^6(a^7)^{10})^{10},$$

այսինքն կրճատվում է բազմապատկումների քանակը: Իսկ եթե մեզ հետաքրքրում է մնացորդը m-ի այս թվի բաժանումից, ապա պատք է սկսել “ներքին փակագծից”: Որոշում ենք մնացորդը a^7 թիվը m-ի վրա բաժանելիս և նշանակում այն a_1, ապա որոշում ենք $a_1^{10} mod\ m$ և ստանում a_2 և այլն, ամեն անգամ ստանալով m^{10} թվից ավելի փոքր թիվ:

Ընդհանուր դեպքի համար, եթե b -ն ներկայացնենք բանաձև (2.21)-ով

$$b = b_{n-1}b_{n-2}\dots b_1b_0 = b_{n-1} \cdot 10^{n-1} + b_{n-2} \cdot 10^{n-2} + \dots + b_1 \cdot 10^1 + b_0 \cdot 10^0 \qquad (2.21)$$

$a^b mod\ m$ արժեքի հաշվարկումը կարելի է ներկայացնել բանաձև (2.22)-ով `

$$a^b = a \cdot b_0(a \cdot b_1 \cdot (a \cdot b_2 \cdot \dots (a \cdot b_{n-3} \cdot (a \cdot b_{n-2} \cdot (a \cdot b_{n-1})^{10})^{10})^{10}) \dots^{10})^{10} \qquad (2.22)$$

Փաստորեն, այս գործողությունը իրագործվում է ծրագրային եղանակով, ընդ որում այս խնդրի ամենահավանական լուծումը՝ a թիվն ինքն իրենով b անգամ բազմապատկման գործողություն կատարելն է: Կրճատել բազմապատկումների քանակը, այսինքն՝ արագացնել գործողության կատարումը կարելի է հետևյալ երկու մեթոդով.

1-ին մեթոդ: Երբ b թիվը զույգ է, կարելի է օգտագործել արտահայտություն (2.23)- ը:

$$x = a^{b/2} \cdot a^{b/2} \quad (2.23)$$

Այս դեպքում բազմապատկման գործողությունների քանակը պակասում է համարյա երկու անգամ և որոշվում է հետևյալ բանաձևով՝ $\frac{b}{2} + 1$:

Երբ b թիվը կենտ է, կարելի է օգտագործել (2.24) արտահայտությունը:

$$x = a^{((b-1)/2)} \cdot a^{((b-1/2)} \cdot a \quad (2.24)$$

Բազմապատկման գործողության քանակը որոշվում է $\frac{b}{2} + 2$ բանաձևով [13]:

2-րդ մեթոդ: Ներկայացնել թիվը երկուական հաշվարկման համակարգում: Աստիճան բարձրացնելու գործողության ապարատային իրագործման ժամանակ նպատակահարմար է կիրառել երկրորդ (բինար) մեթոդը: Հիմնվելով աստիճան բարձրացնելու բինար ալգորիթմի վրա՝ մշակենք մոդուլյար (մնացորդային) էքսպոնենտում իրացնող սարքի կառուցվածքային սխեման:

$a^b mod\ m$ հաշվելու համար ամենապարզ ճանապարհը սկզբում a-ի b աստիճան հաշվելն է, այնուհետև արդյունքի ստանալը ըստ մոդուլ m-ի: Սա շատ պարզ մեթոդ է, բայց այն դանդաղ է, քանի որ a-ի b աստիճան բարձրացնելու արժեքը կարող է լինել մեծ թիվ, որը հաշվելու ժամանակը երկար է: Կատարել ավելի էֆեկտիվ էքսպոնենտում ըստ մոդուլի՝ յուրաքանչյուր բազմապատկումը կատարելով ըստ մոդուլ m-ի:

<u>Բինար ալգորիթմի հիման վրա մոդուլով աստիճան բարձրացնելու սարքի նախագծման սկզբունքները</u>

Դիտարկենք բինար ալգորիթմ $a^b\ mod\ m$ արժեքի հաշվարկման համար. b-ն ներկայացնենք երկուական համակարգում [13]՝

$$b = b_{n-1}b_{n-2}\dots b_1b_0 = b_{n-1} \cdot 2^{n-1} + b_{n-2} \cdot 2^{n-2} + \dots + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

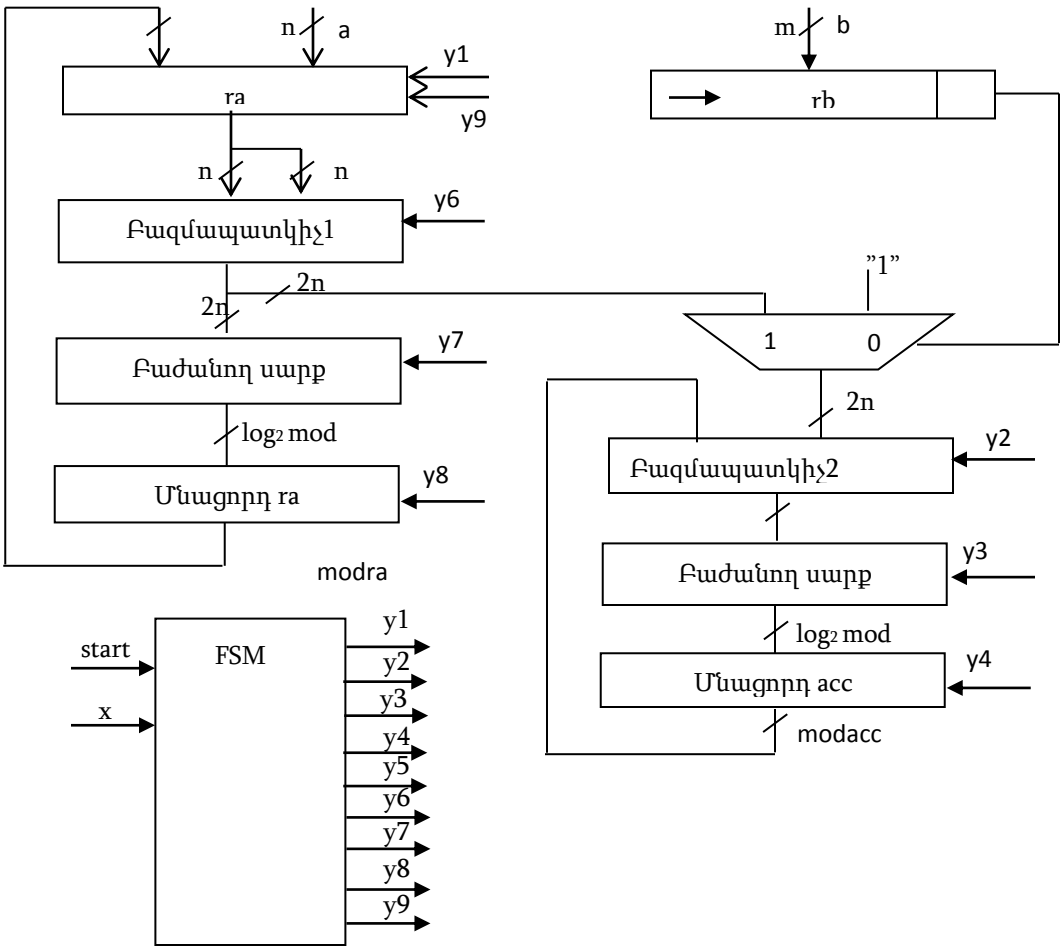
Հաջորդաբար հաշվարկվում են $a_i = a_{i-1} mod\ m$ (2.25) սկզբունքով:

$$a_i = a_{i-1}^2 \cdot a^{b_i} mod\ m = \begin{cases} a_i = a_{i-1}^2 \cdot a\ mod\ m\ , \text{ եթե } bi = 1 \\ a_i = a_{i-1}^2\ mod\ m\ , \text{ եթե } bi = 0 \end{cases} \quad (2.25)$$

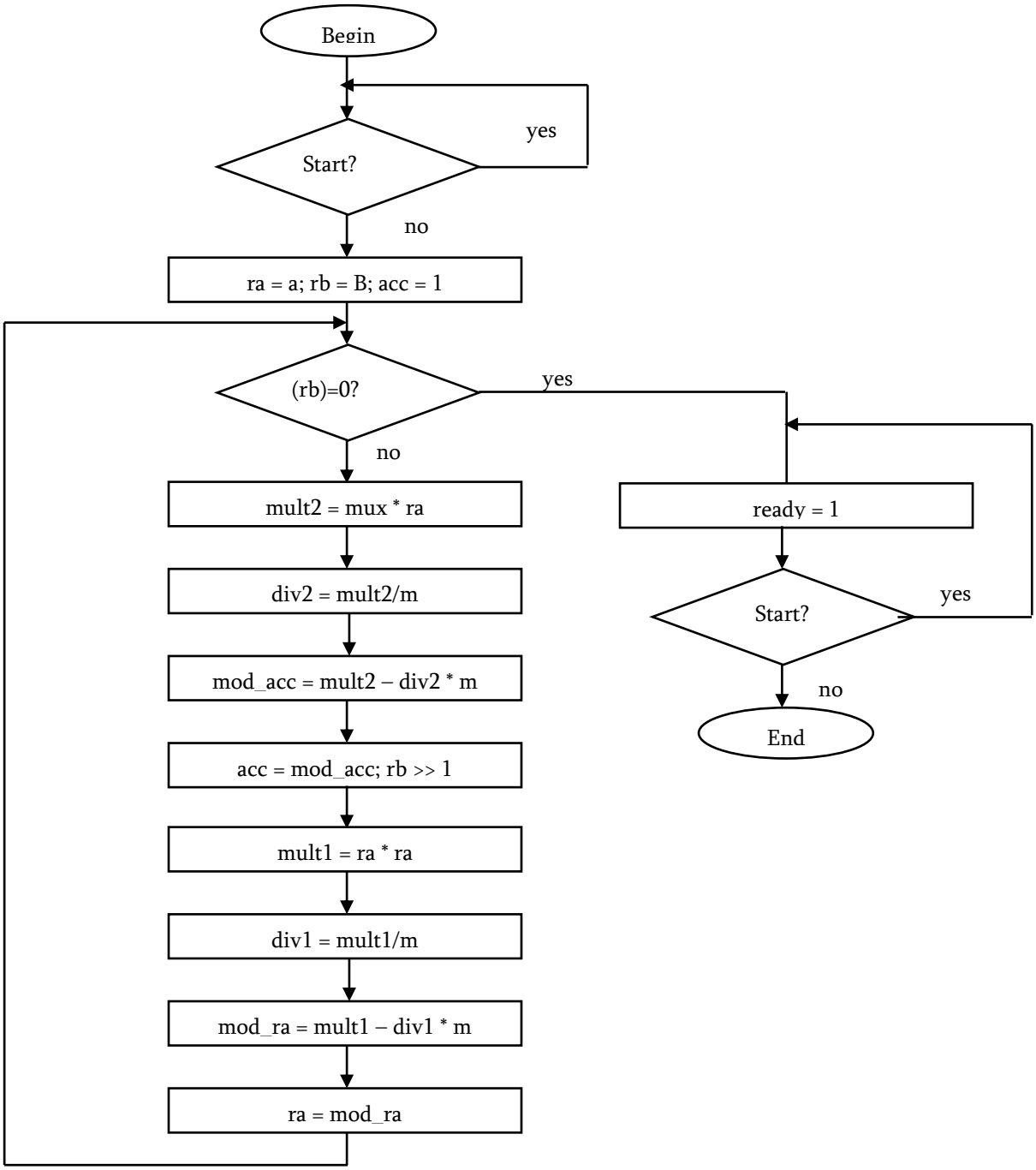
Որտեղ $i \in \{0, \dots, n\}$ և այդ դեպքում $a_{i+1} = a^b mod\ m$ նաև պարզ է, որ ստացվող թվերը չեն գերազանցի m^2: Այս մեթոդը հաճախ անվանում են “քառակուսի բարձրացման- բազմապատկման ” մեթոդ:

Համաձայն ալգորիթմի, $a_{i+1} = a^b mod\ m$ եթե b- ի հերթական բիթը հավասար է “1”-ի, ապա հաշվարկվում է սկզբում a թիվը ըստ մոդուլի, իսկ հետո՝ յուրաքանչյուր արժեքի, քառակուսին, որը և բազմապատկվում է նախորդ բիթի համար որոշված արժեքով: Իսկ եթե b-ի հերթական բիթը հավասար է “0”-ի, ապա նախորդ բիթի համար որոշված արժեքը բազմապատկվում է “1”-ով, այսինքն՝ պարզապես հաշվարկվում է նախորդ բիթի համար որոշված արժեքը ըստ մոդուլ m-ի:

Մոդուլով աստիճան բարձրացման գործողության սարքի կառուցվածքային սխեման պատկերված է նկ. 2.16.-ում, իսկ ալգորիթմի բլոկ-սխեման՝ նկ. 2.17-ում:



Նկ. 2.16. Մոդուլով աստիճան բարձրացման գործողության սարքի կառուցվածքային սխեմա

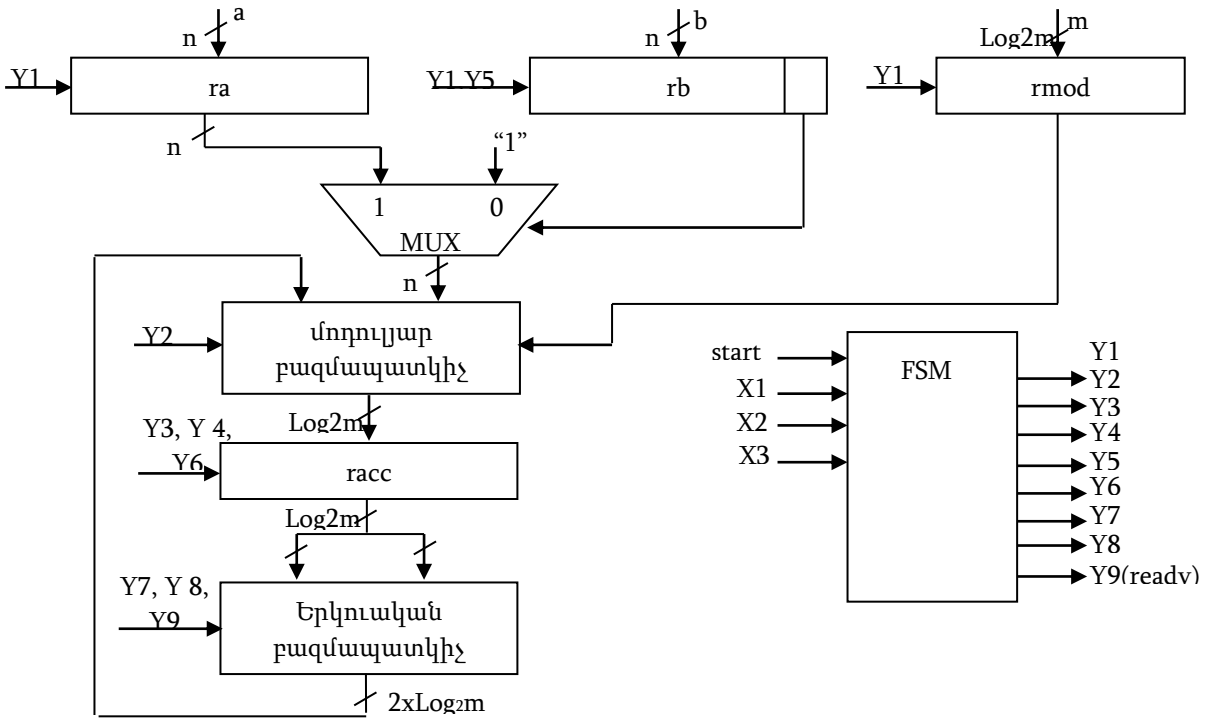


Նկ. 2.17. Մոդուլով աստիճան բարձրացման գործողության ալգորիթմի բլոկ-սխեման

Մշակված ալգորիթմը հիմնված է աստիճանի բարձրացման բինարի մեթոդի վրա, պարզապես բազմապատկման երկու գործողությունները կատարելուց էլ հաշվում և պահում արդյունքների մոդուլ հանդիսացող m-ի բաժանումից ստացված մնացորդները, և այնուհետև հետագա գործողությունները կատարում են մնացորդներով: Start ազդանշանով տրվում է գործողության իրականացման մեկնարկը, ինչից հետո ra

ռեգիստրում գրանցվում է a թիվը, rb ռեգիստրում b թիվը: Այնուհետև, կախված մուլտիպլեքսորի ելքում ստացված արժեքից, կուտակչի պարունակությունը կամ բազմապատկվում է ra-ի պարունակությամբ կամ չի բազմապատկվում, որից հետո հաշվվում է բազմապատկման արդյունքի մնացորդը n -ի բաժանելիս և արդյունքը գրանցում է կուտակիչում: ra ռեգիստրի պարունակությունը քառակուսի է բարձրացվում, որոշվում է ստացված արդյունքի մնացորդը ըստ մոդուլ m-ի և գրանցվում է կուտակչում:

Գործողությունները կրկնվում են մինչև rb ռեգիստրի զրոյանալը, որը որոշում է ծրագրի ավարտը: Սակայն եթե նկ. 2.16.-ի կառուցվածքային սխեմայում բազմապատկիչների և հաջորդող բաժանման սարքերի փոխարեն կիրառենք նախագծված մոդուլյար բազմապատկիչ, սխեմայի կառուցվածքը կարելի է պարզեցնել (նկ. 2.18.):



Նկ. 2.18. "Քառակուսի բարձրացման- բազմապատկման" ալգորիթմի աստիճան բարձրացնելու սարքի կառուցվածքային սխեման

Բացի մոդուլով աստիճան բարձրացնելու՝ ներկայացված տարբերակներից հնարավոր են նաև մասնավոր դեպքեր՝ կախված մոդուլի արժեքներից:

Դիտարկենք մոդուլով աստիճան բարձրացնելու այն տարբերակը, երբ կարելի է կիրառել Ֆերմայի (փոքր) թեորեմը [41].

Եթե m-ը պարզ թիվ է և a -ն չի բաժանվում m-ին, ապա՝

$$a^{m-1} = 1 \ mod \ m \qquad (2.26)$$

Այս աշխատանքում, դիտարկվել է այն ալգորիթմը, որում կկիրառվել է այս բանաձևը, և հետևաբար՝ տվյալ մեթոդը:

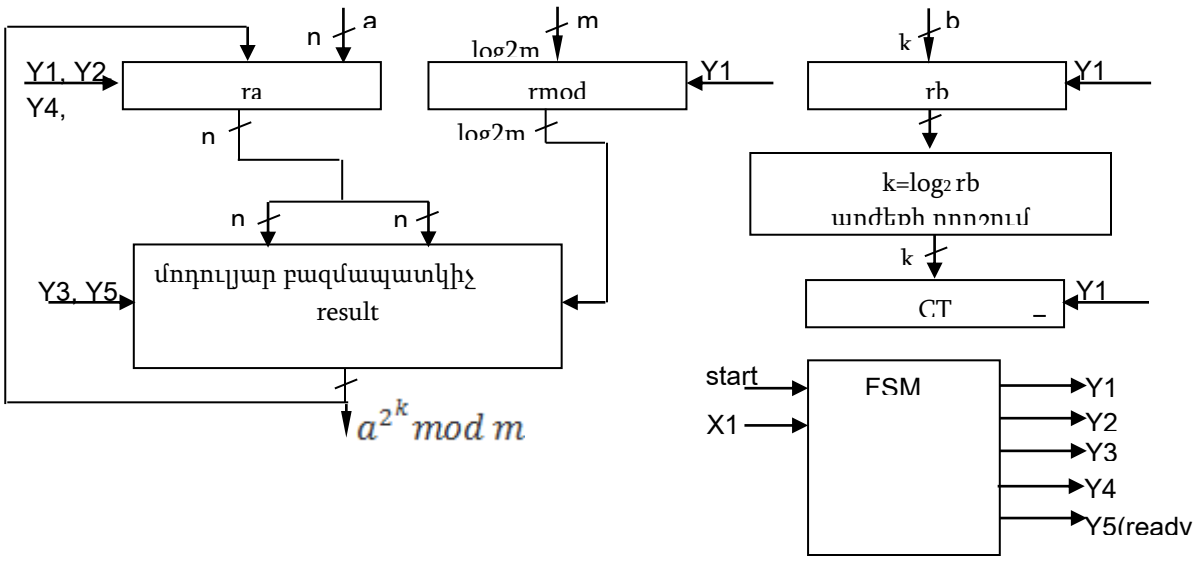
Իսկ այն դեպքում, երբ անհրաժեշտ է հաշվարկել $a^b mod \ m$ արտահայտության արժեքը , որտեղ $b = 2^k$, ապա կարելի է կիրառել բանաձև (2.27)-ը:

$$a^{2^k} \bmod m = ((((a^2 \bmod m)^2 \bmod m)^2 \ldots .) \bmod m) \qquad (2.27)$$

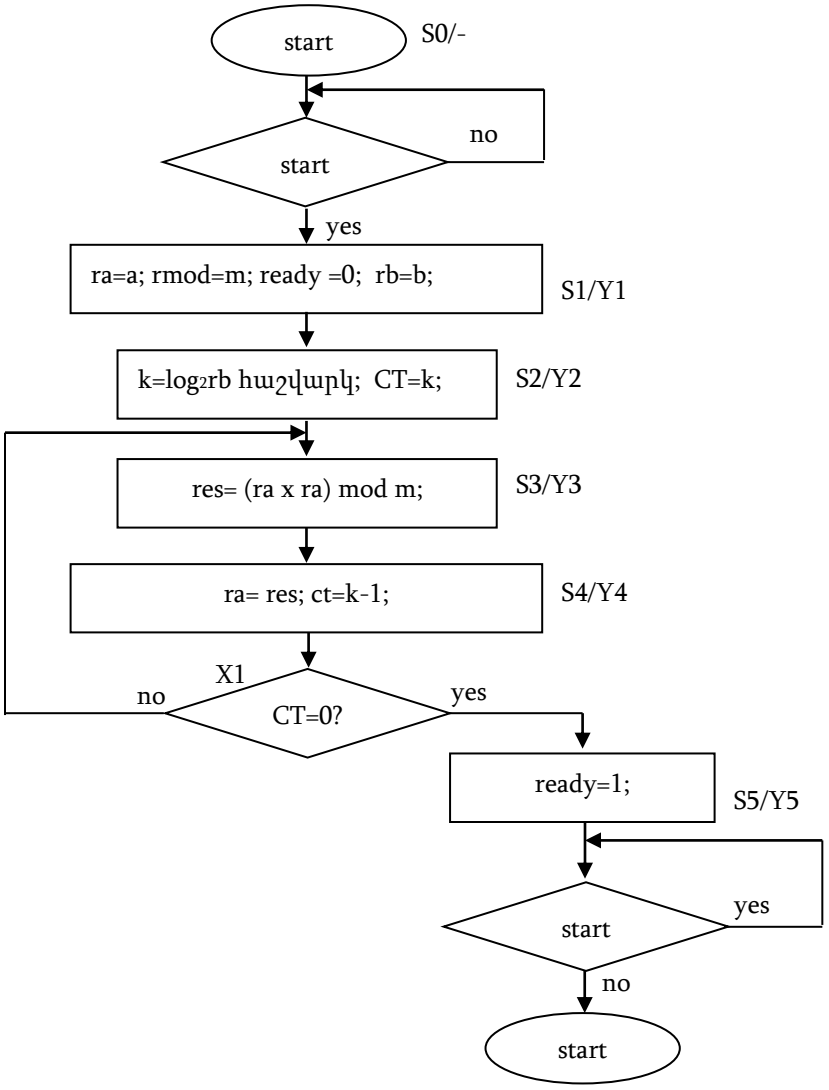
Այս գործողությունը կատարվում է $k-1$ անգամ: $k = 1$տարբերակի համար տվյալ բանաձևը կունենա (2.27)-ի տեսքը [14]:

$$a^{16} \bmod m = ((((a^2 \bmod m)^2 \bmod m)^2) \bmod m)^2) \bmod m \qquad (2.28)$$

Տվյալ արտահայտության հիման վրա մշակված ալգորիթմի բլոկ-սխեման ներկայացված է նկ. 2.20-ում, իսկ սարքի կառուցվածքային սխեման՝ նկ. 2.19-ում:



Նկ. 2.19 $b = 2^k$ արժեքի համար $a^b mod \ m$ արտահայտության հաշվարկման սարքի կառուցվածքային սխեման



Նկ. 2.20. $b = 2^k$ արժեքի համար $a^b mod\ m$ արտահայտության հաշվարկման ալգորիթմի բլոկ սխեման

Աստիճան բարձրացնելու գործողության տվյալ ալգորիթմով իրականացվող սարքի կառուցվածքային սխեման մշակելու համար և այդ սարքում FSM-ի կողմից ձևավորվող կառավարող ազդանշանների ներկայացման համար, ինչպես նախորդ դեպքերում, այստեղ էլ կատարվել է ալգորիթմի բլոկ-սխեմայի Մուրի ավտոմատի նշանադրում:

Գլուխ 1-ում արդեն ներկայացվել են էյլերի ֆունկցիայի հասկացությունը, սահմանումը և օրինակները, որոնց հիման վրա կարելի է կառուցել ըստ մոդուլի աստիճան բարձրացնելու գործողության սարքը՝ էյլերի ֆունկցիայի կիրառմամբ:

Բնական թվերով կատարվող գործողություններից մեկը՝ աստիճան բարձրացնելու գործողությունը, կարելի է ավելի հեշտ և արագ կատարել հիմնվելով մոդուլյար

թվաբանության օրենքների վրա: Այս մեթոդը լայնորեն կիրառվում է ինֆորմատիկայում, հատկապես՝ բաց բանալիով գաղտնագրման ալգորիթմներում: Աստիճան բարձրացնելու գործողության այս տարբերակի առավելությունն այն է, որ օգտագործվում է Էյլերի ֆունկցիան[13, 14]:

$$a^b \pmod m = a \pmod m^{b \pmod{\varphi(m)}} \pmod m \qquad (2.29)$$

Դիցուք տրված են a և m թվերը, որոնք փոխադարձ պարզ են, և $\varphi(m)$-ը Էյլերի ֆունկցիան է: a թիվը փոքրանում է մինչև ըստ մոդուլ m-ի բաժանումից ստացված մնացորդը, իսկ b թիվը՝ ըստ մոդուլ $\varphi(m)$-ի վրա բաժանումից ստացված մնացորդը:

Օրինակ՝ հաշվարկենք $22937^{31} mod\ 23$, որտեղ $a = 22937$, $b = 31$, $m = 23$:

1. Հաշվում ենք a թվի մնացորդը ըստ մոդուլ m-ի՝ $22937 = 6\ mod\ 23$:
2. Որոշում ենք Էյլերի ֆունկցիան m մոդուլի համար՝ $\varphi(23) = 22$:
3. Հաշվում ենք b թվի մնացորդը ըստ φ(m) մոդուլի՝ $31\ mod\ 22 = 9$:
4. Ստանում ենք վերջնական արդյունքը, նախորդ քայլերում ստացված արդյունքների հիման վրա՝ $22937^{31} mod\ 23 = 6^9 mod\ 23$:

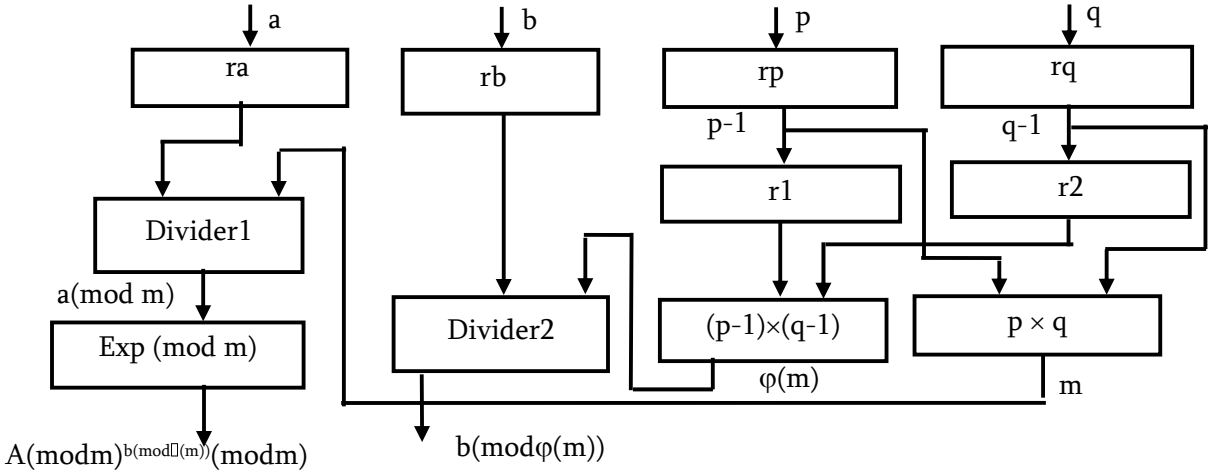
Կառուցենք Էյլերի ֆունկցիայի օգնությամբ ըստ մոդուլի աստիճան բարձրացնելու սարքի կառուցվածքային սխեման: Ալգորիթմի մուտքային տվյալները p և q պարզ ու a և b թվերն են: Սարքի աշխատանքի հետևանքով ելքային տվյալը $a^b mod\ m$ է որտեղ $m = p \cdot q$ մոդուլը պարզ p և q թվերի արտադրյալն է: Սարքի կառուցվածքային սխեման պատկերված է նկ. 2.21.-ում:

ra – a թվի ռեգիստր, rb –b թվի ռեգիստր, rp – p թվի ռեգիստր,

rq – q թվի ռեգիստր, r1 - (p -1) թվի ռեգիստր, r2 – (q-1) թվի ռեգիստր:

Այս ալգորիթմի հիման վրա ըստ մոդուլի աստիճան բարձրացնելու ժամանակ m մոդուլն ընտրվում է այնպես, որ այն լինի p և q պարզ թվերի արտադրյալ, դա հեշտացնում է նրա՝ Էյլերի ֆունկցիայի որոշման փուլը: Ռեգիստրներում m-ի և $\varphi(m)$-ի արժեքների գրանցումից հետո հաշվարկվում և գրանցվում է ըստ մոդում m-ի a թվի մնացորդը և ըստ մոդուլ $\varphi(m)$-ի b թվի մնացորդը:

Էյլերի ֆունկցիայի օգնությամբ ըստ մոդուլի աստիճան բարձրացնելու գործողության ալգորիթմը ներկայացված է նկ. 2.22-ում:



Նկ. 2.21. Ըստ մոդուլի աստիճան բարձրացնելու սարքի կառուցվածքային սխեմա

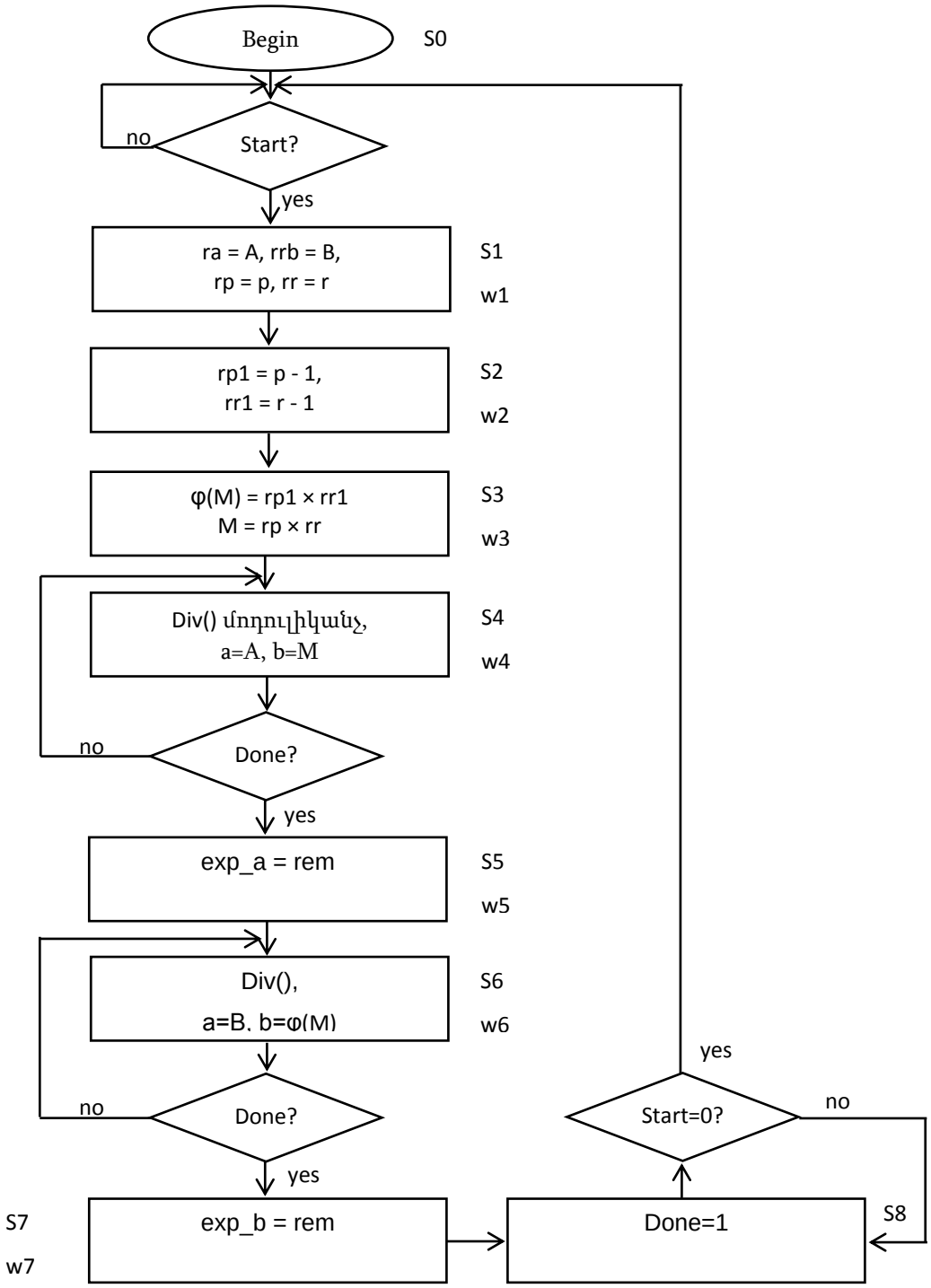
Այս ալգորիթմը ուսումնասիրվել է ավելի մանրամասն և մշակվել էյլերի ֆունկցիայի որոշման բլոկ-սխեման: Մշակված ալգորիթմում մնացորդը որոշվում է Division մոդուլի կանչի օգնությամբ, որը ներառված է ալգորիթմի Verilog (RTL Description) լեզվով նկարագրության կոդում:

Համաձայն ալգորիթմի, a, b, p և q արժեքներն ընտրվում և գրանցվում են համապատասխան ռեգիստրներում: Բոլոր օպերացիաների ավարտից հետո Done ազդանշանը տեղակայվում է “1” վիճակում, իսկ Start ազդանշանը՝ “0” վիճակում:

Ալգորիթմի ավարտից հետո ավտոմատը բերվում է սկզբնական վիճակի և բոլոր կառավարող ազդանշանները զրոյացվում են:

Այսպիսով էյլերի ֆունկցիայի որոշման ալգորիթմի բլոկ-սխեմայի և գրաֆ-սխեմայի մշակումը կարելի է ավարտել:

Verilog լեզվով նկարագրության հիման վրա ISE Project Navigator ծրագրով մշակվել և սինթեզվել է էյլերի ֆունկցիայի որոշման ալգորիթմի տարբերակը: Ծրագրի օգնությամբ որոշվել է թե ինչպիսի և ինչ քանակությամբ տարրեր են օգտագործվում սարքի կառուցման նպատակով և: Արդյունքները ներկայացված են հավելվածում:



Նկ. 2.22. Էյլերի ֆունկցիայի որոշման ալգորիթմի Verilog նկարագրման բլոկ-սխեման

Ինչպես արդեն նշվել է, FPGA-րի օգտագործմամբ սարքերի նախագծման դեպքում կարևոր է FSM-ի կիրառումը[44] և, բացի դրանից, FPGA-րում ներկառուցված բազմա-

պատկիչները զգալիորեն մեծացնում են սարքի արագագործությունը և նվազեցնում FPGA-ի օգտագործված ռեսուրսների քանակը:

Գլուխ 2-ի եզրահանգումները

Ելնելով վերը նշվածից կարելի է եզրակացնել, որ

1. Մոդուլյար գումարիչների մշակված նախագծման մեթոդը չկիրառելով LUT աղյուսակը (նկ. 2.1.), ավելի արդյունավետ է FPGA ռեսուրսների օգտագործման տեսակետից, մոդուլի մեծ արժեքների համար (m=32 և m=64), քանի որ կառավարող սարքը իրագործված է FSM-ի միջոցով:

   $m = 2^k + 1$ և $m = 2^k - 1$ արագագործ մոդուլյար գումարիչները մշակվել են BDD տեխնոլոգիայով, հիմնվելով (2.3) – (2.5) բանաձևերի վրա:

2. Ատենախոսությունում մշակված Բուտի մոդիֆիկացված ալգորիթմով մոդուլյար բազմապատկիչների նախագծման մեթոդները (նկ.2.10 և 2.11) ավելի արդյունավետ են FPGA տեսակետից, մոդուլի մեծ արժեքների համար (m=32 և m=64), համեմատած մոդուլյար բազմապատկիչների նախագծման առկա մեթոդների հետ, քանի որ կառավարող սարքն իրագործված է FSM-ի միջոցով: Կախված վերլուծվող բիտերի քանակից՝ բարդանում են սարքի սխեման իրականացնող տրամաբանական հանգույցների կառուցվածքները, մասնավորապես $Radix\ 8$ մոդիֆիկացման ժամանակ սխեման ներառում է $n + 3$ կարգայնությամբ 8_1 մուլտիպլեքսոր, և $n/2$ մոդուլով հաշվիչ (հաշվիչի կարգայնությունը՝ $log_2 2\ n/2$): Վերակոդավորման գործակիցները՝ -A, -2A, +2A, +A: Իսկ $Radix\ 16$ մոդսիֆիկացման ժամանակ սխեման ներառում է $n + 6$ կարգայնությամբ 16_1 մուլտիպլեքսոր, և $n/3$ մոդուլով հաշվիչ (հաշվիչի կարգայնությունը՝ $log_2 n/3$, որտեղ n-ը բազմապատկվող թվերի կարգերն են: Վերակոդավորման գործակիցները՝ -A, -2A, -3A, -4A, +A, +2A, +3A, +4A: Սակայն այս մեծությունները փոփոխվում են՝ կախված մոդուլի արժեքից: Իսկ $Radix\ 32$ մոդիֆիկացման ժամանակ սխեման ներառում է $n + 9$ կարգայնությամբ 32_1 մուլտիպլեքսոր, և $n/4$ մոդուլով հաշվիչ: Վերակոդավորման գործակիցները +3A,+5A, +6A, +7A, -3A, -5A, -6A, -7A:

Մշակված $m = 2^k$ մոդուլով բազմապատկիչները կարող են առաջարկված կառուցվածքային փոփոխությունների միջոցով (նկ.2.12 և 2.13) կիրառվել նաև $m = 2^k + 1$ և $m = 2^k - 1$ մոդուլներով բազմապատկման գործողություններ իրականացնելու համար: Իսկ ինդեքսային մեթոդի վրա հիմնված մշակված $m = 2^k$ մոդուլով բազմապատկիչներում (նկ. 2.9.), կիրառվել են BDD տեխնոլոգիայով կառուցված գումարիչներ:

3. Մոդուլով աստիճան բարձրացնելու սարքերի նախագծման մեթոդներից արագագործության տեսակետից նախընտրելի են Էյլերի ֆունկցիայի կիրառմամբ սարքերը:

ԳԼՈՒԽ 3. ՄՈԴՈՒԼՅԱՐ ԹՎԱԲԱՆՈՒԹՅԱՆ ԲԼՈԿՈՒՄ ՍԻՆԹԵԶՎԱԾ ՍԱՐՔԵՐԻ ՀԵՏԱԶՈՏՈՒՄԸ և ԱՎՏՈՄԱՏ ՆԱԽԱԳԾՄԱՆ ՀԱՄԱԿԱՐԳԻ ՄՇԱԿՈՒՄԸ

Նախագծման ընթացքում սարքերի կառուցվածքի, կիրառվող տոխնոլոգիաների ընտրությունը հաճախ կատարվում է՝ հիմնվելով նախագծողի անձնական փորձի վրա: Դա մեծ կախվածություն է առաջացնում նախագծողի մասնագիտական որակներից: Օրինակ պարզ սարքերում՝ տվյալ դեպքում մոդուլյար գումարիչների, բազմապատկիչների նախագծման ժամանակ այս մոտեցումն ընդունելի է, ավելի բարդ սարքերում, օրինակ, գաղտնգրման համակարգերում կամ ըստ մոդուլի հսկման համակարգերում անհրաժեշտ է սարքերն իրագործել այնպես, որ հնարավորինս խնամքով օգտագործվեն FPGA-երի ռեսուրսները: Այդ նպատակով առանձին հետազոտվել են մոդուլյար թվաբանության բլոկի մշակված բոլոր տեսակների մոդուլյար գումարիչները, բազմապատկիչները և աստիճան բարձրացնելու սարքերը Spartan-3E FPGA-ի օգտագործմամբ իրագործելիս՝ կախված մոդուլի արժեքի մեծությունից, մուտքային թվերի (օպերանդների) կարգայնությունից: Սարքերի սինթեզումը կատարվել է Xilinx Syntesis Technology ISE 14.0 փաթեթի միջոցով, վերլուծությունները և գնահատումը կատարվել են FPGA-րի ռեսուրսների (LUT-աղյուսակների, Slices-սեկցիաների, մուտք-ելքերի) օգտագործման տեսակետից [2, 12, 37, 70]:

3.1. Սինթեզված մոդուլյար սարքերի հաշվետվությունների ուսումնասիրումը

Գնահատումները կատարվում են ոչ միայն ապարատային ծախսերի, այլ նաև արագագործության տեսակետից: Սարքերի սինթեզման համար ընտրվել է սինխրոազդանշանի գեներացման ժամանակը 5 նվ, այսինքն՝ Verilog նկարագրումը՝ always #5 clk=~clk: Արագագործությունը գնահատվել է սինթեզվող սարքի աշխատանքի սիմուլյացիայի ժամանակ, որն իրականացվել է ModelSim (Mentor Graphics) փաթեթի միջոցով: Այդ արդյունքները ընդհանրացված տեսքով կարելի է ներկայացնել նաև կախվածությունների գրաֆիկի ձևով: Գրաֆիկում ներկայացված է FPGA ընդհանրացված ռեսուրսների քանակի կախվածությունը օպերանդների կարգայնությունից (n=8, n=16 և n=32): Առանձին դիտարկված է m մոդուլի արժեքը [0: 255] միջակայքում, այնուհետ [256: 1023]

միջակայքում: Ինչպես արդեն նշվել է, բոլոր նախագծված սարքերի համար կատարված սինթեզման արդյունքների հաշվետվությունները հավելվածում ներկայացված են աղյուսակներով: Աղյուսակներում զետեղված են տվյալներ FPGA ռեսուրսների օգտագործման և սարքի արագագործության վերաբերյալ: Աղ. 3.1-ում ներկայացված է նախանկան գումարմաբ և մեկ հատ LUT-աղյուսակի օգտագործմամբ մոդուլյար գումարիչի սինթեզման հաշվետվությունը: a և b թվերի n=8, n=16 և n=32 կարգայնության և m մոդուլի արժեքը [0: 255] միջակայքում (այսինքն՝ m-ը 8 կարգանի):

Աղյուսակ 3.1. Նախանկան գումարմաբ և մեկ հատ LUT-աղյուսակի օգտագործմամբ մոդուլյար գումարիչի սինթեզման հաշվետվությունը

Սարքերի օգտագործման հաշվետվություն								
N N	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	Քառամուտք LUTs-ի քանակ	20	1%	34	1%	58	1%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	18	2%	22	1%	36	1%	4656
3.	Կապակցված տրամաբա-նություն պարունակող սեկցիաների (Slices) քանակ	10	100%	10	100%	10	100%	10
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	24	6%	48	26%	80	45%	190
5.	Արագագործություն	25 նվ		25 նվ		25 նվ		

Որպես օգտագործված տրամաբանություն դիտարկվել են՝

- քառամուտք LUT-աղյուսակները, որոնց ընդհանուր քանակը Spartan-3E FPGA-ում 9312 հատ է, n=8 կարգանի a և b թվերի գումարման սարքի սինթեզման դեպքում օգտագործվում է 20 հատ, ինչը կազմում է ընդհանուրի 1%-ը, n=16 դեպքում՝ 34 հատ, ինչը նույնպես կազմում է ընդհանուրի 1%-ը և n=32 դեպքում՝ օգտագործվում է 58 հատ, ինչը նույնպես կազմում է ընդհանուրի 1%-ը,
- Օգտագործված սեկցիաները, որոնց ընդհանուր քանակը Spartan-3E FPGA-ում 4656 հատ է, n=8 կարգանի a և b թվերի գումարման սարքի սինթեզման դեպքում օգտագործվում է 18 հատ, ինչը կազմում է ընդհանուրի 2%-ը, n=16 դեպքում՝ 22 հատ, ինչը նույնպես կազմում է ընդհանուրի 1%-ը, և n=32 դեպքում՝ օգտագործվում է 36 հատ, ինչը նույնպես կազմում է ընդհանուրի 1%-ը,

- Կապակցված տրամաբանություն պարունակող սեկցիաների (Slices), որոնց ընդհանուր քանակը Spartan-3E FPGA-ում 10 հատ է, և ինչպես երևում է աղյուսակից բոլոր դեպքերում օգտագործվել է այս տարրերի 100%-ը,
- Օգտագործված մուտք/ելքերը, որոնց ընդհանուր քանակը Spartan-3E FPGA-ում 190 հատ է, n=8 կարգանի a և b թվերի գումարման սարքի սինթեզման դեպքում օգտագործվում է 24 հատ, ինչը կազմում է ընդհանուրի 6%-ը, n=16 դեպքում՝ 48 հատ, ինչը կազմում է ընդհանուրի 26%-ը, և n=32 դեպքում՝ օգտագործվում է 80 հատ, ինչը կազմում է ընդհանուրի 45%-ը:

Աղյուսակից երևում է նաև, որ անկախ a և b թվերի կարգայնությունից սարքի արագագործությունը 25նվ է:

Աղ. 3.2.-ում ներկայացված է (axb) mod m հաշվարկը $m=2^k$ մոդուլով մոդուլային բազմապատկիչների սինթեզման հաշվետվությունը: a և b թվերի n=8, n=16 և n=32 կարգայնության և m մոդուլի արժեքը [0: 255] միջակայքում (այսինքն՝ m-ը 8 կարգանի) :

Աղյուսակ 3.2. $m=2^k$ մոդուլով մոդուլային բազմապատկիչների սինթեզման հաշվետվությունը

		Սարքերի օգտագործման հաշվետվություն								
N N	Օգտագործված տրամաբանություն	n=8		n=16		n=32		n=64		
		Used	Util.	Used	Util.	Used	Util.	Used	Util.	Available
1.	ՔառամուտքLUTs-ի քանակ	53	1%	94	1%	121	2%	568	8%	9312
2.	Օգտագործված սեկցիա-ների (Slices) քանակ	32	1%	48	1%	72	2%	212	6%	4656
3.	Սեկցիաներում պարունա-կվող տրիգերների քանակ	52	1%	88	1%	106	1%	424	4%	9312
4.	Օգտագործված մուտք/ ելքերի (IOBs) քանակ	16	9%	52	28%	80	43%	148	80%	190
5.	Օգտագործված սինքրո-մուտքերի (CLK) քանակը	2	4%	2	4%	2	4%	6	40%	24
6.	Արագագործություն	40 նվ		40 նվ		40 նվ		40 նվ		

Որպես օգտագործված տրամաբանություն դիտարկվել են՝

- 4 մուտքանի LUT-աղյուսակները, որոնց ընդհանուր քանակը Spartan-3E FPGA-ում 9312 հատ է, n=8 կարգանի a և b թվերի գումարման սարքի սինթեզման դեպքում օգտագործվում է 53 հատ, ինչը կազմում է ընդհանուրի 1%-ը, n=16 դեպքում՝ 96 հատ, ինչը նույնպես կազմում է ընդհանուրի 1%-ը, և n=32 դեպ-քում՝ օգտագործվում է 121 հատ, ինչը կազմում է ընդհանուրի 2%-ը,

- Օգտագործված սեկցիաները, որոնց ընդհանուր քանակը Spartan-3E FPGA-ում 9312 հատ է, n=8 կարգանի a և b թվերի գումարման սարքի սինթեզման դեպքում՝ 32 հատ է, ինչը կազմում է ընդհանուրի 2%-ը, n=16 դեպքում՝48 հատ, ինչը կազմում է ընդհանուրի 1%-ը, և n=32 դեպքում՝ 72 հատ, ինչը կազմում է ընդհանուրի 2%-ը,
- Սեկցիաներում առկա տրիգերներից, որոնց ընդհանուր քանակը Spartan-3E FPGA-ում 4656 հատ է, n=8 կարգանի a և b թվերի գումարման սարքի սինթեզման դեպքում օգտագործվում է 52 հատ, ինչը կազմում է ընդհանուրի 1%-ը, n=16 դեպքում՝ 88 հատ, ինչը նույնպես կազմում է ընդհանուրի 1%-ը, և n=32 դեպքում՝ 108 հատ, ինչը նույնպես կազմում է ընդհանուրի 1%-ը,
- Օգտագործված մուտք/ելքերը, որոնց ընդհանուր քանակը Spartan-3E FPGA-ում 190 հատ է, n=8 կարգանի a և b թվերի գումարման սարքի սինթեզման դեպքում 24 հատ է, ինչը կազմում է ընդհանուրի 6%-ը, n=16 դեպքում՝ 48 հատ, ինչը կազմում է ընդհանուրի 26%-ը, և n=32 դեպքում՝ 80 հատ, ինչը կազմում է ընդհանուրի 45%-ը,
- Օգտագործված սինքրոմուտքերը, որոնց ընդհանուր քանակը Spartan-3E FPGA-ում 190 հատ է, n=8 կարգանի a և b թվերի գումարման սարքի սինթեզման դեպքում օգտագործվում է 24 հատ, ինչը կազմում է ընդհանուրի 6%-ը, n=16 դեպքում՝ 48 հատ, ինչը կազմում է ընդհանուրի 26%-ը, և n=32 դեպքում՝ 80 հատ, ինչը կազմում է ընդհանուրի 45%-ը,
- Օգտագործված սինքրոմուտքերը(CLK), որոնց ընդհանուր քանակը Spartan-3E FPGA-ում 24 հատ է, և ինչպես երևում է աղյուսակից բոլոր դեպքերում օգտագործվել է այս մուտքերի 2 հատը, ինչը կազմում է ընդհանուրի 4%-ը:

Աղյուսակից երևում է նաև, որ անկախ a և b թվերի կարգայնությունից սարքի արագագործությունը 40 նվ է:

Բոլոր նախագծված սարքերի համար կատարված սինթեզման արդյունքները աղյուսակների տեսքով ներկայացված են ատենախոսության հավելվածում:

Գնահատելով սինթեզման արդյունքները՝ առաջարկվում է մոդուլյար սարքերի նախագծման արդյունավետ մեթոդ FPGA-ի ռեսուրսների օգտագործման և այդ սարքի աշխատանքի արագագործության տեսակետից: Գնահատումները կատարվել են հետազոտությունների արդյունքում ձևավորված և փորձնականորեն ստացված հետևյալ բանաձևի հիման վրա՝

$$G = \sum_{i=0}^{5} R_i \cdot T \qquad (4)$$

որտեղ G-ն արդյունավետության հայտանիշն է, Ri-ն FPGA-ի բոլոր օգտագործված ռեսուրսները, T-ն՝ սարքի արագագործությունը,:

3.2. Սինթեզված մոդուլյար գումարիչների հետազոտումը և արդյունքների գնահատումը

Տվյալ աշխատանքում դիտարկված են երեք սկզբունքով իրագործված մոդուլյար գումարիչներ:

Առաջին սկզբունքը ՝ գումարումը (a+b)mod m մեկ հատ մեծ 2^{2n}xM չափսով LUT- աղյուսակի միջոցով (նկ. 1.6), որտեղ M=$\log_2$m, իսկ n-ը a և b թվերի կարգայնությունն է:

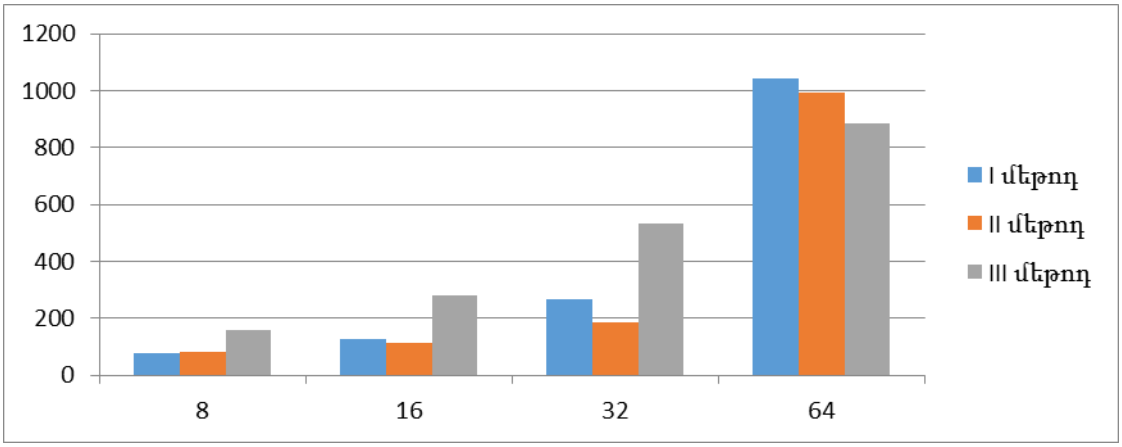
Երկրորդ սկզբունքը ՝ նախական գումարում և գումարի արժեքի մուտքագրումը LUT-աղյուսակ (նկ. 1.7), ինչը կփոքրացնի LUT-աղյուսակի չափերը մինչև 2^{n+1} x M:

Երրորդ սկզբունքը ՝ առանց LUT աղյուսակի օգտագործման սխեմա է, որտեղ օգտագործվում է երկու գումարիչ (նկ. 2.1):

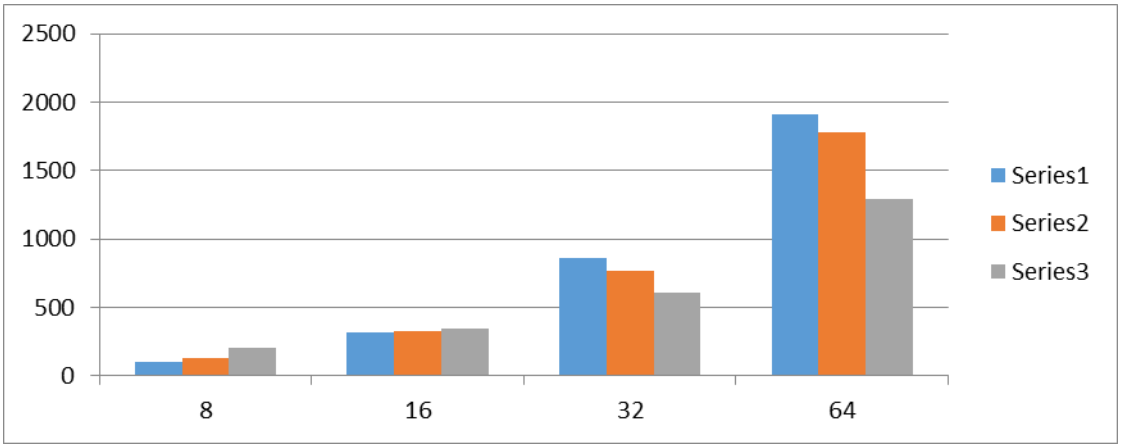
Ներկայացնելով սինթեզման հաշվետվության աղյուսակների արժեքները կախվածության գրաֆիկի միջոցով՝ կարելի է ստանալ նկ. 3.1-ում բերված պատկերը: Տվյալ գրաֆիկը ներկայացնում է FPGA–ի օգտագործված բոլոր տեսակի ռեսուրսների գումարային քանակը՝ ըստ թվերի կարգայնության, m մոդուլի արժեքը [0: 255] միջակայքի (նկ. 3.1.ա) և m մոդուլի արժեքը [256:1023] միջակայքի (նկ. 3.1. բ) դեպքում:

Գնահատելով սինթեզման արդյունքները՝ կարելի է եզրակացնել, որ ըստ մոդուլի արժեքի մեծության և թվերի կարգայնության FPGA-երի ռեսուրսների օգտագործման տեսակետից.

- առաջին մեթոդով նախագծված մոդուլյար գումարիչները նախընտրելի է կառուցել թվերի փոքր (n<16) կարգայնության և m մոդուլի արժեքի [0:255] միջակայքի դեպքում:
- երկրորդ մեթոդով նախագծված մոդուլյար գումարիչները նախընտրելի է կառուցել a և b թվերի մեծ (n>16) կարգայնության և m մոդուլի արժեքի [0:255] միջակայքի դեպքում:
- երրորդ մեթոդով նախագծված մոդուլյար գումարիչները նախընտրելի է կառուցել a և b թվերի մեծ (n>16) կարգայնության և m մոդուլի արժեքի [256:1023] միջակայքի դեպքում:

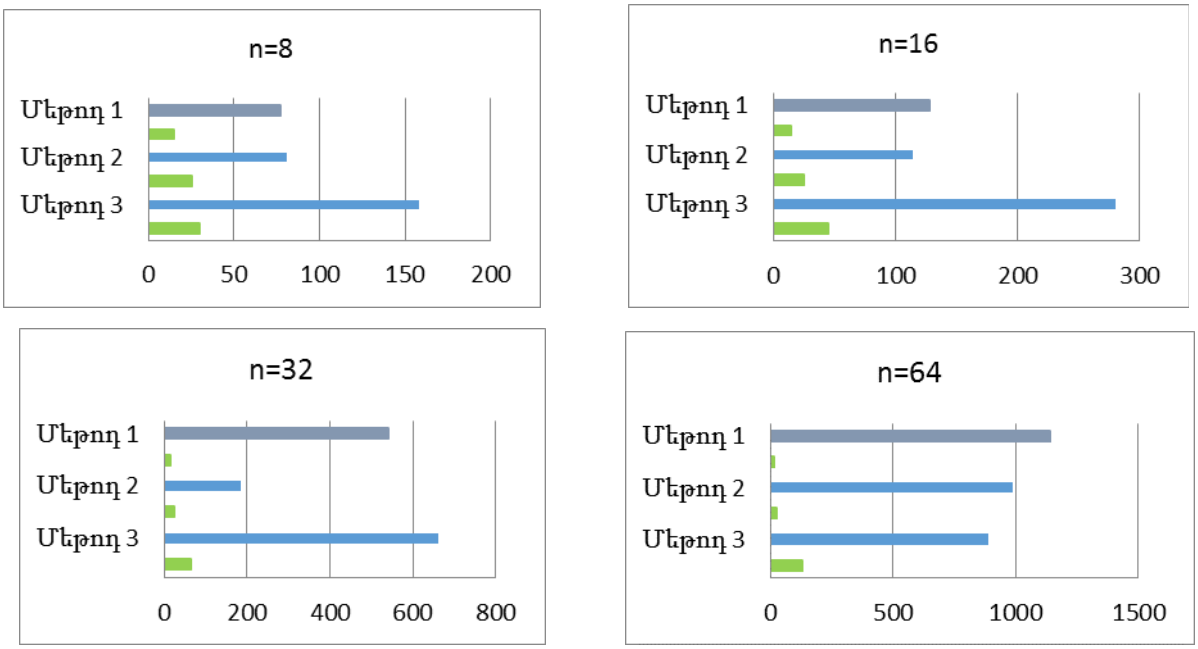


Նկ. 3.1. ա) մոդուլյար գումարիչների նախագծման մեթոդները FPGA ռեսուրսների օգտագործմամաբ m մոդուլի արժեքը [0: 255] միջակայքում



Նկ. 3.1. բ)) մոդուլյար գումարիչների նախագծման մեթոդները FPGA ռեսուրսների օգտագործմամաբ m մոդուլի արժեքը [256:1023] միջակայքում

Համեմատումները կատարվել են նաև արագագործության տեսակետից՝ առաջին երկու մեթոդով նախագծված սարքերի համար այն կախված չէ թվերի կարգայնությունից և մոդուլի արժեքից, սակայն երրորդ տարբերակում թվերի կարգայնությունը ուղղակիորեն ազդում է սարքի արագագործության վրա: Ներկայացնելով սինթեզման հաշվետվության աղյուսակների արժեքները և սարքի աշխատանքի արագագործությունը կախվածության գրաֆիկի միջոցով՝ կարելի է ստանալ հետևյալ պատկերը (նկ. 3.2.):



Նկ. 3.2. Մոդուլյար գումարիչների նախագծման մեթոդների ներկայացումը գրաֆիկի միջոցով m մոդուլի արժեքը [256:1023] միջակայքում

Ինչպես արդեն նշվել է ի սինթեզումը կատարվել է է սինխրոազդանշանի գեներացման 5 նվ ժամանակում, այսինքն Verilog-ով նկարագրվել է always #5 clk=~clk: Այդ արդյունքները ընդհանրացված տեսքով կարելի է ներկայացնել նաև կախվածությունների գրաֆիկի տեսքով: Գրաֆիկում ներկայացված են FPGA ընդհանրացված ռեսուրսների քանակի կախվածությունը թվերի կարգայնությունից (n=8, n=16, n=32 և n=64): Առանձին դիտարկվել է m մոդուլի արժեքը [0:255] միջակայքում, այնուհետ [256:1023] միջակայքում:

Համաձայն նկ. 3.2.-ում պատկերված գրաֆիկների՝ առաջին և երկրորդ մեթոդների (առկա մեթոդներ) կիրառումը նպատակահարմար է օպերանդների փոքր

կարգայնության (n=8 և n=16), իսկ երրորդ մեթոդի (ատենախոսությունում մշակված մեթոդ)՝ օպերանդների մեծ (n=32 և n=64) կարգայնության դեպքում:

3.3. Մոդուլյար բազմապատկիչների հետազոտումը և սինթեզման արդյունքների գնահատումը

Տվյալ աշխատանքում դիտարկվel են հետևյալ սկզբունքներով իրագործված մոդուլյար բազմապատկիչներ:

Առաջին սկզբունքով՝ (axb) mod m մոդուլյար բազմապատկում PROM (Programmable Read-Only-Memory) տիպի հիշող սարքի միջոցով:

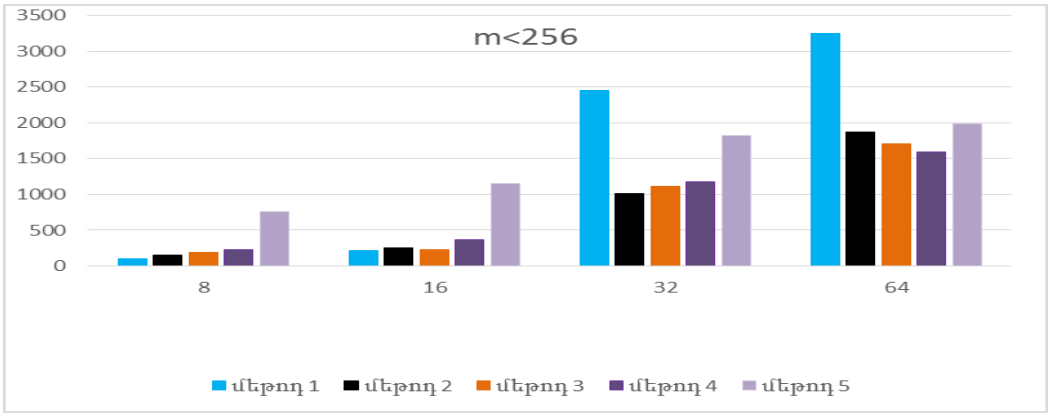
Երկրորդ սկզբունքով՝ (axb) mod m մոդուլյար բազմապատկում ինդեքսային (կամ դիսկրետ-լոգարիթմական) մեթոդով:

Երրորդ սկզբունքով՝ (axb) mod m հաշվարկը քառակուսիների օրենքի հիման վրա նախագծված բազմապատկիչների պայմանական սխեման պատկերված է նկ. 1.10.-ում:

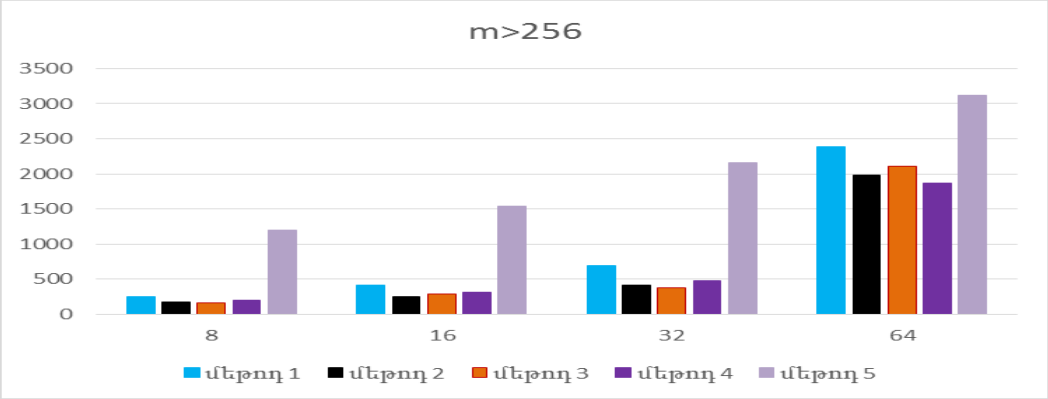
Չորրորդ սկզբունքով՝ (axb) mod m հաշվարկը $m=2^k$ մոդուլով մոդուլյար բազմապատկիչների պայմանական սխեման պատկերված է նկ. 2.14.-ում:

Ինդեքսային մեթոդով (եթե k>3) $m=2^k$ մոդուլով մոդուլային բազմապատկման սարքի պայմանական սխեման ներկայացված է նկ. 2.10.-ում:

Հինգերորդ սկզբունքով՝ (axb) mod m հաշվարկը Բուտի ալգորիթմի կիրառմամբ (Radix 8) սարքի պայմանական սխեման ներկայացված է նկ. 2.11-ում:



Նկ. 3.3.ա) Մոդուլյար բազմապատկիչների նախագծման մեթոդների ներկայացումը գրաֆիկի միջոցով m մոդուլի արժեքը [0:255] միջակայքում



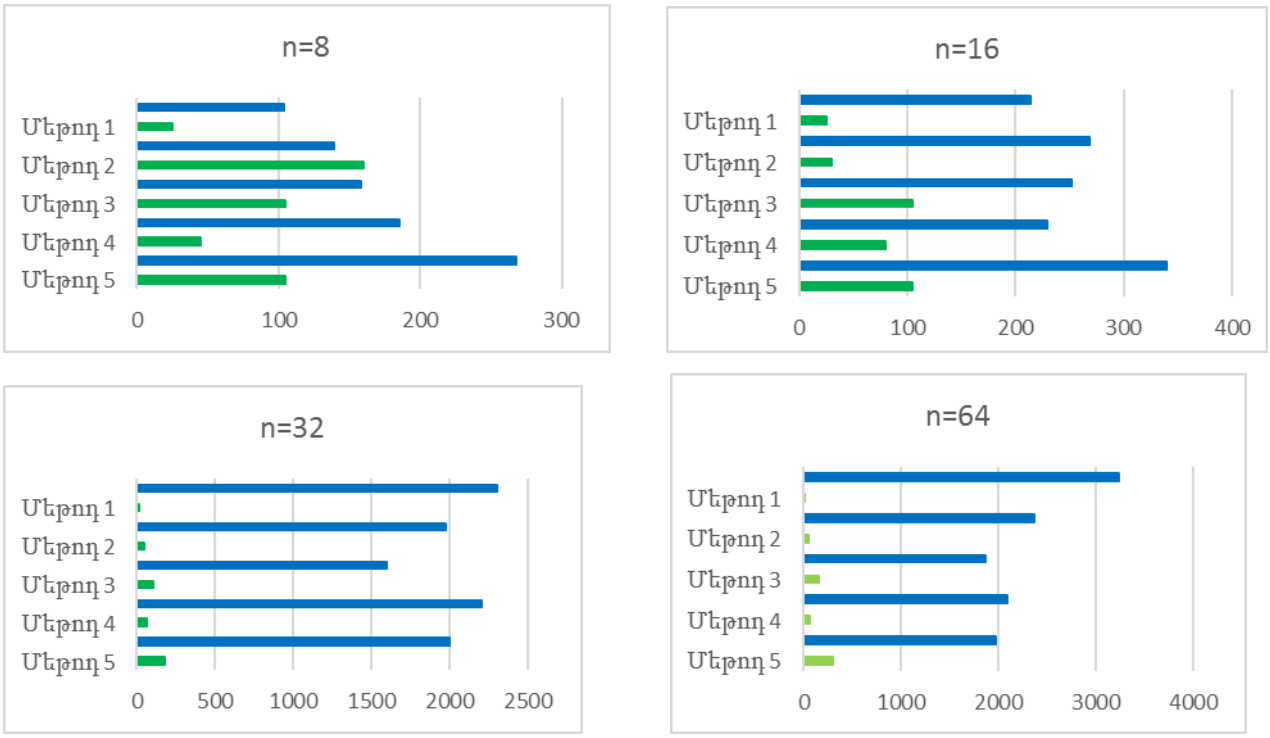
Նկ. 3.3.բ) Մոդուլյար բազմապատկիչների նախագծման մեթոդների ներկայացումը գրաֆիկի միջոցով m մոդուլի արժեքը [256:1023] միջակայքում

Ներկայացնելով սինթեզման հաշվետվության աղյուսակների արժեքները կախվածության գրաֆիկի միջոցով՝ կարելի է ստանալ նկ. 3.3-ում բերված պատկերը: Տվյալ գրաֆիկը ներկայացնում է FPGA–ի օգտագործված բոլոր տեսակի ռեսուրսների գումարային քանակը՝ կախված թվերի կարգայնությունից, m մոդուլի արժեքը [0: 255] միջակայքում (նկ. 3.2.ա) և m մոդուլի արժեքը [256:1023] միջակայքում (նկ. 3.3. բ):

Գնահատելով սինթեզման արդյունքները՝ կարելի է եզրակացնել, որ կախված մոդուլի արժեքի մեծությունից և թվերի կարգայնությունից՝ FPGA-րի ռեսուրսների օգտագործման տեսակետից.

- առաջին մեթոդով նախագծված մոդուլյար բազմապատկիչները նախընտրելի է կառուցել a և b թվերի փոքր (n<16) կարգայնության և մոդուլի արժեքի [0:255] միջակայքի դեպքում:
- երկրորդ մեթոդով նախագծված բազմապատկիչները նախընտրելի է կառուցել a և b թվերի մեծ (n>16) կարգայնության և պարզ թիվ m մոդուլի արժեքի [0:255] միջակայքում:
- երրորդ մեթոդով նախագծված մոդուլյար բազմապատկիչները նախընտրելի է կառուցել a և b թվերի մեծ (n>16) կարգայնության և $m=2^k$ մոդուլի արժեքի [256:1023] միջակայքի դեպքում:
- չորրորդ մեթոդով նախագծված մոդուլյար բազմապատկիչները նախընտրելի է կառուցել a և b թվերի մեծ (n>16) կարգայնության և m մոդուլի արժեքի [0: 255] միջակայքի դեպքում:

Ներկայացնելով սինթեզման հաշվետվության աղյուսակների արժեքները և սարքի աշխատանքի արագագործությունը կախվածության գրաֆիկի միջոցով՝ կարելի է ստանալ նկ. 3.4-ում բերված պատկերը, համաձայն որի առաջին և երկրորդ մեթոդների (առկա մեթոդներ) կիրառումը նպատակահարմար է օպերանդների փոքր կարգայնության (n=8 և n=16), իսկ երրորդ, չորրորդ և հինգերորդ մեթոդների (ատենախոսությունում մշակված մեթոդներ)՝ օպերանդների մեծ (n=32 և n=64) կարգայնության դեպքում:



Նկ. 3.4. Մոդուլյար բազմապատկիչների նախագծման մեթոդների ներկայացումը գրաֆիկի միջոցով m մոդուլի արժեքը [0:255] միջակայքում

3.4. Մոդուլով աստիճան բարձրացնելու սարքերի հետազոտումը և սինթեզման արդյունքների գնահատումը

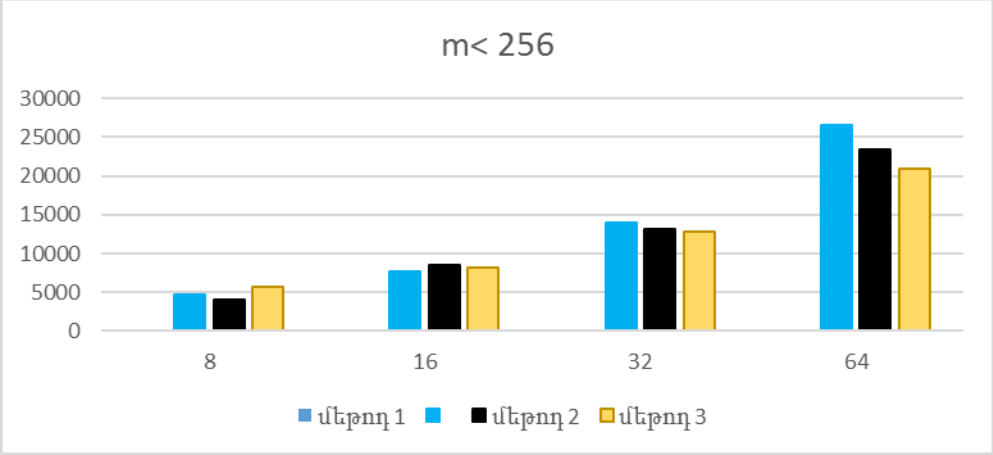
Տվյալ աշխատանքում նախագծված են ըստ մոդուլի աստիճան բարձրացնելու սարքեր՝ իրագործված հետևյալ սկզբունքներով.

Առաջին սկզբունքով բինար ալգորիթմի հիման վրա նախագծված $a^b \bmod m$ մոդուլով աստիճան բարձրացնելու սարքի կառուցվածքային սխեման ներկայացված է նկ. 2.18-ում:

Երկրորդ սկզբունքով Էյլերի ֆունկցիայի կիրառմամբ $a^b \bmod m$ մոդուլով աստիճան բարձրացնելու սարքի կառուցվածքային սխեման ներկայացված է նկ. 2.21.-ում:

Ըստ սինթեզման հաշվետվության աղյուսակների արժեքների կախվածության գրաֆիկը ներկայացված է նկ. 3.5-ում: Տվյալ գրաֆիկը ներկայացնում է FPGA–ի օգտագործված բոլոր տեսակի ռեսուրսների գումարային քանակը, կախված թվերի կարգայնությունից, եթե m մոդուլը [256:1023] միջակայքում է: Գնահատելով սինթեզման արդյունքները՝ կարելի է եզրակացնել, որ FPGA-րի ռեսուրսների օգտագործման տեսակետից.

- առաջին մեթոդով նախագծված մոդուլյար էքսպոնենտման սարքը նախընտրելի է FPGA-րի ռեսուրսների օգտագործման տեսակետից, քանի որ FPGA-երում կան ներկառուցված բազմապատկման սարքեր, որոնք կարելի է նախագծել որպես մոդուլյար բազմապատկիչ,
- երկրորդ մեթոդով նախագծված մոդուլյար էքսպոնենտման սարքը նախընտրելի է արագագործության տեսակետից, ինչը պայմանավորված է նրանով, որ b թիվը էյլերի ֆունկցիան զգալիորեն փոքրացնում է նրա արժեքը, հետևաբար կարգայնությունը:



Նկ. 3.5. Մոդուլով աստիճան բարձրացնելու սարքերի նախագծման մեթոդների ներկայացումը գրաֆիկի միջոցով, m մոդուլի արժեքը [256:1023] միջակայքում

Ինչպես արդեն նշվել է, ատենախոսությունում նախագծված մոդուլյար սարքերի հետազոտումները և արդյունքների գնահատումները կատարվել են ISE Design Suite գործիքամիջոցի կիրառմամբ՝ FPGA-ի օգնությամբ սինթեզման համար: Համեմատման նպատակով աղ. 3.3-ում ներկայացված են տվյալ աշխատանքում նախագծված

մոդուլյար բազմապատկիչների և մոդուլով աստիճան բարձրացնելու սարքերի ու աշխատություններ [70, 93]–ում ներկայացված սինթեզման արդյունքները:

Աղյուսակ 3.3. Առկա և նախագծված մեթոդներով սինթեզված սարքրերի FPGA ռեսուրսների օգտագործումը

NN	Օգտագործված տրամաբանություն	մոդուլյար բազմապատկիչներ		մոդուլով աստիճան բարձրացնելու սարքեր	
		Առկա արդյունքներ x3s100etq	Ստացված արդյունքներ xc3s 500etq	Առկա արդյունքներ	Ստացված արդյունքներ xc3s 500etq
1.	4 մուտքանի LUTs-ի քանակ	4325	1568	568	416
2.	Օգտագործված սեկցիաների (Slices) քանակ	3722	612	432	374
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	2943	824	7632	1389
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	108	148	105	84
	Օգտագործված սինքրոմուտքերի (CLK) քանակը	24	12	6	12

Ատենախոսությունում մշակված սարքերը սինթեզվել են նաև Synopsys ընկերության Design Compiler (.dc) ավտոմատացված նախագծման համակարգի միջոցներով: Մշակված Verilog նկարագրումները սինթեզվել են 32 նմ ստանդարտ բջիջների գրադարանի բազիսում, սինքրոազդանշանի գեներացման ժամանակը 5նվր: Նախագծված և ատենախոսությունում մշակված մեթոդներով սինթեզման արդյունքները՝ սարքերի զբաղեցրած մակերեսի և արագագործության տեսակետից ներկայացված են աղ. 3.4-ում:

Աղյուսակ 3.4. Նախագծված մեթոդներով սինթեզված սարքերի մակերեսները և արագագործությունները

Մոդուլյար սարքերի տեսակներ	Մոդուլյար սարքերի նախագծված մեթոդներ	Զբաղեցրած մակերես (մկմ^2)	Արագագործություն (նվր.)
Մոդուլյար գումարիչներ	Առանց LUT-ի և FSM-ի առկայությամբ	1291.36 (n=32) 1684.56 (n=64)	65 90
Մոդուլյար բազմապատկիչներ	m=2^k մոդուլով բազմապատկում	1734.51 (n=32) 2032.84 (n=64)	105 190
	m=2^k մոդուլով ինդեքսային բազմապատկում	2284.73 (n=32) 2945.62 (n=64)	105 105
	Բուտի ալգորիթմի հիման վրա մոդուլյար բազմապատկում	2649.14 (n=32) 3711.26 (n=64)	185 220

Աղյուսակ 3.4. (շարունակություն)

Մոդուլով աստիճան բարձրացնելու սարքեր	Բինար ալգորիթմի վրա հիմնված մեթոդ	5680.27 (n=32) 7245.14 (n=64)	1710 3655
	Էյլերի ֆունկցիայի կիրառման վրա հիմնված մեթոդ	4511.87 (n=32) 6129.44 (n=64)	1375 2040

3.5. ModSystem ավտոմատացված համակարգի կառուցվածքը և աշխատանքի սկզբունքները

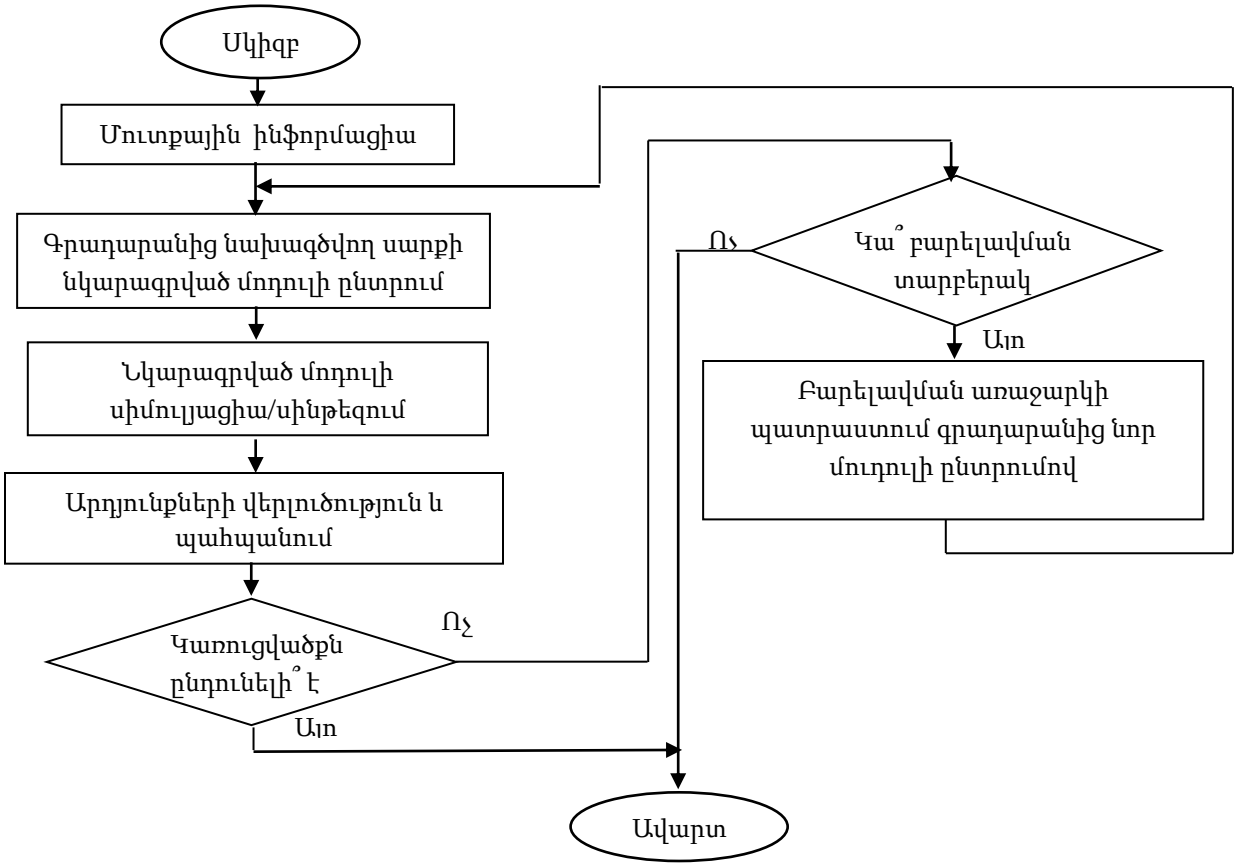
Նախագծման ընթացքում թվաբանական սարքի և ապարատային մասերի կառուցվածքների, կիրառվող տեխնոլոգիաների ընտրությունը հաճախ իրականացվում է՝ հիմնվելով նախագծողի անձնական փորձի վրա: ԲՖ-բլոկերի (FPGA) միջոցով նախագծելիս, երբ նախատեսվում է մեկ բյուրեղի վրա նախագծել տարբեր և մեծ քանակով սխեմաներ, անհրաժշտ է մանրակրկիտ ուսումնասիրել սարքերի կառուցվածքները, և, կախված մոդուլի արժեքի տեսակից ու մեծությունից, կատարել մոդուլյար սարքի նախագծման օպտիմալ մեթոդի ընտրություն [78, 92, 93]:

Որպես ավտոմատացված նախագծման համակարգի մուտքային տվյալներ դիտարկվում են թվաբանական գործողությունների օպերանդների կարգայնությունը, մոդուլի արժեքը և նախագծվող սարքի տեսակը:

Այս մուտքային ինֆորմացիայի հիման վրա կատարվում է գրադարանից նախագծվող սարքի նկարագրված մոդուլի ընտրություն, այնուհետև XST ISE փաթեթի միջոցով կատարվում է ընտրված մոդուլի սինթեզում: Եթե ընտրված մոդուլի կառուցվածքը ընդունելի է նախագծողների համար, FPGA-ն ծրագրավորվում է տվյալ նկարագրման համաձայն:

Եթե առաջարկված կառուցվածքը ընդունելի չէ նախագծողների համար և եթե հնարավոր է կատարել նախագծվող մոդուլյար սարքի կառուցվածքի բարելավում, կրկին կարելի է դիմել ավտոմատացված համակարգի գրադարանին, և կատարել նոր մոդուլի ընտրություն: Այս գործընթացը կրկնվում է, քանի դեռ համակարգի գրադարանում կան նախագծվող սարքի կռուցվածքի ընտրության հնարավորություններ:

Ատենախոսական աշխատանքում մոդուլյար թվաբանության բլոկում թվաբանական սարքերի նախագծման մեթոդների հետազոտման արդյունքների հիման վրա նախագծվել է ավտոմատացված համակարգ, որը կոչվել է ModSystem:



Նկ. 3.6. Մոդուլյար բլոկում մշակված ավտոմատացված նախագծման գործընթացը

Ավտոմատացված համակարգի նախագծման համար առաջադրվել են հետևյալ պահանջները.

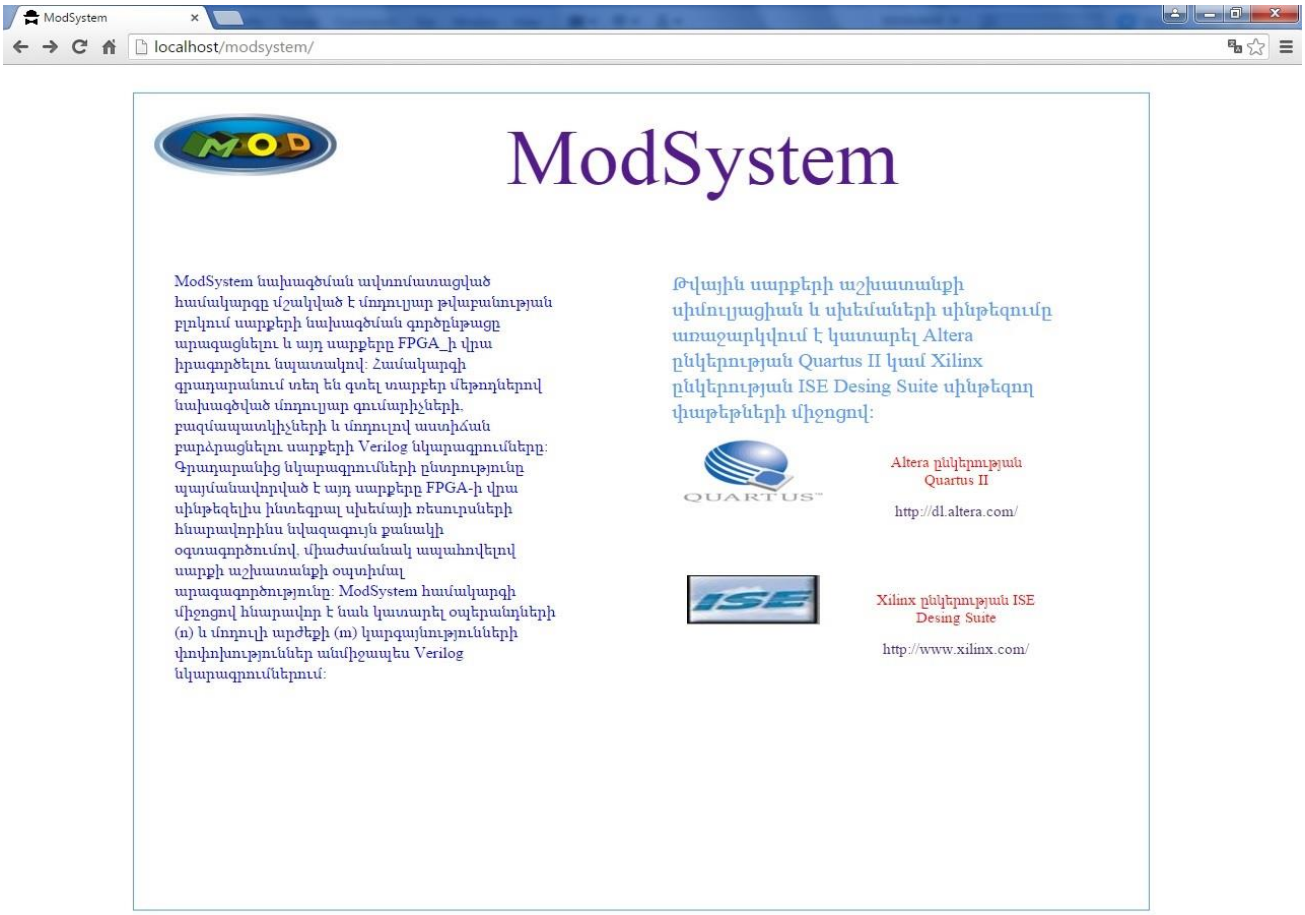
- օգտագործել միայն անվճար և բաց կոդով տարածվող թվային սարքերի նախագծման միջոցներ (XST ISE մոդելավորող և սինթեզող փաթեթը),
- մշակված համակարգը պետք է հնարավոր լինի կիրառել Windows, Linux, Android օպերացիոն համակարգերում:

ModSystem ավտոմատացված համակարգն աշխատում է WEB միջավայրում: Այն բաղկացած է աշխատանքային մի քանի ռեժիմներից: Դրանք են՝

- գրանցման,
- տվյալների հենքերի ձևավորման,
- հիշեցումների կազմակերպման,
- աշխատանքների կատարման ընթացքի մուտքագրման և հուշումների կազմակերպման,

- կատարված աշխատանքի հաշվետվությունների ձևավորման,
- եզրակացությունների տրամադրման,

Նկ. 3.7-ում պատկերված է ստեղծված Web կայքի index.php էջը:

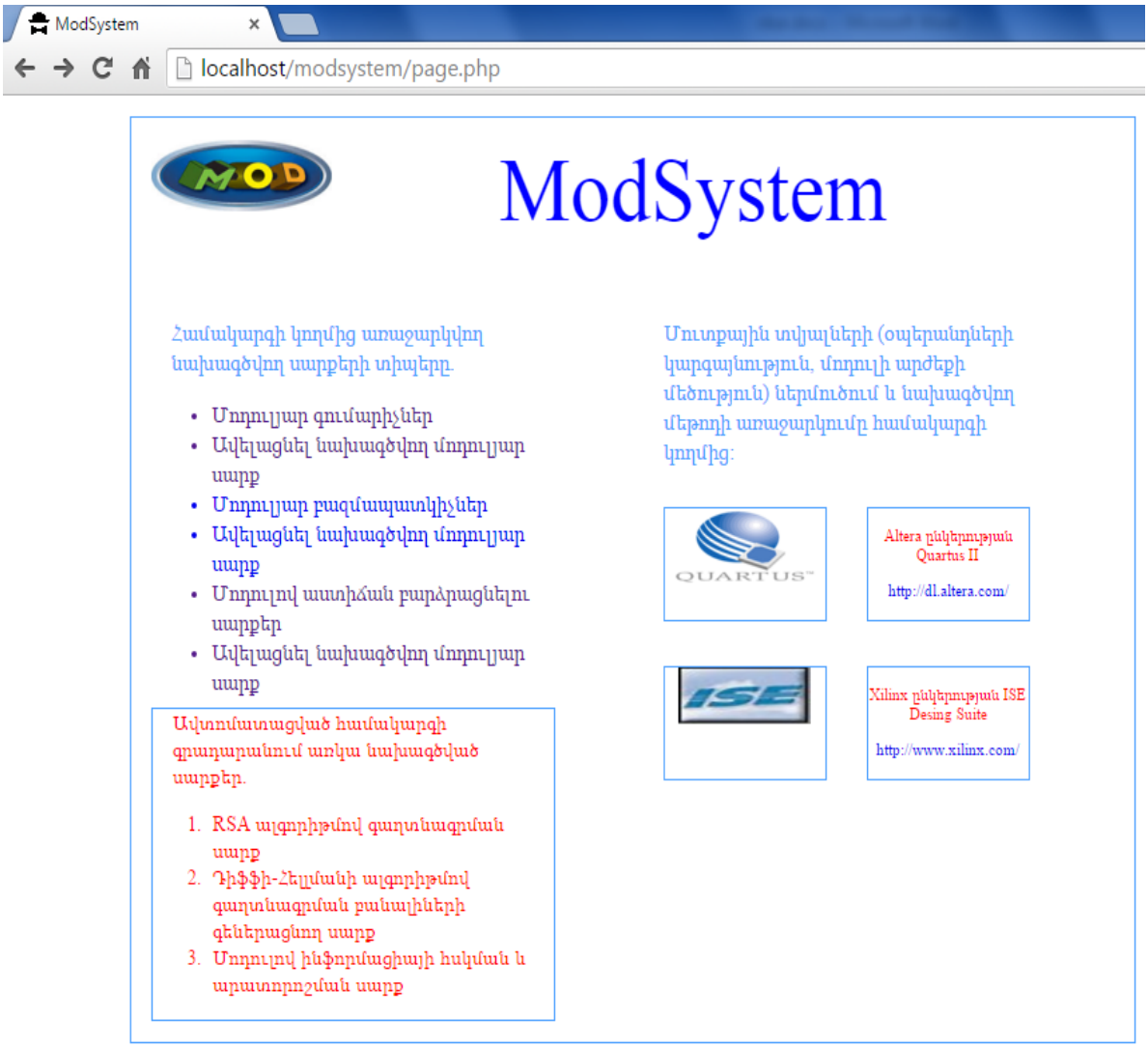


Նկ. 3.7. ModSystem Web կայքի index.php էջը

ModSystem Web կայքը ստեղծվել է PHP օբյեկտային կողմնորոշմամբ ծրագրավորման լեզվով: Անհրաժեշտ տվյալների պահպանման համար օգտագործվել է MySQL տվյալների հենքերի կառավարման համակարգը:

ModSystem ավտոմատացված համակարգը Web կայք է, որի միջոցով կիրառողը հնարավորություն ունի կատարել մոդուլյար թվաբանության բլոկում սարքերի նախագծման մեթոդի ընտրություն: Նկ. 3.8-ում պատկերված է ModSystem Web կայքի page.php էջը, որի միջոցով կարելի է կատարել ընտրություն՝ կախված նախագծվող սարքի տեսակից (մոդուլյար գումարիչներ, բազմապատկիչներ կամ մոդուլով աստիճան բարձրացնելու սարքեր), այսինքն՝ ինչպիսի սարք է ցանկանում նախագծել կիրառողը՝

մոդուլյար գումարիչ, մոդուլյար բազմապատկիչ թե մոդուլով աստիճան բարձրացնելու սարք:

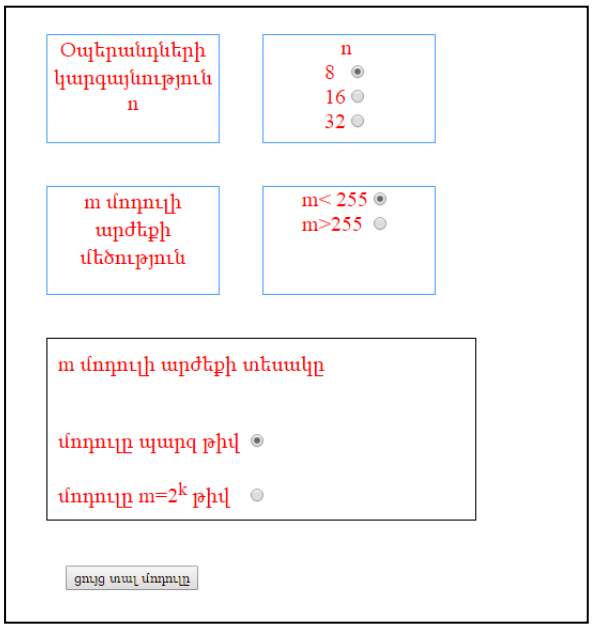


Նկ. 3.8. Համակարգի կողմից առաջարկվող նախագծվող սարքերի տիպերը

Ընտրությունը կարելի է կատարել նաև հիմք ընդունելով սարքի մուտքային տվյալները (թվերի կարգայնություն, մոդուլի արժեք ու մեծություն): Նախագծողը ընտրությունը կարող է կատարել ինքնուրույն, համակարգում ներկայացված բոլոր նախագծված սարքերի Verilog նկարագրման մոդուլներից կամ, օգտվելով այդ նույն սարքերի նախագծման մեթոդի ընտրության առաջարկներից, որոնք մշակված են ատենախոսության աշխատանքում: Քանի որ սարքի նախագծման մեթոդի ընտրության առաջարկները կատարվել են սարքերի սինթեզման հաշվետվությունների հիման վրա,

իսկ գնահատումները՝ FPGA ռեսուրների օգտագործման և սարքի արագագործության տեսակետից, ուստի նախագծողը հնարավորություն ունի ընտրել Verilog նկարագրման մոդուլը՝ կախված թվերի կարգայնությունից և մոդուլի արժեքից կամ մեծությունից:

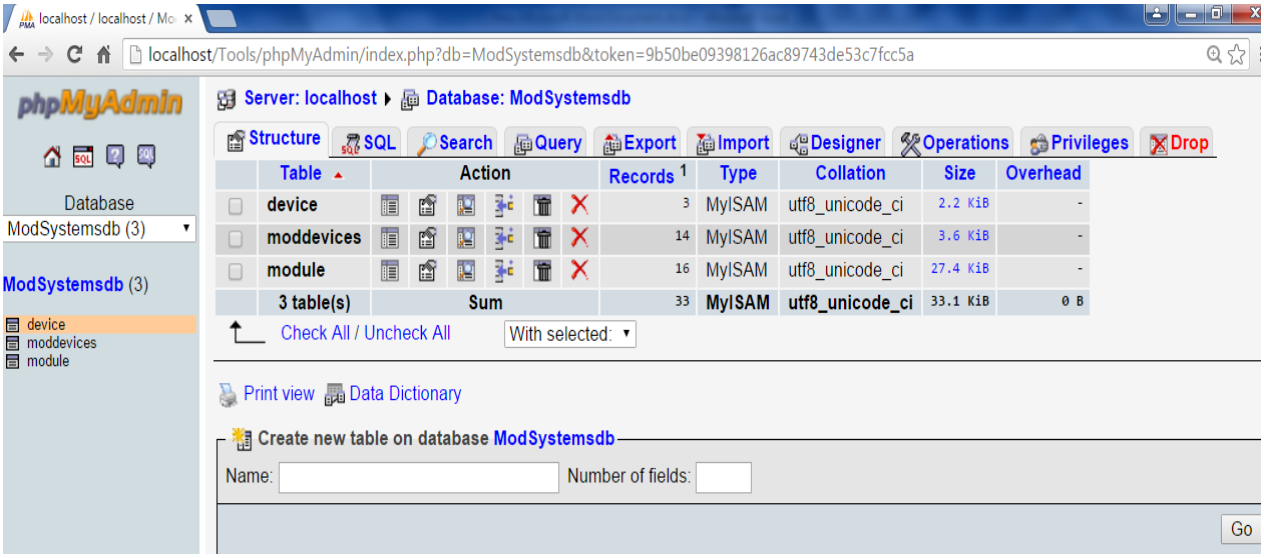
Նկ. 3.9-ում պատկերված է ModSystem ավտոմատացված համակարգում Verilog նկարագրման մոդուլի ընտրության հնարավորությունը ըստ թվերի կարգայնության և մոդուլի արժեքի կամ մեծության:



Նկ. 3.9. Verilog նկարագրման մոդուլի ընտրությունը

Այս մուտքային ինֆորմացիայի հիման վրա տվյալների հենքից կատարվում է ModSystem ավտոմատացված համակարգի նախագծվող սարքի նկարագրված մոդուլի ընտրում, այնուհետև XST ISE փաթեթի միջոցով սինթեզվում է ընտրված մոդուլը: Եթե ընտրված մոդուլի կառուցվածքը ընդունելի է նախագծողների համար, ապա FPGA-ն մշակվում է տվյալ նկարագրման համաձայն:

Եթե առաջարկված կառուցվածքը ընդունելի չէ նախագծողների կողմից, ուստի, եթե հնարավոր է կատարել նախագծվող մոդուլյար սարքի կառուցվածքի բարելավում, կրկին պետք է դիմել ավտոմատացված համակարգի տվյալների հենքին, և կատարել նոր մոդուլի ընտրում: Այս գործընթացը կրկնվում է քանի դեռ համակարգի տվյալների հենքում կան նախագծվող սարքի կռուցվածքի ընտրության հնարավորություններ:



Նկ. 3.10. ModSystemdb տվյալների հենքի կառուցվածքը

Նկ. 3.10-ում պատկերված է ModSystem Web կայքի ModSystemdb տվյալների հենքը, որը բաղկացած է device, moduledevices և module աղյուսակներից:

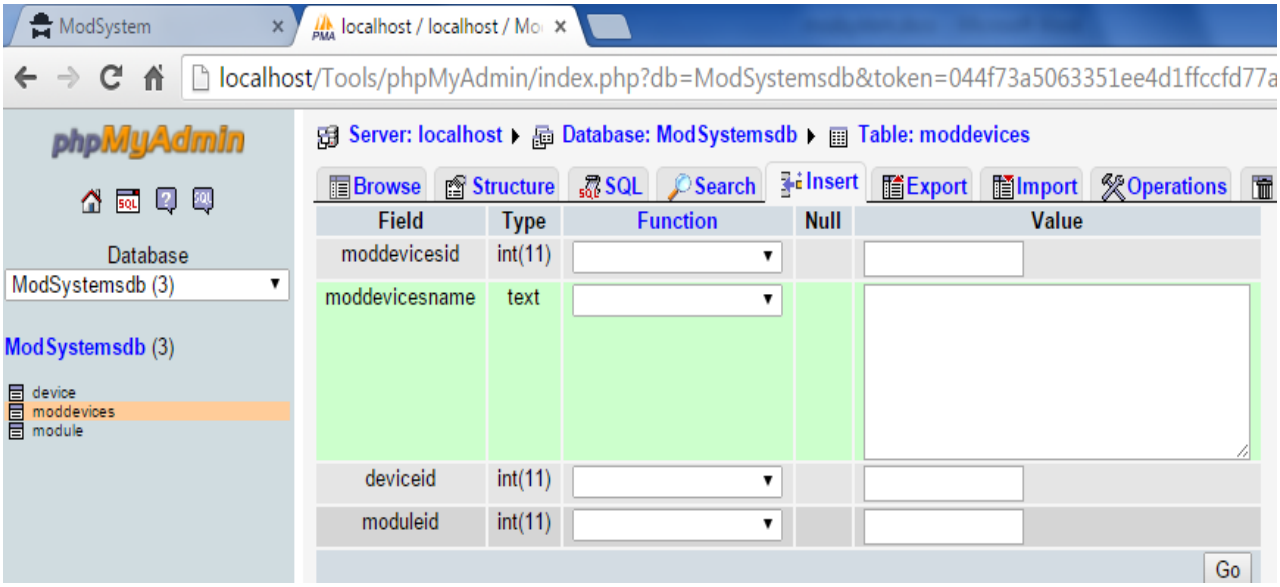
ModSystem նախագծման ավտոմատացված համակարգը մշակված է մոդուլյար թվաբանության բլոկում սարքերի նախագծման գործընթացը արագացնելու և այդ սարքերը FPGA-ի վրա իրագործելու նպատակով: ModSystemdb տվյալների հենքում պահվում են տարբեր մեթոդներով նախագծված մոդուլյար գումարիչների, բազմապատկիչների և մոդուլով աստիճան բարձրացնելու սարքերի Verilog նկարագրությունները:

ModSystemdb տվյալների հենքից նկարագրությունների ընտրությունը պայմանավորված է այդ սարքերը FPGA-ի վրա սինթեզելիս ինտեգրալ սխեմայի ռեսուրսների հնարավորինս նվազագույն քանակի օգտագործումով՝ միաժամանակ ապահովելով սարքի աշխատանքի օպտիմալ արագագործությունը:

ModSystem համակարգի միջոցով հնարավոր է նաև կատարել օպերանդների (n) և մոդուլի արժեքի (m) կարգայնությունների փոփոխություններ անմիջապես Verilog նկարագրություններում:

ModSystem համակարգը հնարավորություն է տալիս տվյալների հենքում ավելացնել մոդուլյար գումարիչի, մոդուլյար բազմապատկիչի, մոդուլով աստիճան բարձրացնելու նախագծման նոր մեթոդներ:

Նկ. 3.11-ում պատկերված է ModSystem ավտոմատացված համակարգի ModSystemdb տվյալների հենքի moddevices աղյուսակում նոր տվյալների ավելացման պատուհանի կառուցվածքը: Համապատասխան տեքստային դաշտերում անհրաժեշտ է մուտքագրել մոդուլյար գումարիչի, մոդուլյար բազմապատկիչի կամ մոդուլով աստիճան բարձրացնելու նախագծման նոր մեթոդ, որից հետո անհրաժեշտ է հաստատել մուտքագրված տվյալները:



Նկ. 3.11. ModSystemdb տվյալների հենքում նոր տվյալների ավելացման պատուհանի կառուցվածքը

ModSystem նախագծման ավտոմատացված համակարգը մշակվել է օբյեկտային կողմնորոշմամբ մոտեցման հիմնական սկզբունքների իրականացմամբ, ինչպես նաև MVC մեթոդաբանության կիրառմամբ, որը թույլ է տալիս համակարգին ավելացնել նոր ֆունկցիոնալ հնարավորություններ:

Գլուխ 3-ի եզրահանգումները

Ընդհանրացնելով գլուխ 3-ը՝ կարելի է կատարել հետևյալ եզրահանգումները՝

1. Մոդուլյար թվաբանության բլոկում առկա և առաջարկված թվաբանական սարքերի համար, XST ISE փաթեթի կիրառմամբ, կատարվել են գնահատումներ FPGA-երի ռեսուրսների օգտագործման և արագագործության տեսակետից, և

Synopsys ԱՆՀ-ի Design Compiler-ի կիրառմամբ՝ սինթեզված սարքերի զբաղեցրած մակերեսի տեսակետից:

2. Կատարված գնահատումների հիման վրա ձևավորվել են առաջարկներ մոդուլյար թվաբանության բլոկում թվաբանական սարքերի նախագծման մեթոդների ընտրության վերաբերյալ՝ կախված թվերի կարգայնությունից և մոդուլի արժեքից,
3. Մոդուլյար թվաբանության բլոկում թվաբանական սարքերի կառուցվածքների ընտրման համար նախագծվել է ավտոմատացված բաց համակարգը, որը թույլ է տալիս օգտագործել սարքերի սինթեզման եղանակների ընտրման վրա նախկինում կատարված սինթեզման արդյունքները:
4. Ավտոմատացված նախագծման բաց համակարգը բաղկացած է չորս ենթահամակարգերից՝ մուտքային արժեքների վերլուծման, սիմուլյացիա/ սինտեզում իրականացման, լավարկված նախագծման մեթոդի ընտության և նախագծված մոդուլյար սարքերի նկարագրումների գրադարանի ենթահամակարգերից:

ԳԼՈՒԽ 4. ՄՈԴՈՒԼՅԱՐ ԹՎԱԲԱՆՈՒԹՅԱՆ ԲԼՈԿՈՒՄ ՄՇԱԿՎԱԾ ՄԵԹՈԴՆԵՐԻ ԿԻՐԱՌՈՒՄԸ

Ինչպես արդեն նշվել է տվյալ աշխատանքում, մնացորդային դասերի համակարգը և մոդուլյար թվաբանությունը դիտարկվում են որպես հաշվարկների արագագործությունը և հուսալիությունը բարձրացնող միջոցներ:

Նշվել է նաև, որ մոդուլյար թվաբանության օգտագործմամբ մասնագիտացված բարդ ֆունկցիոնալ բլոկների առկայությունը կընդլայնի մոդուլյար թվաբանության կիրառման ոլորտները, օրինակ՝ ինֆորմացիայի թվային և թվանշանային հսկումը, սարքերի արատորոշումը և ինֆորմացիայի գաղտնագրումը :

Դեռ վաղ ժամանակներից տեղեկատվության պաշտպանությունը օտարներից նրա ձևափոխման միջոցով հետաքրքրել է մարդկությանը: Հաշվողական համակարգերի զարգացումը հանգեցրեց գաղտնագրման մեթոդների մշակման կատարելագործմանը:

Գաղտնագրման մեթոդները կիրառվում են քոմփյութերներում մշակվող և բազմաթիվ հիշող սարքերում պահվող ինֆորմացայի պաշտպանության, ինչպես նաև հաշվողական համակարգի տարբեր հանգույցների միջև կապուղիներով փոխանցվող տեղեկատվության գաղտնիությունը ապահովելու համար:

Ստորև ներկայացված և ուսումնասիրված գաղտնագրման և հսկման համակարգերը հնարավոր է իրագործել ինչպես ծրագրային, այնպես էլ ապարատային մեթոդներով: Ծրագրային իրագործումը ավելի գործնական և ճկուն է, մինչդեռ ապարատայինը օժտված է արագագործությամբ, պաշտպանվածությամբ, պարզությամբ, սակայն ի տարբերություն ծրագրայինի, ավելի թանկ է:

4.1. Մշակված մեթոդների կիրառումը RSA գաղտնագրման ալգորիթմում

Ժամանակակից գաղտնագրումը ներառում է հետևյալ չորս խոշոր բաժինները.

- սիմետրիկ կրիպտոհամակարգեր
- բաց բանալիով կրիպտոհամակարգեր
- էլեկտրոնային ստորագրությամբ կրիպտոհամակարգեր
- բանալիների գեներացում:

Այսպիսով՝ ինֆորմացիան գաղտնագրման միջոցով ձևափոխվում է այնպես, որ նրա ընթերցումը (վերականգնումը) հնարավոր է դառնում միայն համապատասխան բանալու դեպքում[35]:

Գաղտնագրման համակարգերը լինում են՝

- սիմետրիկ (համաչափ)
- բաց բանալիով (անհամաչափ)

Սիմետրիկ գաղտնագրման համակարգերում կոդավորումը և վերծանումը տեղի է ունենում միևնույն բանալու միջոցով:

Բաց բանալիով գաղտնագրման համակարգում օգտագործվում է երկու բանալի՝ բաց և փակ, որոնք մաթեմատիկորեն կապված են միմյանց հետ: Ինֆորմացիան գաղտնագրվում է բաց բանալու միջոցով, որը հայտնի է բոլորին և վերծանվում է փակ բանալու օգնությամբ, որը հայտնի է միայն հաղորդագրությունը ստացող կողմին:

Գաղտնագրման կայունություն է կոչվում կոդի այն բնութագրիչը, որը սահմանում է կոդի կայունությունը վերծանման նկատմամբ, երբ բանալին հայտնի չէ (կրիպտոանալիզ): Գոյություն ունի գաղտնագրման կայունության մի քանի ցուցանիշ, որոնց շարքում են հետևյալները.

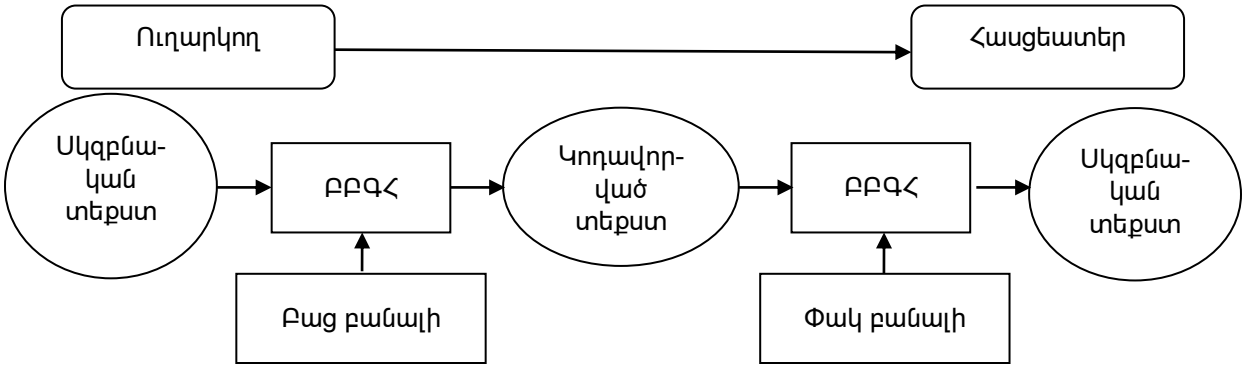
- բոլոր հնարավոր բանալիների քանակությունը,
- ինֆորմացիայի վերծանման համար պահանջվող միջին ժամանակը:

Արդեն նշելվել է, որ տվյալների գաղտնագրման գործընթացը կարելի է իրագործել ինչպես ծրագրային, այնպես էլ ապարատային ճանապարհով: Ապարատային իրագործումը զգալիորեն ծախսատար է, սակայն այն ունի նաև առավելություններ՝ արագագործություն, պարզություն, պաշտպանվածություն և այլն, մինչդեռ ծրագրային իրագործումը պրակտրկ և ճկուն է օգտագործման համար:

Անկախ բարդությունից և հուսալիության աստիճանից գաղտնագրման համակարգի թույլ տեղը գործնականում իրագործման ժամանակ բանալիների բաշխման խնդիրն է:

Որպես այս խնդրի լուծում առաջարկվել է բաց բանալիով գաղտնագրման համակարգի (ԲԲԳՀ) օգտագործումը՝ հիմնվելով դասական և ժամանակակից հանրահաշվի միջոցով ստացված արդյունքների վրա (նկ. 4.1.):

Բաց բանալիով գաղտնագրման համակարգերում օգտագործվում են այսպես կոչված ոչ դարձելի կամ միակողմանի ֆունկցիաներ, որոնք ունեն հետևյալ հատկությունը. տրված x արժեքի դեպքում համեմատաբար հեշտ է որոշել f(x) ֆունկցիան, սակայն եթե տրված է y=f(x), ապա x-ի որոշման համար հեշտ ճանապարհ գոյություն չունի: Ոչ դարձելի ասելով հասկանում ենք ոչ թե տեսականորեն, այլ գործնականորեն ոչ դարձելի ֆունկցիա, երբ հակադարձ արժեքը հաշվարկելը, ժամանակակից միջոցներ կիրառելով և տեսանելի ժամանակահատվածի ընթացքում, ուղղակի անհնար է [76, 88, 89]:



Նկ. 4.1. Ինֆորմացայի կոդավորման և վերծանման փուլերը

Այս պատճառով ինֆորմացիայի հուսալի անվտանգություն ապահովելու համար բաց բանալիով գաղտնագրման համակարգերին ներկայացվում են հետևյալ կարևոր և ակնհայտ պահանջները.

- Սկզբնական տեքստի ձևափոխումը պետք է լինի անհակադարձելի, ինչպես նաև նրա վերականգնումը բաց բանալու հիման վրա պետք է լինի անհնար:
- Փակ բանալու դուրս բերումը բաց բանալու հիման վրա նույնպես պետք է որ ժամանակակից տեխնոլոգիաների մակարդակով անհնար լինի:

Չնայած գոյություն ունեցող բաց բանալիով գաղտնագրման համակարգերի բազմազանությանը՝ ամենահայտնի գաղտնագրման համակարգը համարվում է 1977-78թ.-ին մշակված RSA ալգորիթմը, որն իր անվանումը ստացել է իր իսկ ստեղծողների՝ Ռոն Ռիվեստի (R.Rivest), Ադի Շամիրի (A. Shamir) և Լեոնարդ Ադլմանի (L. Adleman) պատվին:

RSA ալգորիթմի հուսալիությունը երաշխավորված կերպով գնահատելու հնարավորությունը այս գաղտնագրման համակարգի հաճախակի օգտագործման պատճառներից մեկն է, ուստի լայնորեն կիրառվում է բանկերի ցանցային համակարգերում, մասնավորապես հեռավոր օգտատերերի հետ կապված գործառույթներում [77, 89]:

RSA ալգորիթմը բաց բանալիով գաղտնագրման համակարգ է: Դրա հիման վրա տվյալների գաղտնագրման համար գեներացվում են բաց և փակ բանալիներ: Բաց բանալիով ցանկացած գաղտնագրման ալգորիթմի իրագործման առաջին փուլը բանալիների ստեղծումն է, այսինքն՝ բաց և փակ բանալիների գեներացումը և բաց բանալու հրապարակումը: RSA ալգորիթմի դեպքում բանալիների ստեղծումը բաղկացած է հետևյալ փուլերից [77].

1. Ընտրվում է երկու պարզ թիվ ՝ p և q:
2. Հաշվարկվում է դրանց արտադրյալը՝ n = (pxq):
3. Ընտրվում է e (e<n) կամայական ամբողջ թիվն այնպես, որ e և $(p-1)\cdot(q-1)$ թվերը լինեն փոխադարձ պարզ:
4. Էվկլիդեսի մեթոդով լուծվում է ամբողջ թվերի հետևյալ հավասարումը, որը լուծվում d և k փոփոխականների նկատմամբ. $e\cdot d+(p-1)\cdot(q-1)\cdot k=1$: Էվկլիդեսի մեթոդն առաջարկում է բազմաթիվ (d, k) զույգեր, որոնք էլ հենց հանդիսանում են հավասարման լուծումն են:
5. (e, n) թվազույգը հրապարակվում է որպես բաց բանալի:
6. d թիվը պահվում է գաղտնի, հենց դա էլ փակ բանալին է, որը թույլ կտա վերծանել բոլոր այն հաղորդագրությունները, որոնք գաղտնագրվել են (e, n) թվազույգի միջոցով:

Աստիճան բարձրացման բինար ալգորիթմի օգնությամբ կամ Էվկլիդեսի ալգորիթմի կիրառմամբ հակադարձ տարրի հաշվարկը իրականացվում է հետևյալ քայլերի միջոցով. $e\cdot d=1(\mathrm{mod}\,n)$ արտահայտության համար եթե պետք է գտնել e թիվը, ապա ալգորիթմն աշխատում է միայն այն դեպքում, երբ d և k թվերը փոխադարձ պարզ են: e թվի հաշվարկը կատարվում է $k\cdot x+d\cdot e=1$ արտահայտության լուծման միջոցով, որտեղ x թիվը էական չէ:

Եթե M-ը հաղորդագրության նիշերի բազմությունն է, որի տարրերն են mi, ապա յուրաքանչյուր տարրի համար հաշվարկվում է բերված արտահայտությունը.

$$c_i = ((m_i)^e \bmod n \tag{4.1}$$

c_i-ն գաղտնագրի նիշերն են, որոնց բազմությունը գաղտնագրված հաղորդագրությունն է: Գաղտնագրի հերթական նիշի թվային արժեքը բարձրացվում է d աստիճան, և արդյունքը մոդուլյար բաժանվում n-ի, ինչի արդյունքում վերծանվում է բաց տեքստի հերթական նիշի թվային արժեքը.

$$((c_i)^d \bmod n = ((m)_i^{ed}) \bmod n = m_i: \tag{4.2}$$

Կոդավորում՝

- Մուտքային տվյալներ՝ M – հաղորդագրություն, ամբողջ թվերից բաղկացած,
- Ելքային տվյալներ՝ T – կոդավորված հաղորդագրություն:

Վերծանում՝

- Մուտքային տվյալներ՝ T – կոդավորման արդյունք,
- Ելքային տվյալներ՝ M – սկզբնական հաղորդագրություն:

<u>Օրինակ</u>

1. Ընտրում ենք երկու պարզ թիվ՝ p = 3557, q = 2579
2. Հաշվում ենք դրանց արտադրյալը՝ $n = p \cdot q = 3557 \cdot 2579 = 9173503$:
3. Դուրս ենք բերում էյլերի ֆունկցիան՝ $\varphi(n) = (p-1) \cdot (q-1) = 9167368$:
4. Ընտրում ենք բաց ցուցանիշ՝ e = 3:
5. Դուրս ենք բերում գաղտնի ցուցանիշը՝ d = 6111579:
6. Տարածում ենք բաց բանալին՝ (e,n) = (3, 9173503)
7. Գաղտնի ենք պահում փակ բանալին՝ (d,n) = (6111579, 9173503):
8. Ընտրում ենք սկզբնական հաղորդագրությունը (բաց տեքստ)՝ M = 1022:
9. Ստանում ենք կոդավորված տեքստը՝

$$T = M^e \bmod n = 1022^3 \bmod 9173503 = 116:$$

10. Վերծանում ենք սկզբնական տեքստը՝

$$M = T^d \bmod n = 116^{611579} \bmod 9173503 = 1022.$$

Դիտարկենք այս թվերի օգնությամբ իրականացվող կոդավորման բուն գործառույթը: Ուղարկողը բլոկների է մասնատում իր հաղորդագրությունը: Յուրաքանչյուր բլոկի չափը՝ $k = [\log_2(n)]$ բիթ է, վերցված ամբողջ մասով: Հակադարձ գործողությունը կոչվում է՝ լոգարիթմում վերջավոր դաշտում՝ և համարվում է մի քանի կարգ ավելի բարդ: Այսինքն, նունիսկ եթե չարագործին հայտնի են e և n թվերը, ապա c_i –ի միջոցով նա ոչ մի կերպ չի կարողանա ընթերցել սկզբնական mi հաղորդագրությունը: Իսկ ահա ընդունող կողմին վերծանման գործընթացը կհաջողվի, քանի որ նրան հայտնի է գաղտնիության մեջ պահվող d թիվը: Բավականին վաղ ժամանակներում ապացուցվել է Էյլերի թեորեմը, մասնավոր դեպք, ըստ որի, եթե n թիվը կարելի է ներկայացնել երկու պարզ՝ p և q թվերի միջոցով, ապա ցանկացած x –ի համար ճիշտ է հետևյալ հավասարությունը՝

$$(x(p-1)\cdot(q-1))\bmod n = 1 \qquad (4.3)$$

RSA - հաղորդագրությունների վերծանման համար օգտվենք այս բանաձևից:

Հավասարում (4.3)-ի երկու մասը աստիճան բարձրացնենք՝

$$\left(x^{(-y)}\cdot(p-1)\cdot(q-1)\right)\bmod n = 1^{(-y)} = 1 \qquad (4.4)$$

Այժմ (4.4) հավասարման երկու կողմը բազմապատկենք x – ով՝

$$\left(x^{(-y)}\cdot(p-1)\cdot(q-1)+1\right) mod\ n = 1\cdot x = x \qquad (4.5)$$

Իսկ հիմա հիշենք, թե ինչպես ենք բաց և փակ բանալիները ստեղծել. ըստ Էվկլիդեսի ալգորիթմի d թիվը ընտրել ենք այնպես, որ

$$e\cdot d + (p-1)\cdot(q-1)\cdot x^y = 1 \qquad (4.6)$$

այսինքն՝

$$e\cdot d = (p-1)\cdot(q-1)\cdot x^{(-y)} + 1 \qquad (4.7)$$

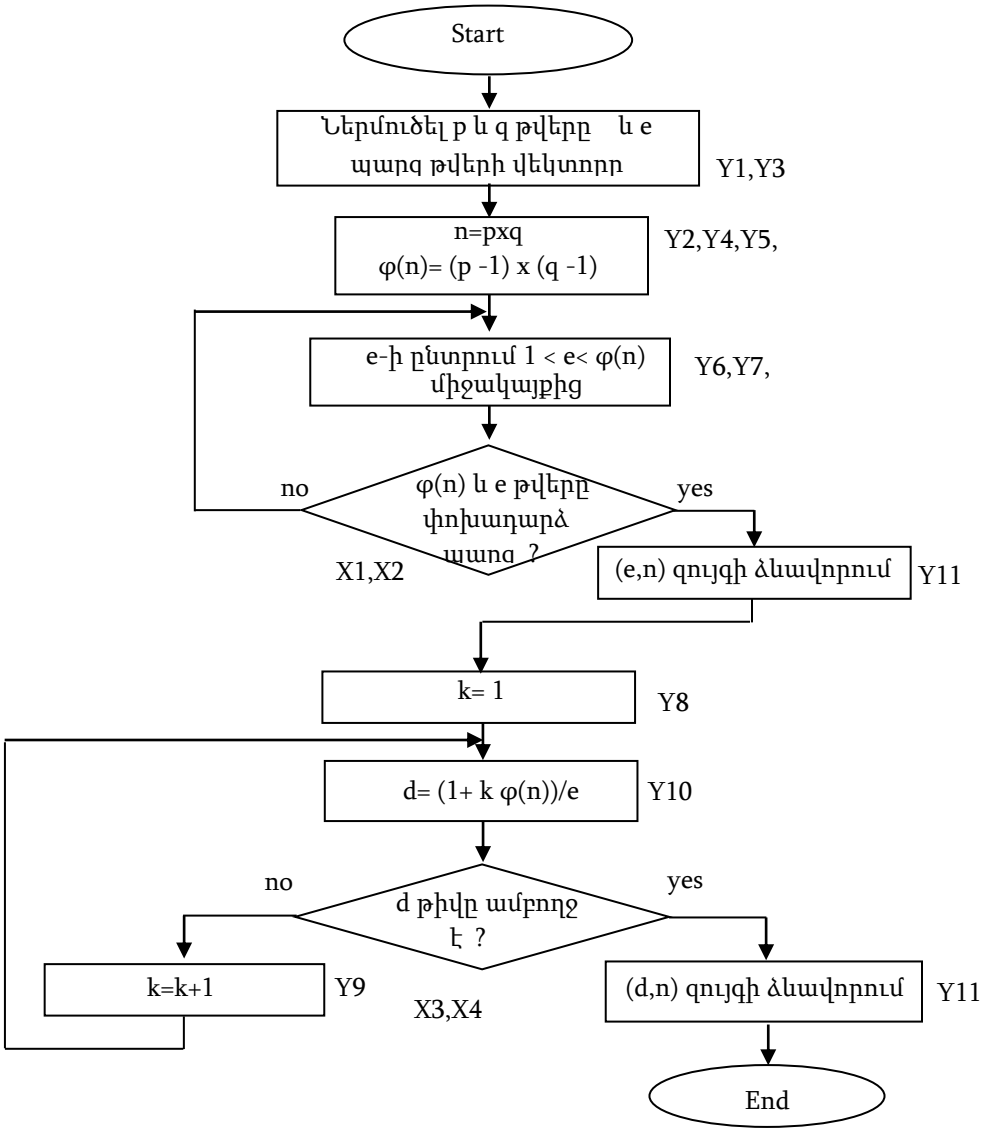
Հետևաբար (4.5) հավասարման մեջ աստիճանի ցուցանիշը (4.7) արտահայտությամբ փոխարինելուց, կստանանք՝

$$\left(x^{e\cdot d}\right) mod\ n = x \qquad (4.8)$$

Այսպիսով, $c_i = (m_i^e)\ mod\ n$ հաղորդագրությունը վերծանելու համար բավական է այն աստիճան բարձրացնել d թվով և հաշվել մոդուլ n – ով:

$$((c_i)^d \bmod n = ((m)_i^{e\cdot d}) \bmod n = m_i: \qquad (4.9)$$

Ինֆորմացիայի գաղտնագրման և վերծանման համար ձևավորվող n,e և d արժեքների որոշման ալգորիթմի ընդհանրացված բլոկ-սխեման բերված է նկ. 4.2-ում:



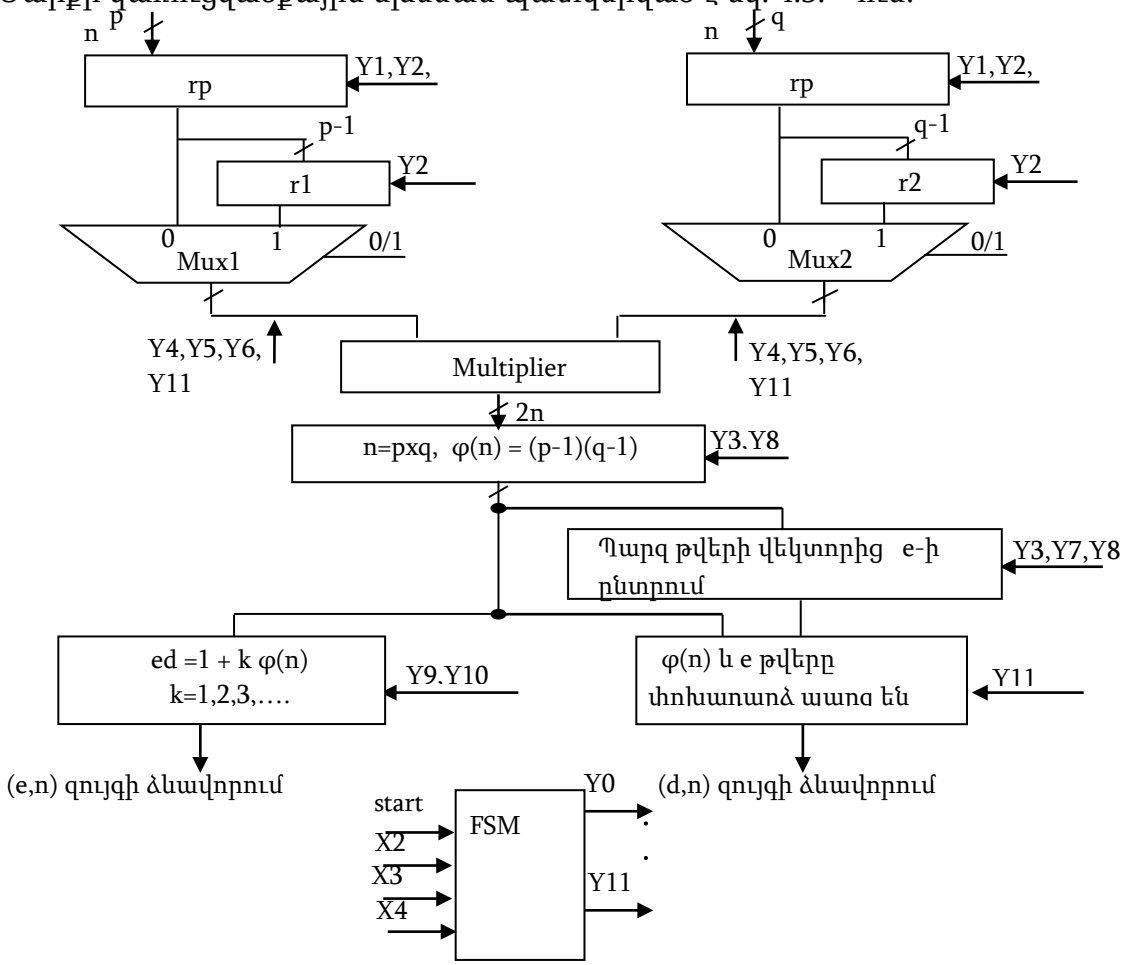
Նկ.4.2. Ձևավորվող n,e և d արժեքների որոշման ալգորիթմի ընդհանրացված բլոկ-սխեման

Իրականում մեծ թվերի աստիճան բարձրացնելու գործընթացը բավականին աշխատատար է ժամանակակից պրոցեսորների համար, նույնիսկ եթե դրանք արտադրվել են գերժամանակակից ալգորիթմներով: Այս պատճառով հիմնական տեքստը կոդավորվում է բլոկների սկզբունքով (շատ ավելի արագագործ), բայց օգտագործվում է սեանսի բանալի, որն արդեն կոդավորվում է անհամաչափ ալգորիթմով՝ ընդունողի բաց բանալու կիրառմամբ, և տեղադրվում է ֆայլի սկզբում:

Ապարատային իրագործման ժամանակ աստիճան բարձրացնելու դեպքում նպատակահարմար է օգտագործել երկուական (բինար) ալգորիթմը[5]: Ալգորիթմի մուտքային տվյալներն են՝ p ,q և e պարզ թվերը: Սարքի աշխատանքի հետևանքով ելքային տվյալներն են՝ (e,n) և (d,n) թվային զույգերը: e թվի որոշման գործընթացը հեշտացնելու համար առաջարկվում այն է ընտրել պարզ թվերի վեկտորից, օրինակ առաջին տասնյակից .

2	3	5	7	11	13	17	19	23	29

Սարքի կառուցվածքային սխեման պատկերված է նկ. 4.3. – ում:



Նկ. 4.3. RSA ալգորիթմով ձևավորվող n,e և d արժեքների որոշման սարքի կառուցվածքային սխեմա

rp – p թվի ռեգիստր, rq – q թվի ռեգիստր, r1 – (p-1) թվի ռեգիստր, r2 – (q-1) թվի ռեգիստր:

Mux1, Mux2 – նույն բազմապատկիչի միջոցով n= p x q արտադրյալը որոշելու համար (եթե հասցեական մուտքը “0” է) և $\varphi(n) = (p-1)\cdot(q-1)$ արտադրյալը որոշելու համար(եթե հասցեական մուտքը “1” է):

Multiplier–ը բազմապատկիչ սարք է, որի ելքում ձևավորվում են n և φ(n) թվերը:

φ(n) թիվը ստանալուց հետո հնարավոր է ընտրել e թիվն այնպես, որը փոխադարձ պարզ է φ(n) թվի հետ և բավարարում է հետևյալ անհավասարությանը՝ 1 < e < φ(n): e թիվը որոշելուց հետո $e \cdot d = 1 + k \cdot \varphi(n)$ բանաձևով (որտեղ k=1,2,3…) հաշարկվում է փակ բանալու զույգը՝ d ամբողջ թիվը:

Համաձայն ալգորիթմի՝ n, d, e արժեքներն ընտրվում և որոշվում են ըստ որոշակի պայմանների: Հետևաբար, անհրաժեշտ է ալգորիթմի՝ վերը նշված բլոկ-սխեմայի համապատասխան (e,n) և (d,n) զույգերի որոշման օպերացիոն բլոկները ներկայացնել առավել մանրամասն կառուցվածքով՝ հաշվի առնելով մոդուլյար թվաբանության առանձնահատկությունները և կանոնները (նկ. 4.4):

Ինչպես նշվեց, ըստ ալգորիթմի անհրաժեշտ է, որ e և φ(n) լինեն փոխադարձ պարզ: Այդ նպատակով ներմուծվում է rem փոփոխականը, որի միջոցով հետևում ենք վերը նշված պայմանի կատարմանը՝

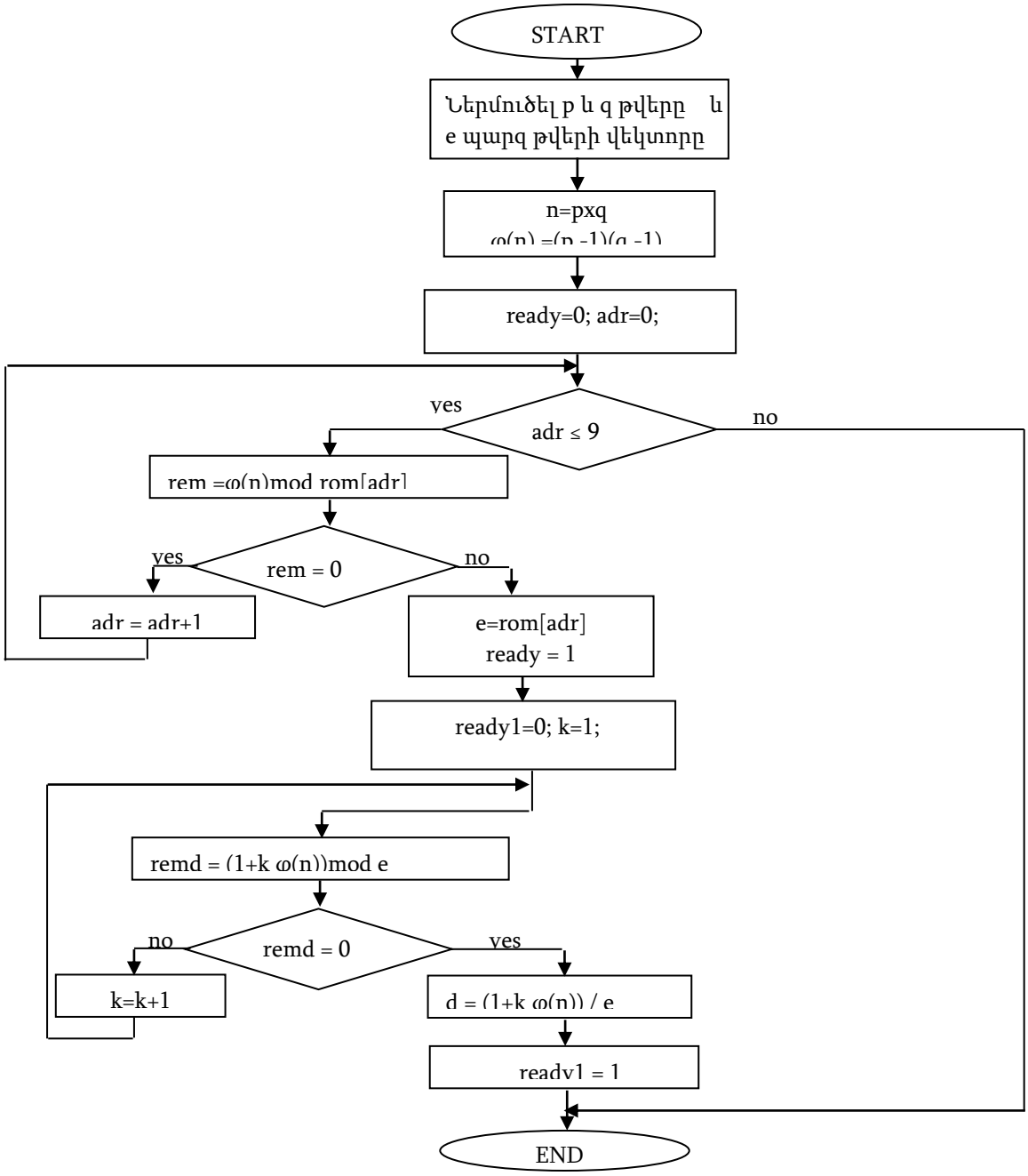
$$rem = \varphi(n) mod (rom[adr]) \qquad (4.10)$$

Այստեղ rom[0] ÷ rom[9] նախապես առաջարկված պարզ թվերի վեկտորն է:

Համաձայն ալգորիթմի՝ (d,n) զույգի d թիվը պետք է լինի ամբողջ թիվ: Այն հաշվարկելու համար ներմուծվում է remd փոփոխականը՝

$$remd = \left(1 + k \cdot \varphi(n)\right) mod\ e \qquad (4.10)$$

Նկար 4.5.-ում ալգորիթմի բլոկ-սխեման ձևափոխված է՝ հետագայում ծրագրային միջոցներով դրա իրականացումը առավել դյուրին դարձնելու նպատակով: Այսինքն ըստ մոդուլի բաժանումը վերածված է սովորական թվաբանական բաժանման և մնացորդի որոշման գործողությունների: RTL-նկարագրումը կազմելու եղանակներից մեկը՝ ԹSU-ի ներկայացումն է օպերացիոն և ղեկավարող սարքերի համագործակցումով [5]:



Նկ.4.4. Բանալիների գեներացման ալգորիթմի ճշգրտված բլոկ-սխեմա

Տվյալների գաղտնագրման պրոցեսը կարելի է իրականացնել և ծրագրային, և ապարատային միջոցով: Ապարատային իրագործումն ունի ավելի մեծ արժեք, նրան հատուկ են նաև առավելություններ՝ բարձր արագագործություն, հուսալիություն և այլն: Ծրագրային իրագործման ժամանակ յուրաքանչյուր հրամանի կատարումն անցնում է հրամանի ընտրման, վերծանման, օպերանդների ընտրման (հնարավոր է հիշողությունից, ինչը դանդաղեցնում է հաշվարկները), կատարման փուլերով, իսկ

ապարատային իրագործման դեպքում հաշվարկները կատարվում են ավելի արագ, քանի որ թվարկված փուլերը բացակայում են: Մշակելով RSA ալգորիթմով գաղտնագրման սարքի ապարատային իրագործումը՝ կարելի է պնդել.

- Ըստ RSA ագորիթմի՝ գաղտնագրման և վերծանման գործողություններն ունեն հետևյալ ընդհանրացված տեսքը՝ $a^b \bmod n$:
- Գաղտնագրման և վերծանման գործողությունների կատարումը արագացնելու համար, ի տարբերություն աստիճան բարձրացնելու գործողության ավանդական ալգորիթմի, նպատակահարմար է օգտագործել աստիճան բարձրացնելու գործողությունը՝ ըստ mod n–ի:
- Մոդուլով բաժանման համար նպատակահարմար է օգտագործել արդյունքի ձևավորման փոփոխական ժամանակով բաժանման ալգորիթմը:
- Verilog լեզվով RTL-նկարագրման համար յուրաքանչյուր բլոկ ներկայացվում է առանձին always պրոցեդուրային օպերատորի միջոցով, իսկ եթե այն կոմբինացիոն սխեմա է, ապա կարող է նկարագրվել նաև assign օպերատորի կիրառմամբ:

Արդյունքում՝ ստացվում է սարքի RTL-նկարագրումը, որի հիմքի վրա կարելի է սինթեզել սխեման: RSA ալգորիթմով ինֆորմացիայի կոդավորման և վերծանման (*Encryption, Decryption)* բլոկներում մոդուլյով աստիճան բարձրացնելու գործողությունն իրացվել է էյլերի ֆունկցիայի օգնությամբ և նախագծված սարքի աշխատանքը մոդելավորվել Xilinx սիմուլյատորի օգնությամբ: Նշված սարքի և այլ հեղինակների կողմից նույն սիմուլյատորով ստացված, աշխատություններ [70, 96] ներկայացված, RSA ալգորիթմով ինֆորմացիայի կոդավորման և վերծանման բլոկների արագագործությունները բերված են աղ. 4.1-ում:

Աղյուսակ 4.1. Առկա և նախագծված մեթոդներով ինֆորմացիայի կոդավորման և վերծանման բլոկների (RSA ալգորիթմ) արագագործությունների համեմատում

RSA ալգորիթմի ինֆորմացիայի կոդավորում/ վերծանում	Արագագործություն (նվր)	
	Նախագծված սարքեր	Առկա սարքեր
Կոդավորման (*Encryption)* բլոկ	275	312 [70], 330 [96]
Վերծանման (*Decryption)* բլոկ	285	333 [70], 340 [96]

Ինչպես երևում է աղ.4.1-ից կոդավորման բլոկի արագագործությունը նախագծված մոդուլով աստիճան բարձրացման սարքի կիրառման շնորհիվ աճել է 12 %-ով, իսկ վերծանման բլոկինը՝ 14%-ով:

RSA գաղտնագրման ալգորիթմով սարքի ապարատային իրագործումը FPGA-ի վրա ընդունվել է օգտագործման “ԵրԿՄԳՀԻ” ՓԲԸ-ում, որպես փոխանցվող ինֆորմացիայի պաշտպանությունն ապահովելու միջոց և “Ֆեմբոքս” ՍՊԸ-ում, որպես բաց կապուղիներով թվային հեռուստաալիքների հեռարձակման ապահովման միջոց: Ներդրման ակտերը ներկայացված են հավելվածում:

Այս սարքերում, ի տարբերություն աստիճան բարձրացնելու գործողության ավանդական ալգորիթմի, օգտագործվել է մշակված էյլերի ֆունկցիայի կիրառմամբ մոդուլով աստիճան բարձրացնելու բլոկը: Տվյալ մեթոդով նախագծված մասնագիտացված թվային սարքի կիրառմամբ կրճատվել է FPGA-ի օգտագործվող ռեսուրսները մինչև 18%:

4.2. Մշակված մեթոդների կիրառումը Դիֆֆի-Հելլմանի բանալիների գեներացման ալգորիթմում

1976թ. Ուիթֆիլդ Դիֆֆին և Մարթին Հելլմանը գաղտնագրային համակարգում առաջարկեցին նոր մոտեցում՝ բաց բանալիով գաղտնագրություն, որը փոխեց գաղտնագրության մասին պատկերացումները:

Գաղտնագրման ալգորիթմներում առավել բարդ խնդիր է բանալիների մշակումը:

Գոյություն ունեն սեանսային բանալիներ արտադրելու բազմաթիվ մեթոդներ:

Ենթադրենք ունեք երկու բաժանորդ՝ A կողմ և B կողմ, և երկու թվեր՝ g և p, որոնք հայտնի են երկու կողմերին եվ գաղտնիք չեն նաև այլ բաժանորդների համար:

Բոլորի համար անհայտ բանալի ստեղծելու համար երկու կողմերը վերցնում են երկու մեծ պատահական թվեր՝ A կողմը A-ն, իսկ B կողմը B-ն, ապա A կողմը հաշվում է A- ի արժեքը ըստ բանաձև (4.12)-ի՝

$$A = g^a \bmod p \qquad (4.12)$$

և ուղարկում է B կողմին , իսկ B կողմը հաշում է b-ն ըստ բանաձև (4.13)-ի՝

$$B = g^b \bmod p \qquad (4.13)$$

և ուղարկում A կողմին :

Ենթադրենք , որ հարձակվողը կարող է ստանալ այդ երկու արժեքները , այլ ոչ թե դրանց փոփոխությունները:

Երկրորդ փուլում A կողմը հաշվարկում է ըստ բանաձև (4.12)-ի՝

$$B^a \bmod p = g^{a \cdot b} \bmod p \qquad (4.14)$$

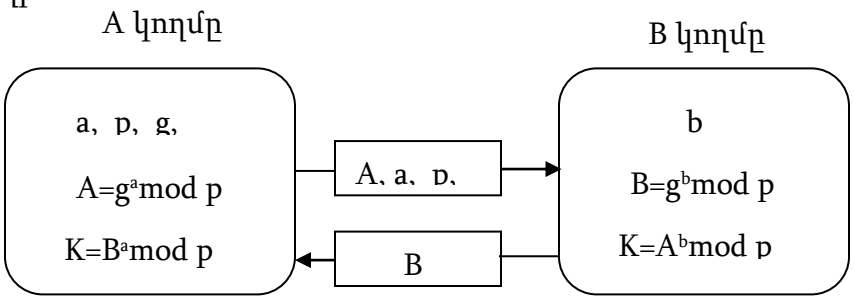
Իսկ B կողմը հաշվարկում է ըստ բանաձև 4.13-ի՝

$$A^b \bmod p = g^{a \cdot b} \bmod p \qquad (4.15)$$

Բանաձևեր (4.14) –ից և(4.15)-ից երևում է, որ A և B կողմերը ձևավորում են միևնույն արժեքը՝

$$K = g^{a \cdot b} \ mod \ p : \qquad (4.16)$$

K-ն կարելի օգտագործել որպես գաղտնի բանալի, քանի որ այն անհասանելի կլինի հարձակվողին:



$$K = A^b \bmod p = (g^a \bmod p)^b \bmod p = g^{a \cdot b} \bmod p = (g^b \bmod p)^a \bmod p = B^a \bmod p$$

Նկ. 4.5. A և B կողմերի բանալիների փոխանակումը և K գաղտնի բանալիու ստեղծումը

Արդյունքում Դիֆֆի-Հելլմանի գաղտնագրման համար ստեղծվեց բանալի K թիվը:

Դիֆֆի-Հելլմանի ալգորիթմով կարելի է գեներացնել գաղտնագրման k բանալին, որը հետագայում կկիրառվի սիմետրիկ գաղտնագրման ժամանակ, որը պետք չէ փոխանակել կողմերի միջև: Ալգորիթմի առավելությունն այն է, որ լուծում է դասական գաղտնագրման հիմնական խնդիրը՝ բանալիների փոխանակման խնդիրը [35, 77]:

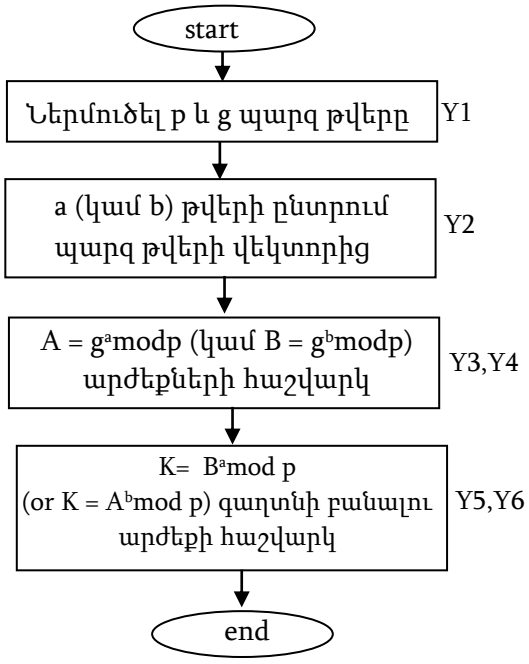
Դիֆֆի-Հելլմանի ալգորիթմի կիրառման դեպքում գաղտնագրման բանալու ստեղծումը և յուրաքանչյուր կողմի կատարած աշխատանքը բաղկացած է հետևյալ փուլերից.

1. Գեներացնում է պատահական **a** թիվը, որը հանդիսանում է **A** կողմի փակ

2. Համաձայնեցված մյուս կողմի հետ՝ ընտրում է **p** և **g** բաց պարամետրերը (սովորաբար p-ն և g-ն ընտրում է մի կողմը և փոխանցում մյուս կողմին), որտեղ p-ն և g-ն պատահական պարզ թվեր են:
3. Հաշվարկվում է **A** բաց բանալին՝ օգտագործելով **a** փակ բանալին՝ $\mathbf{A=g^a \bmod p}$:
4. Փոխանակվում է **A** բաց բանալին **B** կողմի հետ, որը նույն սկզբունքով ձևավորում է իր **B** բաց բանալին՝ $\mathbf{B=g^a \bmod p}$:
5. Հաշվարկվում է ընդհանուր **K** գաղտնի բանալին՝ օգտագործելով մյուս կողմի բաց բանալին և սեփական փակ բանալին՝ $\mathbf{K=B^a \bmod p=A^b \bmod p}$:

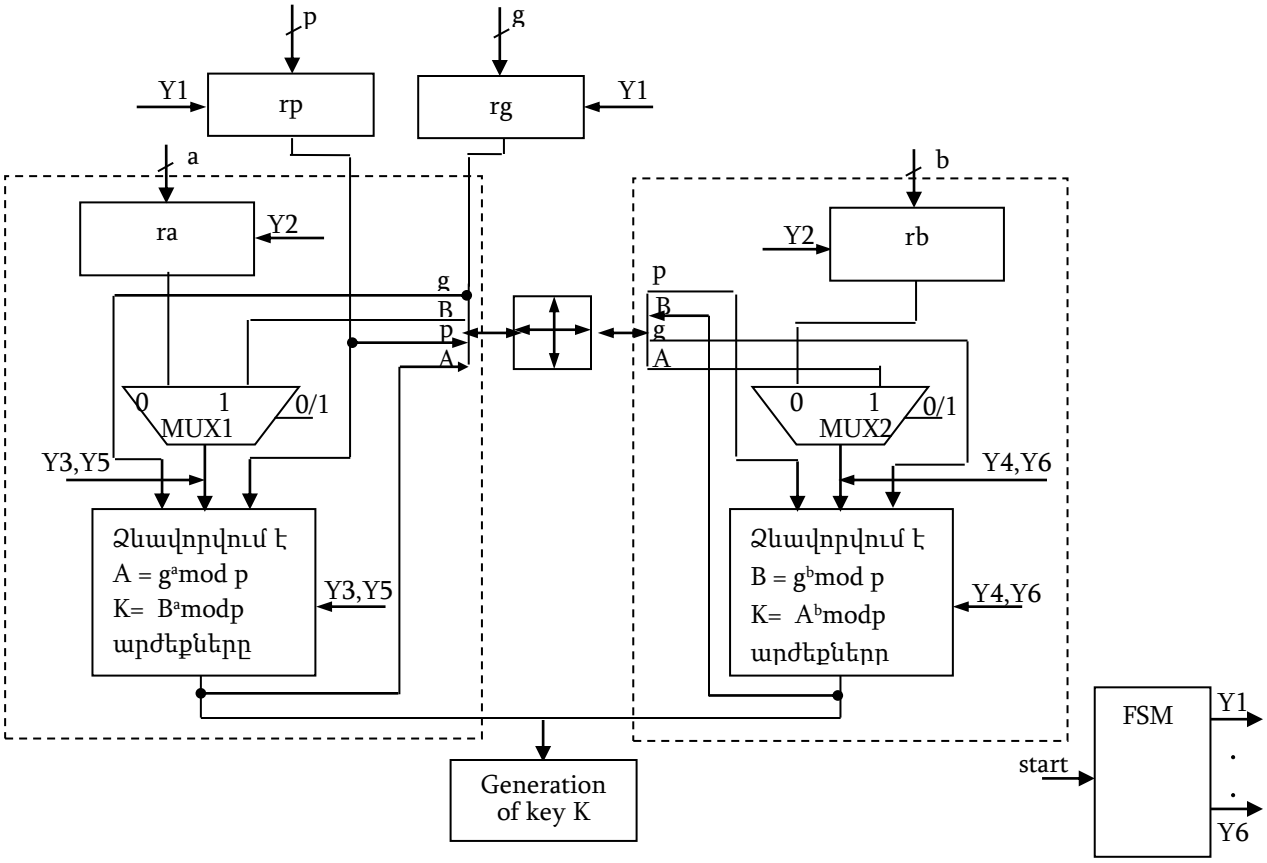
Այս գործողությունների արտասովոր արդյունքն այն է, որ իրարից անկախ կողմերը ստանում են միևնույն **K** արժեքը, որը օգտագործվում է որպես գաղտնի բանալի, քանի որ չարագործը կկանգնի անլուծելի խնդրի առջև՝ խելամիտ ժամանակում հաշվարկել a և b փակ բանալիները՝ ունենալով բաց կապուղիներով փոխանցվող A և B բաց բանալիները, եթե a, b և p թվերը տրված են բավականաչափ մեծ, կարելի է ասել անհնար է:

K գաղտնի բանալու արժեքի որոշման ալգորիթմի ընդհարացված բլոկ-սխեման պատկերված է նկ. 4.6.-ում: Գաղտնագրման բանալու արժեքի որոշման սարքի կառուցվածքային սխեման պատկերված է նկար 4.7.-ում:



Նկ. 4.6. K գաղտնի բանալու արժեքի որոշման ալգրիթմի բլոկ-սխեման

Ալգորիթմի բլոկ-սխեմայում գործողությունների հաջորդականությունը կատարում է նախաձեռնող կողմը, իսկ փակագծերում նշված են երկրորդ կողմի կատարվող գործողությունները[97]:

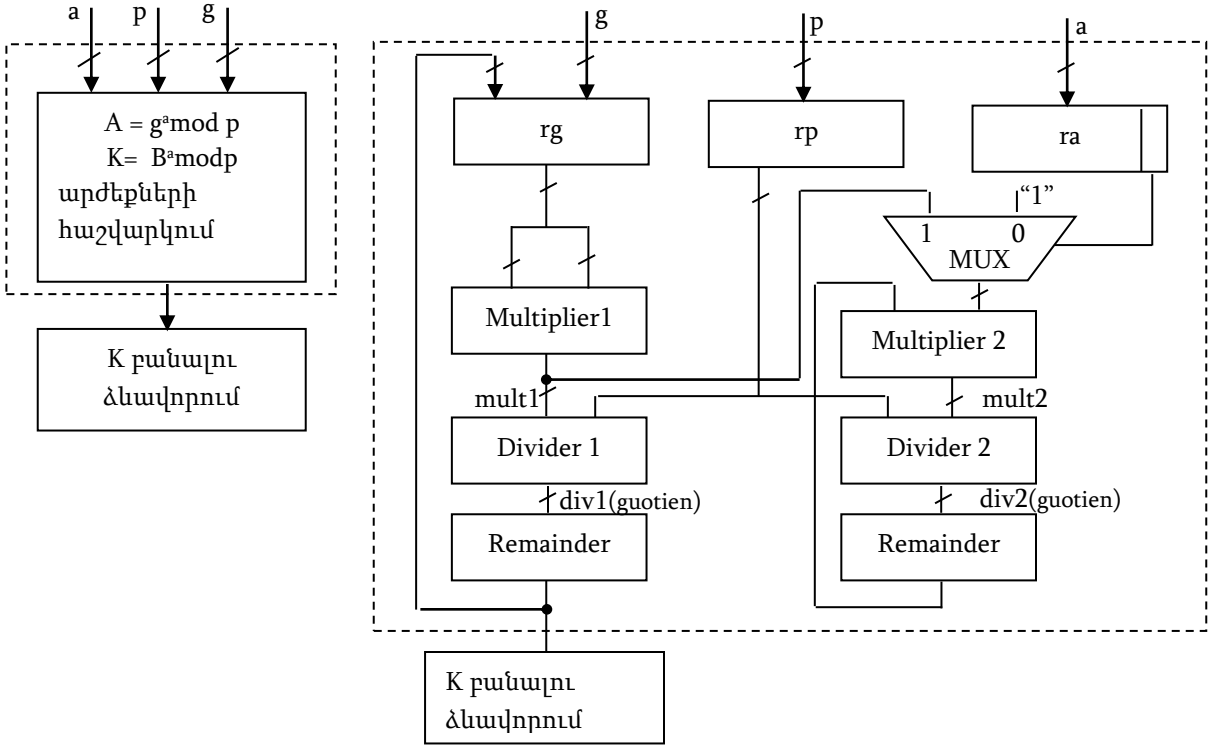


Նկ. 4.7. Գաղտնի բանալու արժեքի որոշման սարքի կառուցվածքային սխեման

Սարքի մուտքային տվյալները p,g և a(b) թվերն են: Սարքի աշխատանքի հետևանքով ելքային տվյալը ձևավորվում է K գաղտնի բանալու տեսքով:

rp ռեգիստրը p թվի ռեգիստրն է, rg ռեգիստրը g թվի ռեգիստրն է: Պարզ թվերի վեկտորից կարելի է ընտրել a և b փակ բանալիների արժեքները: MUX1 և MUX2 մուլտիպլեքսորներն օգտագործվում են նույն սարքի միջոցով կողմերի փոխանակման A և B բաց բանալիների որոշման համար (եթե հասցեական մուտքը «0» է) և K=K1=K2 կողմերի գաղտնի բանալիները որոշելու համար (եքե հասցեական մուտքը «1» է): Ինֆորմացայի գաղտնագրման համար K գաղտնի բանալու ստեղծումը կարելի է համարել ավարտված: Սարքը, որն աշխատում է A և B բաց և K գաղտնի բանալիների որոշման համար, իրականում պետք է կատարի մեծ թվերի աստիճան բարձրացնելու

գործողություններ: FSM-ը տվյալ սարքի աշխատանքը ղեկավարող ավտոմատի կառուցվածքն է, որի Y1-Y6 ելքային ազդանշանները կառավարում են օպերացիոն սարքը: FSM-ի մուտքային տվյալը կարող է հանդիսանալ գաղտնի բանալու գեներացման գործողության մեկնարկի (start) ազդանշանը: Ավանդական ճանապարհով $a^b \pmod n$ հաշվելու համար նախ պետք է a –ն բարձրացնել b-ի աստիճան, ապա ստացված շատ մեծ արդյունքը հաշվարկել ըստ mod n–ի: Քանի որ Դիֆֆի-Հելլմանի ալգորթմով գաղտնի բանալու հաշվարկման սարքի անխոցելիությունը (защищенность) ապահովելու համար առաջարկվում են a և b փակ բանալիները, ինչպես նաև p թիվը ընտրել շատ մեծ միջակայքում, ապա կարելի է պատկերացնել թե որքան շատ ժամանակ և մեծ կարգայնությամբ սարքեր (ռեգիստրներ, հաշվիչներ, գումարիչներ և այլն) կպահանջվեն ավանդական ճանապարհով $A = g^a \bmod p$ և $B = g^b \bmod p$ բաց բալանիների և K գաղտնի բանալին հաշվարկելու համար: p,g A, և B արժեքները փոխանակվում են բաց կապուղիներով և կարելի է ասել սեանսային (сеансовые), այսինքն՝ կարող են ժամանակ առ ժամանակ թարմացվել: Մեկ կողմի գաղտնի բանալու ձևավորող մոդուլյար էքսպոնենտում կատարող սարքը կունենա հետևյալ տեսքը (նկ. 4.8.):



Նկ. 4.8. Մոդուլյար էքսպոնենտում կատարող սարքի կառուցվածքային սխեման

Առաջարկվում է թիվը (mod p) հաշվել ոչ թե աստիճան բարձրացված արդյունքի համար, այլ կատարել էքսպոնետում ըստ մոդուլի՝ յուրաքանչյուր բազմապատկման գործողությունից հետո հաշվել տվյալ արտադրյալը (mod p), ապա ավելի փոքր մնացորդն օգտագործել հաջորդ արտադրյալը ձևավորելու համար:

Ստացված թվի mod p հաշվարկելու գործընթացը առավել արագագործ դարձնելու նպատակով առաջարկվում է բաժանման գործողությունը կատարել արդյունքի ձևավորման փոխանակման ժամանակի մեթոդով, ուր բաժանարարի նորմալացումը n կարգանի թվի համար կարելի է արագացնել մինչև 8 անգամ: Տվյալների գաղտնագրման պրոցեսը կարելի է իրականացնել և ծրագրային, և ապարատային: Ապարատային իրագործումն ունի ավելի մեծ արժեք, սակայն դրան հատուկ են նաև առավելություններ.

- ապարատային կոդավորումն ունի մեծ արագագործություն: Գաղտնագրման ալգորիթմները բաղկացած են մեծ քանակի բարդ գործողություններից, որոնք կատարվում են բաց տեքստի բիթերի նկատմամբ: Հատուկ (մասնագիտացված) սարքավորումները կատարում են այս գործողություններից ավելի արագ:
- սարքավորումները ֆիզիկապես պաշտպանել արտաքին ներխուժումներից ավելի հեշտ է: Դրանց կարելի է տեղակայել հատուկ կոնտեյներներում, որոնք գործառության սխեմաները պաշտպանում են փոփոխություններից:
- գաղտնագրման սարքը հեշտ է տեղակայել այն սարքերի վրա (հեռախոսներ, ֆաքսեր, մոդեմներ և այլն), որտեղ քոմփյութերները լրիվ ավելորդ են:
- ըստ Դիֆֆի-Հելլմանի ալգորիթմի, գաղտնագրման և վերծանման գործողություններն ունեն $a^b \bmod n$ ընդհանրացված տեսքը :
- գաղտնագրման և վերծանման գործողությունների կատարումը արագացնելու համար, ի տարբերություն աստիճան բարձրացնելու գործողության ավանդական ալգորիթմի, նպատակահարմար է օգտագործել աստիճան բարձրացնելու գործողությունը ըստ mod n –ի:
- մոդուլով բաժանման համար նպատակահարմար է օգտագործել արդյունքի ձևավորման փոփոխական ժամանակով բաժանման ալգորիթմը[97]:

- Verilog լեզվով RTL-նկարագրման համար յուրաքանչյուր բլոկը նկարագրվում է առանձին always պրոցեդուրային օպերատորի միջոցով, իսկ եթե այն կոմբինացիոն սխեմա է, կարող է նկարագրվել նաև assign օպերատորի կիրառմամբ:

Արդյունքում ստացվում է սարքի RTL-նկարագրումը, որի հիմքի վրա կարելի է սինթեզել սխեման:

4.3. Մշակված մեթոդների կիրառումը ինֆորմացիայի հսկման և արատորոշման համակարգերում

Ցանկացած տեխնիկական համակարգի կամ սարքի, այդ թվում համակարգչի հսկում ասելով հասկանում ենք նրանում առկա անսարքությունների (սխալների) փաստի հայտնաբերումը, իսկ արատորոշում ասելով՝ անսարքության տեղի որոշումը: Հսկումը և արատորոշումը կատարվում են լրացուցիչ ինֆորմացիայի կիրառմամբ, որն իրականացվում է կամ ապարատային եղանակով ավելցուկային սարքավորում ներդնելու, կամ ծրագրային, թեստային եղանակով լրացուցիչ ժամանակ ծախսելու միջոցով: Առաջինն արագագործ է, սակայն՝ թանկ, քանի որ լրացուցիչ ֆինանսական ներդրում է պահանջում, իսկ երկրորդը դանդաղագործ է, սակայն էժան: Համակարգիչներում երկու եղանակներն էլ լայնորեն կիրառվում են: Ապարատայինը նպատակահարմար է օգտագործել հիշասարքերում ինֆորմացիայի պահպանման, գրանցման և ընթերցման, համակարգչի տարբեր հանգույցների միջև ինֆորմացիայի փոխանակման, թվաբանական գործողությունների հսկման և արատորոշման համար, իսկ թեստայինը՝ տրամաբանական սխեմաների, հանգույցների հսկման և արատորոշման (ինքնաստուգման) կանխարգելիչ միջոցառումների համար [3, 35]: Տարբերում են ըստ մոդուլի հսկման երկու եղանակ՝

- թվային հսկում,
- թվանշանային հսկում:

Ընդհանուր դեպքում ra հսկման կոդը, որպես A թվի ֆունկցիա կարելի է ներկայացնել հետևյալ տեսքով.

$$f(A) = r_a \bmod m: \tag{4.18}$$

Ըստ մոդուլի հսկման դեպքում կարելի է դիտարկել այդ ֆունկցիայի երկու արժեք: Առաջինը, երբ f(A)=A և ra հսկող կոդ, կապված A թվի հետ, կարելի է ներկայացնել հետևյալ ձևով.

$$A \equiv r_a \bmod m: \tag{4.19}$$

Այսինքն այս դեպքում մենք ունենք թվային հսկում: Այս հսկման ժամանակ թվի հսկման կոդ է հանդիսանում մնացորդը, որը ստացվում է A թիվը modq-ի բաժանելիս:

Թվանշանային հսկման դեպքում հսկման կոդը որոշվում է հսկվող թվի թվանշանների նկատմամբ իրականացվող գործողություններով, մասնավորապես, թվանշանների գումարով և այս դեպքում

$$f_2(A) = \sum_{i=1}^{n} a_i \bmod m, \tag{4.20}$$

որտեղ a_i -ն A թվի թվանշաններն են:

Ի տարբերություն թվային հսկման՝ այս հսկումը լայն կիրառություն է գտել: Այս դեպքում հսկման կոդ է հանդիսանում մասնավորապես A թվի թվանշանների գումարը mod m-ի բաժենելիս ստացված մնացորդը: Քանի որ մեծ թվի թվանշանների գումարի կարգայնությունը բավականին փոքր է այդ թվի կարգայնությունից, մնացորդի հաշվումը պարզեցվում է, չնայած լրացուցիչ պահանջվում է թվի թվանշանների գումարի հաշվարկման գործողություն[35]:

Դիտարկենք ըստ կամայական մոդուլի մնացորդի ձևավորման գործընթացը: Հնարավոր է հսկման q մոդուլի և հաշվարկման համակարգի p հիմքի հարաբերության հետևյալ երեք դեպքերը՝

1. <u>m>p, 2. m=p, 3. m<p:</u>

Դիտարկենք յուրաքանչյուր դասի համար հսկման եղանակը: Առաջինում, երբ $q=p^m-1$, նախ A հսկվող թիվը p-ական հաշվարկման համակարգից փոխադրվում է $P=m^m$ համակարգ՝ ներկայացվելով հետևյալ տեսքով.

$$A = \sum_{i=1}^{n} b_i, \tag{4.21}$$

որտեղ b_i (i = 1… k) – P-ական համակարգի թվանշաններն են k=n/m, ըստ որում, եթե n կարգերը չեն բավարարում m կարգերից կազմված լրիվ խմբավորում կատարել, ապա ձախից ավելացվում են անհրաժեշտ քանակով 0-ներ: Այնուհետև այդ տեսքով ներկայացված թվի թվանշանները գումարվում են ըստ մոդուլ m-ի, որի արդյունքը

$$r_a = \sum_{i=1}^{n} b_i \bmod m \qquad (4.22)$$

ներկայացնում է A թվի հսկման կոդը ըստ մոդուլ m-ի:

Առաջինում, նախ գումարվում են երկու թվանշաններ, արդյունքին գումարվում է երրորդ թվանշանը և այսպես շարունակ: Երկրորդում՝ բոլոր թվանշանները բաժանվում են զույգերի և գումարվում միաժամանակ, այնուհետև ստացված արդյունքները նորից զույգերով միաժամանակ գումարվում են: Առաջինում փոքր է լրացուցիչ սարքավորման ծախսը, սակայն փոքր է նաև արագագործությունը: Երբ $m=p^m+1$, նախ դարձյալ A հսկվող թիվը p-ական հաշվարկման համակարգից փոխադրվում է $P=p^m$ համակարգ: Այնուհետև P-ական թվանշանները աջից ձախ, 1 համարից սկսած, համարակալվում են: Առանձին-առանձին կենտ և զույգ համարներով թվանշանները գումարվում են ըստ մոդուլ m-ի, և որոշվում է դրանց տարբերությունը: Եթե այն դրական է, ապա ներկայացնում է հսկման կոդը, իսկ եթե բացասական է՝ հսկման կոդ է ներկայացնում նրա լրացումը մինչև m:

Կատարելով հետևյալ նշանակումները՝

Sկ – կենտ համարներով թվանշանների գումարը ըստ մոդուլ m-ի,

Sզ – զույգ համարներով թվանշանների գումարը ըստ մոդուլ m-ի,

Sտ=Sկ-Sզ կստանանք՝

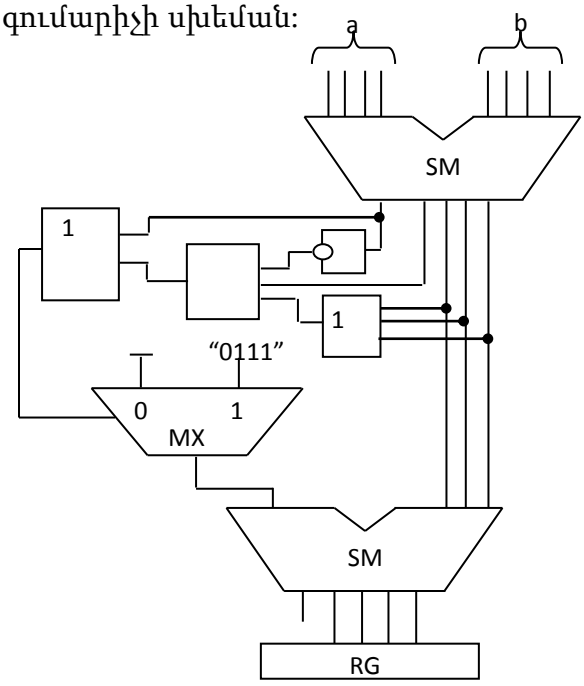
$$r_a = \begin{cases} S_տ, & \text{եթե } S_տ \geq 0 \\ q+S_տ, & \text{եթե } S_տ \leq 0 \end{cases} \qquad (4.23)$$

Հսկման կոդի ձևավորման կարևոր գործընթացներից է տրված մոդուլով գումարման իրականացումը: Լայն տարածում ունեն $m = p^k + 1$ և $m = 2^k - 1$ մոդուլներով կոմբինացիոն կամ կուտակող տիպի գումարիչները: Դիտարկենք $m=2^k+1$ մոդուլով գումարումը, որն իրականացվում է հետևյալ կերպ.

1. Կիրառվում է (n+1) կարգանի գումարիչ:

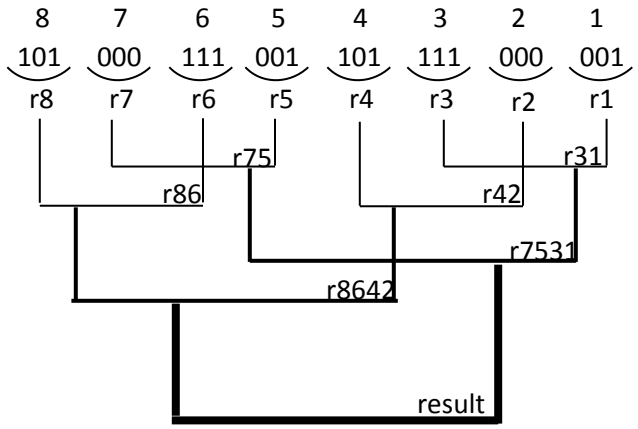
2. Բարձր կարգից փոխանցման դեպքում գումարիչին գումարվում է $2^{k+1} - m$ արժեք:
3. Ձևավորվում է փոխանցում և գումարիչի պարունակությանը գումարվում է $2^{k+1} - m$ արժեք նաև գումարման հետևյալ x-ի արդյունքի դեպքում, եթե $m<x<2^{n+1}$:
4. Փոխանցումները հետադարձ կապով ցիկլիկ փոխանցման շղթայով գումարիչի մուտքին են հաղորդվում τ հապաղմամբ:
5. Գումարման զ արդյունքի դեպքում գումարիչից դուրս է բերվում 0 արժեք:

Սույն ատենախոսական աշխատանքի շրջանակներում ուսումնասիրվել է զուգահեռ եղանակով ըստ մոդուլ 9-ի ապարատային հսկումը: Նկ. 4.9-ում առաջարկվում է համապատասխան կոմբինացիոն գումարիչի սխեման:



Նկ. 4.9. Զուգահեռ եղանակով ըստ մոդուլ 9-ի ապարատային հսկման կառուցվածքային սխեմա

Առաջադրվում է նաև ըստ մոդուլ 9-ի զուգահեռ եղանակով ապարատային հսկման իրականացումը և նկարագրումը Verilog լեզվով:



Նկ. 4.10. Զուգահեռ եղանակով ըստ մոդուլ 9-ի ապարատային հսկում

Թիվը ներկայացնենք 2-ական համակարգում համապատասխանաբար 24 կարգով (բիթով): Քանի որ $9=2^3+1$, հետևաբար տրված թիվը բաժանում ենք եռյակների(բերում ենք 8-ական համակարգ) և համարակալում՝ աջից-ձախ, սկսած 1-ից: (նկ. 4.10)

Ծրագրային կոդում վերը նշված եռյակները իրենց համարներին(ինդեքսներին) համապատասխան վերագրվում են $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8$ օպերանդներին: Կենտ ինդեքսներով եռյակները գումարվում են ըստ մոդուլ 9-ի՝ զուգահեռ եղանակով , նույն սկզբունքով գումարվում են զույգ ինդեքսներով եռյակները: Արդյունքները վերագրվում են հետևյալ օպերանդներին՝

$$r_{31} = r_3 + r_1, \qquad r_{42} = r_4 + r_2,$$

$$r_{75} = r_7 + r_5, \qquad r_{86} = r_8 + r_6:$$

Հաջորդ փուլում ըստ մոդուլ 9-ի գումարվում են $r_{31}, r_{75}, r_{42}, r_{86}$ քառյակները և արդյունքները վերագրվում են հետևյալ օպերանդներին՝ $r_{31} = r_{7531} + r_{8642}$:

$S_կ = r_{7531} = r_{75} + r_{31}$, $S_զ = r_{8642} = r_{86} + r_{42}$:

Ծրագրային հատվածում մնացորդը նշանակենք՝ result , այն որոշելու համար կատարում ենք հետևյալ գործողությունը ՝

$$result = S_տ = S_կ - S_զ$$

$$result = \begin{cases} S_տ, & \text{եթե } S_տ \geq 0 \\ q + S_տ, & \text{եթե } S_տ \leq 0 \end{cases}$$

Հսկման գործողության կատարման համար որպես մոդուլյար գումարիչ առաջարկվում է կիրառել կամ LUT աղյուսակների միջոցով կառուցված մոդուլյար գումարիչներ, կամ առանց LUT աղյուսակի մոդուլյար գումարիչ, որի կառուցման մեթոդը բերված է գլուխ 2-ում, և Verilog նկարագրումը՝ ատենախոսության հավելվածում:

Թեև ժամանակակից օպերացիոն համակարգերում ապարատային հսկումն իրականացվում է ծրագրային ընդհատումների միջոցով, սակայն ԲՖ բլոկների կիրառումը հնարավորություն ընձեռեց մասնագիտացված սարքերում օգտագործել թվային հսկման ապարատային իրագործման մեթոդները, ինչը պահանջում է փոքր ծախսեր և ապահովում մեծ արագագործություն:

Գլուխ 4-ի եզրահանգումները

1. Ատենախոսությունում որպես մշակված, նախագծված և հետազոտված սարքերի կիրառման ոլորտներ դիտարկվել են գաղտնագրման և թվային հսկման համակարգերը:
2. Գլուխ 2-ում նախագծված մոդուլյար բազմապատկիչների և մոդուլով աստիճան բարձրացնելու սարքերի կիրառմամբ նախագծվել են RSA ալգորիթմով գաղտնագրման սարքի և Դիֆֆի-Հելլմանի բանալիների գեներացման սարքի կառուցվածքները, նախագծվել սարքավորումների նկարագրման բարձր մակարդակի Verilog լեզվով, և սինթեզումից հետո ծրագրավորվել FPGA-ի օգնությամբ:

   Նշված սարքրերում օգտագործվել են էյլերի ֆունկցիայի կիրառմամբ մոդուլով աստիճան բարձրացնելու սարքեր, քանի որ տվյալ ալգորիթմներում մոդուլի արժեքը պետք է լինի պարզ թիվ, մոդուլով բաժանման համար նպատակահարմար է օգտագործել արդյունքի ձևավորման փոփոխական ժամանակով բաժանման ալգորիթմը[4]: Արդյունքում՝ ստացվել է սարքի RTL-նկարագրումը, որի հիմքի վրա սինթեզվել է սխեման:
3. Թվային ապարատային հսկման համակարգերում կիրառվել են գլուխ 2-ում 2^k+1 և 2^k-1 մոդուլներով նախագծված մոդուլյար բազմապատկիչներ և 2^k+1 և 2^k-1 մոդուլներով մոդուլյար գումարիչներ: Թվային սարքի աշխատանքը նկարագրվել է Verilog լեզվով, և սինթեզումից հետո ծրագրավորվել FPGA-ի օգնությամբ:

4. Մշակվել է պարզ թվերի գեներացման սարքը, որի Verilog մոդուլը հայցվում է (կանչվում է) RSAալգորիթմի պարզ թվերի գեներացման համար:
5. Մշակվել և Verilog լեզվով նկարագրվել է պարզ մոդուլների արժեքների համար պարզունակ արմատի որոշման սարքը, որը նույնպես սինթեզումից հետո ծրագրավորվել FPGA-ի օգնությամբ:
6. Մշակված միջոցների հիման վրա կազմվել են “Ծրագրավորվող տրամաբանական սարքեր” մեթոդական ցուցումները, որոնք ներդրվել են “Քոմփյութերային համակարգեր և ցանցեր” ամբիոնում բակալավրի կրթական ծրագրով դասավանդվող “Թվային սարքերի նախագծման միջոցները” առարկայի շրջանակներում լաբորատոր աշխատանքների իրականացման գործընթացում:

ԵԶՐԱԿԱՑՈՒԹՅՈՒՆ

Ատենախոսությունում հետազոտությունների հիման վրա հիմնավորվել է մոդուլյար թվաբանության բլոկում մասնագիտացված թվային սարքերի նախագծման նոր մեթոդների մշակման անհրաժեշտությունը: Հետազոտության հիմնական արդյունքներն են.

1. Մշակվել են մոդուլյար գումարիչների կառուցման տարբեր մեթոդներ և գնահատվել է այդ մեթոդների կիրառման արդյունավետությունը՝ կախված մուտքային թվերի և մոդուլի արժեքի կարգայնությունից, որի արդյունքում կրճատվել է FPGA օգտագործվող ռեսուրսների 8-10%
2. Մշակվել են մոդուլյար բազմապատկիչների կառուցման տարբեր մեթոդներ և գնահատվել է այդ մեթոդների կիրառման արդյունավետությունը՝ կախված մուտքային թվերի և մոդուլի արժեքի կարգայնությունից, ինչի արդյունքում կարելի է կրճատել FPGA ներառված ռեսուրսների 7-12%:
3. Մշակվել են մոդուլով աստիճան բարձրացում իրականացնող սարքերի կառուցման տարբեր մեթոդներ և գնահատվել է այդ մեթոդների կիրառման արդյունավետությունը՝ կախված մուտքային թվերի և մոդուլի արժեքի կարգայնությունից, ինչի արդյունքում կարելի է կրճատել FPGA ռեսուրսների ≈15%:
4. Գնահատվել են մոդուլյար բլոկում թվաբանական գործողություններ կատարող սարքերի արագագործությունը և բարդությունները FPGA-ի ռեսուրսների օգտագործման տեսակետից:
5. Հետազոտությունների արդյունքում արվել են առաջարկներ մոդուլյար սարքերի տարբեր մեթոդների նախագծման առավելությունների վերաբերյալ
6. Մշակվել է մոդուլյար թվաբանության բլոկի համար ավտոմատացված նախագծման Modsystem համակարգը, որում միավորվել են մշակված մեթոդները և, որի օգնությամբ կարելի է կատարել մոդուլյար թվաբանության բլոկում մասնագիտացված թվային սարքերի նախագծման մեթոդի ընտրություն:

Գրականության ցանկ

1. Թվային սարքերի նախագծման հիմունքները Verilog HDL լեզվի կիրառմամբ: Հաշվողական համակարգերի ճարտարապետություն առարկայի կուրսային աշխատանքի մեթոդական ցուցումներ/ Ա.Թումանյան, Խ.Շարոյան, Ա.Համազասպյան, Ա.Սաղաթելյան և ուր.;/ ՀՊՃՀ.- Եր.: «Ճարտարագետ», 2011.- 96 էջ:
2. Ծրագրավորվող տրամաբանական սարքեր: «Թվային սարքերի նախագծման միջոցներ» առարկայի դասընթացի մեթոդական ցուցումներ/ Ա.Թումանյան, Ա.Համազասպյան, Ա.Սաղաթելյան և ուր.;/ ՀՊՃՀ.- Եր.: «Ճարտարագետ», 2014.- 90 էջ:
3. Պետրոսյան Ռ.Պ. Հաշվողական համակարգերի և համալիրների շահագործում: Կուրսային նախագծի մեթոդական ցուցումներ:-Երևան.-«Ճարտարագետ» ՀՊՃՀ հրատ., -2010.- 61 էջ:
4. Սաղաթելյան Ա.Կ. Արդյունքի ձևավորման փոփոխական ժամանակով ամբողջ թվերի բաժանման արագագործ սարքի կառուցվածքը // ՀՊՃՀ-ի Լրաբեր. Գիտական և մեթոդական հոդվածների ժողովածու.- Երևան, 2010.- Հատոր 2, № 1.- էջ 236-239:
5. Սաղաթելյան Ա.Կ., Թումանյան Ա.Կ., Քամալյան Ա.Գ. RSA ալգորիթմի ապարատային իրագործումը // ՀՊՃՀ-ի Լրաբեր. Գիտական հոդվածների ժողովածու. Մաս 1.- Երևան: «Ճարտարագետ»., 2013. - էջ 179-185:
6. Սաղաթելյան Ա.Կ. Մնացորդային դասերի համակարգում $m=2^k$ մոդուլով բազմապատկման սարքերի նախագծում // Հայաստանի գիտությունների ազգային ակադեմիայի և Հայաստանի ազգային պոլիտեխնիկական համալսարանի տեղեկագիր. Տեխնիկական գիտությունների սերիա. - 2015.- Հատոր 68, № 1. – էջ 44-50:
7. Կիրակոսյան Գ.Տ., Սաղաթելյան Ա.Կ. Մնացորդային դասերի համակարգում ինդեքսային մեթոդով բազմապատկման սարքերի նախագծումը //ՀՊՃՀ-ի Լրաբեր. Գիտական հոդվածների ժողովածու. Մաս 1.- Երևան: Ճարտարագետ, 2014. - էջ 110-115:
8. Акушский И.Я., Юдицкий Д.И. Машинная арифметика в остаточных классах. - М.: Советское радио, 1968. – 440 с.
9. Амербаев В.М. Теоретические основы машинной арифметики. - Алма-Ата: Наука, 1976. - 324с.
10. Амербаев В.М., Стемпковский А.Л., Широ Г.Э. Быстродействующий согласованный фильтр, построенный по модулярному принципу // Информационные технологии. -2004.- №9.- С. 5-12.
11. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. - М.: Мир, 1979.- 536 с.
12. Бухтев А. Создайте собственный маршрут проектирования ПЛИС в системе Active-

nDL //ЭЛЕКТРОНИКА: Наука, Технология, Бизнес.- 2005. -№ 3.-С. 64-66.

13. Виноградов И.М. Основы теории чисел. – М.: Наука, Главная редакция физико-математической литературы, 1981 - 176с.
14. Евстигнеев В.Е. Недвоичная машинная арифметика и специализированные процессоры / Под ред. И.Я. Акушского. - М.: МИФИ «Сервис», 1992. -267 с.
15. Исследование методов проектирования и разработка программных средств синтеза быстродействующих арифметических устройств: Отчет о НИР (заключ.), шифр «Вега-а. К-2004» / ИППМ РАН; рук. А.Л. Стемпковский. – М., 2004. - № ГР 01200410528. - Инв. № 02200406453.
16. Исследование и разработка методов аппаратной реализации базовых элементов и основных вычислительных процедур для специализированных устройств цифровой обработки сигналов с применением нетрадиционной арифметики: Отчет о НИР (заключ.), шифр «Вега-К-2004» / ИППМРАН; рук. А.Л. Стемпковский. – М., 2005. - № ГР01200410531. - Инв. № 02200601688.
17. Исследование методов проектирования и разработка программных средств синтеза быстродействующих арифметических устройств: Отчет о НИР (заключ.), шифр «Вега-0 -Ст-2005» / ИППМ РАН; рук. А.Л. Стемпковский. – М., 2005. - № ГР 01200606057. - Инв. № 02200603585.
18. Исследование методов проектирования и разработка программных средств синтеза быстродействующих арифметических устройств: Отчет о НИР (заключ.) шифр «Вега-0-Ст-2006» / ИППМ РАН; рук. А.Л. Стемпковский. – М., 2006. - № ГР 01200606060. – Инв. №02200701842.
19. Калашников В.С. Основные виды архитектур модулярных сумматоров для двух 160 битных операндов // Микроэлектроника и информатика.- Тезисы докладов одиннадцатой всероссийской межвузовской конференции студентов и аспирантов.- М.: МИЭТ, 2004.- С.219.
20. Калашников В.С. Принципы построения двоичных и модулярных мультиоперандных Сумматоров// Микроэлектроника и информатика.- Тезисы докладов двенадцатой всероссийской межвузовской конференции студентов и аспирантов.- М.:МИЭТ, 2005.- С.196.
21. Калашников В.С., Ласточкин О.В., Семенов М.Ю. Лабораторный практикум по курсу «Основы логического синтеза средствами САПР Synopsys с использованием Verilog

HDL». - М.: МИЭТ, 2004. – 88 с.

22. Кнут Д. Искусство программирования. Том 2: Получисленные алгоритмы. - М.: Издательский дом «Вильямс», 2001. – 832 с.
23. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. - М.: МЦПМО, 2001.- 960 с.
24. Корнилов А.И., Семенов М.Ю., Ласточкин О.В., Калашников В.С. Применение современных методов проектирования при реализации модулярных вычислительных процедур: Сборник/ Институт проблем проектирования в микроэлектронике РАН.- 2011.- 15/3а.- С. 158-164,
25. Корнилов А.И., Семенов М.Ю. Преобразователь из модулярного представления в двоичную систему счисления на основе алгоритма с предварительной обработкой данных// Известия вузов. Электроника. - 2003. - № 3. - С. 54-58.
26. Корнилов А.М., Исаева Т.Ю., Семенов М.Ю. Методы логического синтеза сумматоров с ускоренным переносом по модулю ($2^{к}$–l) на основе BDD-технологии.// Известия вузов. Электроника. - 2004. *–Т 3.*- С. 54-60.
27. Корнилов А.И., Семенов М.Ю., Калашников В.С., Ласточкин О.В. Методология проектирования специализированных вычислителей на основе автоматизированной генерации технологически независимых IP-блоков // Проблемы разработки перспективных микроэлектронных систем: Сборник научных трудов/ Под общ. ред. А.Л. Стемпковского. - М.: ИППМ, 2005. - С.452-487.
28. Корнилов А.И., Семенов М.Ю., Калашников В.С., Ласточкин О.В. Принципы построения специализированных вычислителей с применением модулярной арифметики // Проблемы разработки перспективных микроэлектронных систем: Сборник научных трудов/ Под общ. ред. А.Л. Стемпковского. - М.: ИППМ РАП, 2005. - С. 346-351.
29. Корнилов А.И., Семенов М.Ю., Калашников В.С., Ласточкин О.В. Применение современных методов проектирования при реализации модулярных вычислительных процедур// Сборник научных трудов: Юбилейной Международной научно-технической конференции (В рамках V Международной научно-технической конференции «Электроника и информатика – 2005») «50 лет модулярной арифметике».- М.: ОАО"Ангстрем", МИЭТ, 2006. - С. 369-383.
30. Корнилов А.И., Семенов М.Ю., Ласточкин О.В., Калашников В.С. Реализация специализированных быстродействующих вычислителей на основе нетрадиционных алгоритмов с применением 1Р-генераторов// Материалы V конференции

Международной научно-технической конференции «Электроника и информатика».- Часть 1. - М.: МИЭТ, 2005. - С. 192-193.

31. Корнилов А.И., Семенов М.Ю., Калашников В.С. Методы аппаратной оптимизации сумматоров для двух операндов в системе остаточных классов // Известия ВУЗов. Электроника. - 2004. - № 1. - С. 75-82.
32. Корнилов А.И., Семенов М.Ю., Ласточкин О.В. Принципы построения модулярных индексных умножителей // Известия ВУЗов. Электроника. - 2004. - № 2. - С. 48-55.
33. Корнилов А.И., Семенов М.Ю., Калашников В.С., Ласточкин О.В. Особенности построения умножителей по модулю ($2^к$ –l) // Известия ВУЗов. Электроника. - 2006. - № 1.- С. 55-59.
34. Кравченко В., Радченко Д. Виртуальное прототипирование для аппаратно-программной верификации СБИС // Электроника: Наука, Технология, Бизнес. - 2003. № 7/- С. 34-67.
35. Крэндалл Р., Померанс К. Простые числа: Криптографические и вычислительные аспекты/ Пер с англ.;/ Под ред. и с предисл. В.Н. Чубарикова.- М.: Книжный дом «ЛИБРОКОМ», 2011.- 664 с.
36. Ласточкин О.В. Особенности реализации умножителей в устройствах, построенных с применением принципов модулярной арифметики // Микроэлектроника и информатика: Тезисы докладов одиннадцатой всероссийской межвузовской конференция студентов и аспирантов.- М.: МИЭТ, 2004. - С. 219.
37. Ласточкин О.В. Принципы построения IP-блоков двоичньк и специализированных умножителей с применением языка Verilog HDL// Микроэлектроника и информатика: Тезисы докладов двенадцатой всероссийской межвузовской конференция студентов и аспирантов.- М.: МИЭТ, 2005. - С. 104.
38. Ласточкин О.В. Методология разработки IP-генераторов и их применение в общем маршруте проектирования СБИС// Микроэлектроника и информатика: Тезисы докладов тринадцатой всероссийской межвузовской конференция студентов и аспирантов.- М.: МИЭТ, 2006. - С.78..
39. Лобес М.В. Разработка методов и алгоритмов модулярных вычислений для задач большой алгоритмической сложности // Научно-технический вестник С-ПбГУ информ. Технологий.- СПб., 2011.- Выпуск 6 (70). – С. 67-72.

40. Лохов А., Рабоволюк А. Средства проектирования FPGA компании Mentor Graphics/ ЭЛЕКТРОНИКА: Наука, Технология, Бизнес. № 4/2004. -С. 60-62.

41. Ìàëîâ Ñ.Â., Ïîçäíÿêîâ Ñ.Í., Ðûáèí Ñ.Â. Îñíîâû äèñêðåòíîé ìàòåìàòèêè: Ó÷åá. ïîñîáèå.- ÑÏá.: Èçä-âî ÑÏáÃÝÒÓ "ËÝÒÈ", 2008.- 72 ñ.

42. Немудров В., Мартин Г. Системы на кристалле. Проектирование и развитие. - М.: - Техносфера, 2004. – 216 с.

43. Немудров В., Евтушенко Е., Сырцов И. Методология проектирования систем на кристалле: основные принципы, методы, программные средства // Электроника: Наука, Технология, Бизнес. - 2003. – *Т 6.* - С. 7-11.

44. Орлов С., Цилькер Б. Организация ЭВМ и систем: Учебник для вузов. 2-е изд.- СПб.: Питер, 2011.- 688 с.

45. Рабаи Ж., Чандракасан А., Николич Б. Цифровые интегральные схемы, 2-е изд. /Пер с англ.- М.: ООО «И.Д.Вильямс», 2007.- 912 с.

46. Сагателян А.К. Принципы построения модулярных сумматоров // Materiały IX Międzynarodowej naukowi praktycznej konuferencji, Wschodnie partnerstwo.- Techniczne nauki, Przemyśl Nauka i studia. 2013 Volume 35, - С. 41-45.

47. Сагателян А.К.,Туманян А.К., Построение схем умножения на основе комбинационных умножителей// Вестник ГИУА. Серия «Моделирование, Оптимизация, Управление».- 2011.- Выпуск 14, том 1.- С. 82-91.

48. Семенов М., Калашников В., Ласточкин О. Структура оптимизированных сумматоров, функционирующих в системе остаточных классов// Микроэлектроника и информатика: Тезисы докладов десятой всероссийской межвузовской конференция студентов и аспирантов.- М.: МИЭТ, 2006. - С. 79.

49. Семенов М. Особенности интегрального исполнения устройств цифровой обработки сигналов в модулярной арифметике// Микроэлектроника и информатика: Тезисы докладов одинадцатой всероссийской межвузовской конференция студентов и аспирантов.- М.: МИЭТ, 2004. - С. 238.

50. Семенов М., Калашников В., Ласточкин О. Применение аппарата модулярной арифметики для построения фильтра с конечной импульсной характеристикой // Известия ВУЗов. Электроника. - 2005. - № 3. - С. 46-50.

51. Стемпковский А., Корнилов А., Семенов М. Особенности реализации устройств цифровой обработки сигналов в интегральном исполнении с применением модулярной арифметики // Информационные технологии. - 2004. - № 2. - С. 2-9.

52. Стемпковский А., Семенов М. Основы логического синтеза средствами САПР Synopsys с использованием Verilog HDL: Учебное пособие. - М.: МИЭТ, 2005. -140 с.
53. Стемпковский А.Л., Корнилов А.И., Семенов М.Ю., Калашников В.С., Ласточкин О.В. Построение систем повышенной надежности на основе аппарата модулярной арифметики с применением современных методов и средств проектирования // Проблемы разработки перспективных микроэлектронных систем: Сб. научных трудов / Под общ. ред. А.Л. Стемпковского. - М.: ИППМ РАП, 2006. - С. 253-258.
54. Торгашов В.А. Система остаточных классов и надежность ЦВМ. - М.: Советское радио,1973.-120 с.
55. Abdel-Hamid A., Tahar S., El Mostapha Aboulhamid. IP Watermarking Techniques: Survey and Comparison // System-on-Chip for Real-Time Applications: Proceedings. The 3th IEEE International Workshop, 30 Jun. - 2 Jul., 2003.- Beijing,P. 60-65.
56. Ahmed Al Faresi. Oregon State University Corvallis, Analysis of Low-Power Elliptic Curve Cryptography Using Scaled Modular Arithmetic. Oregon 97331–4501. http://cs.ucsb.edu/~koc/cren/project/pp/alfaresi.pdf
57. Bayomi M.A., JuUien G.A. A VLSI Implementation of the Residue Adders// IEEE Trans, on Circuits and Systems. - Mar., 1987. - Vol.34, №3. - P. 284-288.
58. Bening L., Foster H. Principles of Verifiable RTL Design a functional coding style supporting verification processes in Verilog.- Kluwer Academic Publishers. Second Printing.- 2000.- 253p.
59. Bricaud P. IP Reuse Creation for System-on-a-Chip Design // Custom Integrated Circuits: Proceedings of the IEEE 16-19 May, Beijing.- 1999.- P.359-401.
60. Bricaud P., Antipolis S. VC Rating and Quality Metrics: Why Bother?// Quality Electronic 163 Design, 2002. Proceedings. International Symposium on 18-21 Mar. 2002. Pages: 257-260.
61. Bricaud P., Remond F. Set-Top box System-on-Chip Design Methodology// Electronics Systems and Software.- Feb.2003. Vol.1, issue 1.- P. 10-13.
62. Cardarilli G.C., Lojacono R., Martinelli G., Salerno M. Structurally Passive Digital Filters in Residue Number Systems// IEEE Trans, on Circuits and Systems.- February 1988. Vol.35, No.2.- P. 149-158.
63. Cardarilli G.C., Re M., Lojacono R. A new RNS FIR Filter Architecture// DSP-97: IEEE 13th International Conference on Digital Signal Processing, 2-4 Jul., Vol.2.- 1997.- P. 671-674.
64. Cardarilli G.C., Nannarelli A., Re M. Reducing Power Dissipation in FIR Filters using the

Residue Number System// Proc. of 43rd IEEE Midwest Symposium, on Circuits and Systems.- Aug. 2000.- P. 320-323.

65. Cardarilli G.C., Del Re A., Nannarelli A., Re M. Residue Number System Reconfigurable Datapath // ISCAS: IEEE International Symposixmi on Circuits and Systems.- May2002.-Vol. 11. P. 11-756-11-759.
66. Charbon E., Torunoglu I. On Intellectual Property Protection// Custom Integrated Circuits Con-fe-rence, CICC 2000: Proceedings of the IEEE, 21-24 May, 2000.- 2001 Pages: 517-523.
67. Chang S., Kim S.D. Reuse-based Methodology in Developing System-on-Chip// Software Engineering Research, Management and Applications, 4th International Conference, 9-11 Aug., 2006.- P.125-131.
68. Chiranth E, Chakravarthy H.V.A, Nagamohanareddy P, Umesh T.H, Chethan Kumar M. Implementation of RSA Cryptosystem Using Verilog. International Journal of Scientific & Engineering Research Vol. 2, issue 5, May-2011 1 ISSN 2229-5518. http://www.ijser.org/researchpaper/Implementation_of_RSA_Cryptosystem_Using_Verilog.pdf
69. Ch.Sri Nagendra, K.Sarath Babu. Implementation of Polynomial Modular Multiplication for RSA Public. International Journal of Advanced and Innovative Research ISSN: 2278-7844. Available, http://ijair.jctjournals.com Volume 1, Issue 2, July 2012.
70. Coussy P., Baganne A., Martin E. Platform-Based Design for Digital Signal Processing Systems: A Case Study of MPEG-2/JPEG2000 Encoder// International Conference Communications, Circuits and Systems and West Sino Expositions, IEEE, 29 Jun.- 1 Jul. 2002.-Vol. 2.- P.1361-1366.
71. Design Ware IP Family Reference Guide// Synopsys Inc., vyww.svnopsvs.com, Jun. 2006.- 456 p.
72. Development System Reference Guide// Xilinx, Inc. www.xilinx.com. Dec. 2005. – 454 p.
73. Efstathiou C., Vergos H., Nikolos D. Modified Booth Modulo 2 " - l Multipliers// IEEE Transactions on Computers.- Mar. 2004. -Vol.53, No.3.- P.370-374.
74. Eroy F. A Core-Based System-to-Silicon Design Methodology// Design & Test of Computers,164 IEEE. Oct.-Dec. 1997. Vol. 14, issue 4.- P.36-41.

75. Grobschadl J. The Chinese Remainder Theorem and its Application in a High-Speed RSA Crypto Chip // 16 Annual Conference Computer Security Application, ACSAC'OO.- Dec.2000. P.384-393.
76. Gupta R., Zorian Y. Introducing Core-Based System Design// Design & Test of Computers,IEEE. Oct.-Dec. 1997. Vol. 14, issue 4.- P.15-25.
77. HDL Designer Reference Manual // Mentor Graphics, Inc. Online documentation, www.mentor.com.
78. Hekmatpour A., Goodnow K., Shah H. Standards-Compliant IP-Based ASIC and SoC Design//SOC Conference: Proceedings. IEEE International, 25-28 Sep.- 2005. P.322-323.
79. Hiasat A. A. High-Speed and Reduced-Area Modular Adder Structures for RNS// IEEE Transactions on Computers, vol. 51, no. 1, Jan. 2002.
80. Hiasat A. New memoryless, mod $(2^k \pm 1)$ residue multiplier // Electrionic letters, 30 Jan., 1992.- Vol.28, No.3.- P.314-315.
81. Hoare R., Tung S. Combining Mentor Graphic's HDL designer FPGA flow with a reconfigurable system on a programmable chip, education opportunity or insanty?// Microelectronics Systems Education Proceedings, IEEE International Conference, 1-2 Jun., 2003.- P. 128-130.
82. Hunt M., Rowson J. Blocking in a system on a chip// Spectrum, IEEE.- Nov.1996. Vol.33, issue 11. - P.35-41.
83. International Technology Roadmap for Semiconductors (ITRS).- 2005. Available: http://www.public.itrs.net.
84. Jenkins W.K., JuUien G.A., Dimitrov V.S. Residue Arithmetic With Applications in Digital Signal Processing, 1999.- P. 114-116.
85. JuIlien G.A. Number Theoretic Techniques in Digital Signal Processing// Advances in Electronics and Electron Physics.-Academic Press Inc., 1991.- Vol. 80. P.131.
86. Kahng A.B., Lach J., Mangione-Smith W.H., Mantik S., Markov I., Potkonjak M., Tucker P.,Wang H., Wolfe G. Watermarking Techniques for Intellectual Property Protection// Design Automation Conference, 15-19 Jan. 1998. –P.776-781.
87. Keutzer K., Chinnery D. Closing the Gap Between ASIC and Custom: An ASIC Perspective //Deisgn Automation Conference, 5-9 Jun. 2000. P.637-642.

88. Keutzer K., Malik S., Newton A.R. From ASIC to ASIP: The Next Design Discontinuity// Computer Design: VLSI in Computers and Processors: Proceedings IEEE International Conference, 16-18 Sept..- 2002.- P.84-90.

89. Kim Y., Song Bang-Sup, Grosspietsch J., Gilling S. A Carry-Free 54x54 Multiplier Using Equivalent Bit Conversion Algorithm// IEEE Journal of Solid-State Circuits.-Oct. 2001.-Vol. 36, No. 10,- P.1538-1545.

90. Lakshmanan, Othman M., Mohamad Alauddin Mohd AH. High Performance Parallel Multiplier using Wallace-Booth Algorithm// Semiconductor Electronics:. Proceedings. IEEE International Conference , 12-21 Dec., 2002.-2003.-P.433-436.

91. Li Li, Gao M., Cheng Z., Zhang D., He S. A New Platform-Based Orthogonal SoC Design Methodology// ASIC: Proceedings. 5th International Conference, 21-24 Oct., 2003. - Vol. 1.-2004.-P.428-432.

92. Marcelo E. KAIHARA. A Hardware Algorithm for Modular Multiplication/Division Based on the Extended Euclidean Algorithm. IEICE TRANS. FUNDAMENTALS, VOL.E88–A, NO.12 DECEMBER 2005. http://search.ieice.org/bin/summary.php?id=e88 a_12_3610&category=

93. Martin G. Design Methodologies for System Level IP// Design, Automation and Test in Europe. Proceedings. 23-26 Feb.,1998.-1998.- P.286-289.

94. Martin G., Chang H. Tutorial 2 System-on-Chip Design// ASIC,2001. Proceedings. 4* International Conference.- 23-25 Oct., 2001.- P.12-17.

95. Narayan N., Newbould R., Carothers J., Rodriguez J., Holman T. IP Protection for VLSI Designs via Watermarking of Routers // ASIC/SOC Conference: Proceedings. 14* Annual IEEE International, 12-15 Sept., 2001.-2001.- P.406-410.

96. Rehan Shams , Fozia Hanif Khan2 and Mohammad. Cryptosystem an Implementation of RSA Using Verilog. VOL. 1, NO. 3, AUGUST 2013, 102–109 Available online at: www.ijcncs.org ISSN 2308-9830 Umair3http: // www.ijcncs.org/published/volume1/issue3/p5_1-3.pdf

97. Saghatelyan A.K. The Hardware Implementation of the Cryptographic Key Generation based on Diffie-Hellman algorithm // Proceedings of Engineering academy of Armenia. - 2013. – Vol. 10, № 4. - P. 760-763.

98. Qi H., Jiang Z., Wei J. IP Reusable Design Methodology// ASIC'2001. Proceedings. 4th International Conference.-23-25 Oct., 2001. Pages: 756-795.

99. https://en.wikipedia.org/wiki/Xilinx_ISE

100. http://zetan.org/63218-xilinx-ise-design-suite-133-all-platforms-x86x64-2011-eng-crack.html
101. https://www.altera.com/products/fpga/overview.html
102. http://www.ni.com/rioeval/
103. https://ru.wikipedia.org/wiki/Actel
104. http://geektimes.ru/hub/fpga/
105. http://ru.wikipedia.org/ wiki. Эвристический_алгоритм.
106. http://algolist.manual.ru/defence/well_known/elgamal.php
107. http://compress.ru/article.aspx?id=10110

Հավելված

Առկա և նախագծված մեթոդներով մոդուլյար գումարիչների, բազմապատկիչների և մոդուլով աստիճան բարձրացման գործողության սարքերի սինթեզման արդյունքների հաշվետվությունները:

Առաջին սկզբունքը՝ գումարումը $(a+b) \bmod m$ մեկ հատ մեծ $2^{2n} \times M$ չափով LUT- աղյուսակի միջոցով, որտեղ $M=\log_2 m$, իսկ n-ը a և b թվերի կարգայնությունն է: Կառուցվածքային սխեման ներկայացված է նկ. 1.6-ում: n=8, n=16 և n=32 կարգայնության a և b թվերի և m մոդուլի արժեքը [0: 255] միջակայքում (այսինքն m-ը 8 կարգանի) սինթեզման արդյունքները ներկայացված են աղ. 1-ում:

Աղյուսակ 1.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Օգտա գործ.	% հաշ-վարկ	Օգտա գործ.	% հաշ-վարկ	Օգտա գործ	% հաշ-վարկ	Ընդհա-մենը
1.	4 մուտքանի LUTs-ի քանակ	20	1%	46	1%	118	2%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	10	1%	30	1%	61	2%	4656
3.	Կապակցված տրամաբա-նություն պարունակող սեկցիաների (Slices) քանակ	10	100%	10	100%	10	100%	10
4.	Օգտագործված մուտք/ ելքերի (IOBs) քանակ	32	6%	48	26%	80	45%	190
5.	Արագագործություն	15 նվ		15 նվ		15 նվ		

n=8, n=16 և n=32 կարգայնության a և b թվերի և m մոդուլի արժեքը [256: 1023] միջակայքում (այսինքն m-ը 10 կարգանի) սինթեզման արդյունքները ներկայացված են աղ. 2-ում:

Աղյուսակ 2.

Սարքերի օգտագործման հաշվետվություն								
N N	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	65	1%	168	2%	440	5%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	35	1%	90	2%	228	5%	4656
3.	Կապակցված տրամաբա-նություն պարունակող սեկցիաների (Slices) քանակ	10	100%	10	100%	10	100%	10
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	36	18%	56	29%	87	45%	190
5.	Արագագործություն	15 նվ		15 նվ		15 նվ		

Երկրորդ սկզբունքը՝ նախական գումարում և գումարի արժեքի մուտքագրումը LUT-աղյուսակ, ինչը կփոքրացնի LUT-աղյուսակի չափերը մինչև $2^{n+1} \times M$: Առաջարկված տարբերակով մշակված սխեման բերված է նկ. 1.7-ում: n=8, n=16 և n=32 կարգայնության

a և b թվերի և m մոդուլի արժեքը [0: 255] միջակայքում (այսինքն m-ը 8 կարգանի) սինթեզման արդյունքները ներկայացված են աղ.3-ում:

Աղյուսակ 3.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
6.	4 մուտքանի LUTs-ի քանակ	20	1%	34	1%	58	1%	9312
7.	Օգտագործված սեկցիաների (Slices) քանակ	18	2%	22	1%	36	1%	4656
8.	Կապակցված տրամաբա-նություն պարունակող սեկցիաների (Slices) քանակ	10	100%	10	100%	10	100%	10
9.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	24	6%	48	26%	80	45%	190
10.	Արագագործություն	25 նվ		25 նվ		25 նվ		

n=8, n=16 և n=32 կարգայնության a և b թվերի և m մոդուլի արժեքը [256: 1023] միջակայքում (այսինքն m-ը 10 կարգանի) սինթեզման արդյունքները ներկայացված են աղ.4-ում:

Աղյուսակ 4.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	42	1%	168	2%	440	5%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	26	1%	90	2%	228	5%	4656
3.	Կապակցված տրամաբա-նություն պարունակող սեկցիաների (Slices) քանակ	10	100%	10	100%	10	100%	10
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	24	12%	52	27%	84	44%	190
5.	Արագագործություն	25 նվ		25 նվ		25 նվ		

Երրորդ սկզբունքը ՝ առանց LUT աղյուսակի օգտագործմամբ սխեմա է, որտեղ օգտագործ-վում է երկու գումարիչ: Այս տարբերակով մշակված սխեման բերված է նկ. 2.1-ում: n=8, n=16 և n=32 կարգայնության a և b թվերի և m մոդուլի արժեքը [0: 255] միջակայքում (այսինքն m-ը 8 կարգանի) սինթեզման արդյունքները ներկայացված են աղ. 5-ում:

Աղյուսակ 5.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	46	1%	84	1%	146	2%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	38	2%	72	2%	158	3%	4656
3.	Կապակցված տրամաբ. պարունակող սեկցիաների (Slices) քանակ	10	100%	10	100%	10	100%	10
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	24	12%	52	27%	92	48%	190
5.	Սեկցիաներում պարունակվող տրիգերների քանակ	40	0%	72	0%	126	0%	9312
6.	Արագագործություն	30 նվ		45 նվ		65 նվ		

n=8, n=16 և n=32 կարգայնության a և b թվերի և m մոդուլի արժեքը [256: 1023] միջակայքում (այսինքն m-ը 10 կարգանի) սինթեզման արդյունքները ներկայացված են աղ.6-ում:

Աղյուսակ 6.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	68	1%	100	2%	172	3%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	52	2%	88	2%	190	5%	4656
3.	Կապակցված տրամաբ. պարունակող սեկցիաների (Slices) քանակ	10	100%	10	100%	10	100%	10
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	26	12%	56	29%	112	58%	190
5.	Սեկցիաներում պարունակվող տրիգերների քանակ	48	0%	88	0%	126	2%	9312
6.	Արագագործություն	25 նվ		40 նվ		60 նվ		

Առաջին սկզբունքով (axb) mod m մոդուլյար բազմապատկում PROM (Programmable Read-Only-Memory) տիպի հիշող սարքի միջոցով: Սարքի պայմանական սխեման պատկերված է նկ. 2.4-ում: Նույն կարգայնության a և b թվերի համար և m մոդուլի արժեքը [0: 255] միջակայքում (այսինքն m-ը 8 կարգանի) սինթեզման արդյունքները ներկայացված են աղ.7-ում:

Աղյուսակ 7.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	74	1%	118	2%	216	3%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	42	2%	56	2%	66	2%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	55	1%	100	2%	163	2%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	24	13%	40	21%	72	38%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	1	4%	1	4%	1	4%	24
6.	Արագագործություն	15 նվ		15 նվ		15 նվ		

նույն կարգայնության a և b թվերի համար և m մոդուլի արժեքը [256: 1023] միջակայքում սինթեզման արդյունքները ներկայացված են աղ.8.-ում:

Աղյուսակ 8.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	94	1%	166	2%	302	4%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	66	2%	85	2%	113	3%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	71	1%	118	2%	202	3%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	24	13%	40	21%	72	38%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	1	4%	1	4%	1	4%	24
6.	Արագագործություն	15 նվ		15 նվ		15 նվ		

Երկրորդ սկզբունքով՝ (axb) mod m մոդուլյար բազմապատկում ինդեքսային (կամ դիսկրետ-լոգարիթմական) մեթոդով: Սարքի պայմանական սխեման պատկերված է նկար 1.8.-ում: n=8, n=16 և n=32 կարգայնության a և b թվերի և m մոդուլի արժեքը [0: 255] միջակայքում պարզ թվերի (այսինքն m-ը 8 կարգանի) սինթեզման արդյունքները ներկայացված են աղ.9.-ում:

Աղյուսակ 9.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	60	1%	90	1%	134	2%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	34	1%	46	1%	73	2%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	46	1%	98	1%	158	2%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	18	9%	18	9%	18	9%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	2	4%	2	4%	2	4%	24
6.	Արագագործություն	105 նվ		105 նվ		105 նվ		

n=8, n=16 և n=32 կարգայնության a և b թվերի և m մոդուլի արժեքը [256: 1023] միջակայքում պարզ թվերի (այսինքն m-ը 10 կարգանի) սինթեզման արդյունքները ներկայացված են աղ.10.-ում:

Աղյուսակ 10.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	64	1%	86	2%	162	4%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	34	1%	42	2%	80	3%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	52	1%	90	2%	134	3%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	24	12%	26	13%	32	14%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	2	4%	2	4%	2	4%	24
6.	Արագագործություն	115 նվ		115 նվ		115 նվ		

Երրորդ սկզբունքով՝ (axb) mod m հաշվարկը քառակուսիների օրենքի հիման վրա նախագծված բազմապատկիչների պայմանական սխեման բերված է նկ. 1.10-ում: n=8, n=16 և n=32 կարգայնության a և b թվերի և m մոդուլի արժեքը [0: 255] միջակայքում բոլոր թվերի (այսինքն m-ը 8 կարգանի) սինթեզման արդյունքները ներկայացված են աղ.11.-ում:

Աղյուսակ 11.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	56	1%	82	1%	105	2%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	32	1%	43	1%	66	2%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	50	1%	84	1%	99	1%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	20	11%	21	12%	24	13%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	2	4%	2	4%	2	4%	24
6.	Արագագործություն	45 նվ		45 նվ		45 նվ		

նույն կարգայնության a և b թվերի համար և m մոդուլի արժեքը [256: 1023] միջակայքում բոլոր թվերի սինթեզման արդյունքները ներկայացված են աղ.12.-ում:

Աղյուսակ 12.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	60	1%	86	2%	120	2%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	31	1%	42	2%	66	2%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	48	1%	90	2%	134	2%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	32	16%	41	21%	56	30%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	2	4%	2	4%	2	4%	24
6.	Արագագործություն	45 նվ		45 նվ		45 նվ		

Հորիզոնտական սկզբունքով՝ (axb) mod m հաշվարկը $m=2^k$ մոդուլով մոդուլային բազմապատկիչների պայմանական սխեման պատկերված է նկար 2.14.-ում: n=8, n=16 և n=32 կարգայնության a և b թվերի և $m=2^k$ մոդուլի արժեքը [0: 255] միջակայքում (այսինքն m-ը 8 կարգանի) սինթեզման արդյունքները ներկայացված են աղ. 13.-ում:

Աղյուսակ 13.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
7.	4 մուտքանի LUTs-ի քանակ	53	1%	94	1%	121	2%	9312
8.	Օգտագործված սեկցիաների (Slices) քանակ	32	1%	48	1%	72	2%	4656
9.	Սեկցիաներում պարունակվող տրիգերների քանակ	52	1%	88	1%	106	1%	9312
10.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	16	9%	52	28%	80	43%	190
11.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	2	4%	2	4%	2	4%	24
12.	Արագագործություն	40 նվ		40 նվ		40 նվ		

n=8, n=16 և n=32 կարգայնության a և b թվերի և $m=2^k$ մոդուլի արժեքը [256: 1023] միջակայքում (m-ը 10 կարգանի) սինթեզման արդյունքները (աղ.14)

Ինդեքսային մեթոդով (եթե k>3) $m=2^k$ մոդուլով մոդուլային բազմապատկման սարքի

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utiliza-tion	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	72	1%	106	2%	148	2%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	40	1%	62	2%	92	3%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	50	1%	110	2%	166	2%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	36	18%	52	28%	80	43%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	2	4%	2	4%	2	4%	24
6.	Արագագործություն	50 նվ		50 նվ		50 նվ		

պայմանական սխեման ներկայացված է նկ. 2.10-ում: Սարքի պայմանական սխեման պատկերված է նկար 2.8.-ում: n=8, n=16 և n=32 կարգայնության a և b թվերի և m մոդուլի արժեքը [0: 255] միջակայքում պարզ թվերի (այսինքն m-ը 8 կարգանի) սինթեզման արդյունքները ներկայացված են աղ.15-ում:

Աղյուսակ 15.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	148	2%	90	1%	134	2%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	78	2%	46	1%	73	2%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	66	2%	98	1%	158	2%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	32	16%	48	26%	64	34%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	3	12%	3	12%	3	12%	24
6.	Արագագործություն	25 նվ		25 նվ		25 նվ		

n=8, n=16 և n=32 կարգայնության a և b թվերի և m մոդուլի արժեքը [256: 1023] միջակայքում պարզ թվերի սինթեզման արդյունքները բերված են աղ.16.-ում:

Աղյուսակ 16.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	196	2%	236	4%	420	5%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	88	3%	112	3%	198	5%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	78	1%	134	2%	201	4%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	32	16%	48	26%	64	34%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	3	12%	3	12%	3	12%	24
6.	Արագագործություն	25 նվ		25 նվ		25 նվ		

Հինգերորդ սկզբունքով՝ (axb) mod m հաշվարկը Բուտի ալգորիթմի կիրառմամբ (Radix 8) սարքի պայմանական սխեման ներկայացված է նկ. 2.11: n=8, n=16 և n=32 կարգայնության a և b թվերի և $m=2^k$ մոդուլի արժեքը [0: 255] միջակայքում սինթեզման արդյունքները ներկայացված են աղ.17-ում:

Աղյուսակ 17.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	203	4%	324	5%	411	6%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	105	3%	165	6%	210	6%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	168	2%	222	3%	345	4%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	32	17%	40	21%	80	42%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	2	4%	2	4%	2	4%	24
6.	Արագագործություն	100 նվ		170 նվ		330 նվ		

n=8, n=16 և n=32 կարգայնության a և b թվերի և $m=2^k$ մոդուլի արժեքը [256:1023] միջակայքում սինթեզման արդյունքները բերված են աղ.18.-ում:

Աղյուսակ 18.

Սարքերի օգտագործման հաշվետվություն								
NN	Օգտագործված տրամաբանություն	n=8		n=16		n=32		
		Used	Utilization	Used	Utilization	Used	Utilization	Available
1.	4 մուտքանի LUTs-ի քանակ	560	6%	700	8%	945	11%	9312
2.	Օգտագործված սեկցիաների (Slices) քանակ	258	6%	372	9%	501	12%	4656
3.	Սեկցիաներում պարունակվող տրիգերների քանակ	340	4%	467	5%	624	7%	9312
4.	Օգտագործված մուտք/ելքերի (IOBs) քանակ	38	20%	46	25%	84	45%	190
5.	Օգտագործված սինքրոմուտքերի (CLK) քանակը	3	12%	3	12%	3	12%	24
6.	Արագագործություն	110 նվ		180 նվ		340 նվ		