ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹՈՒԹՅԱՆ ԵՎ ԳԻՏՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ (ՀԻՄՆԱԴՐԱՄ)

ՄԱԳԻՍՏՐՈՍԱԿԱՆ ԿՐԹԱԿԱՆ ԾՐԱԳԻՐ

**ԹԵՄԱ՝ Թվային սխեմաների իրադարձային տրամաբանական մոդելավորման միջոցի մշակումը և հետազոտումը**

Կարապետյան Ռազմիկ Կարենի

«Էլեկտրոնային նախագծման ավտոմատացում» մասնագիտությամբ ճարտարագիտության մագիստրոսի որակավորման աստիճան հայցելու ատենախոսություն

ԵՐԵՎԱՆ 2019

ՀԱՍՏԱՏՄԱՆ ԹԵՐԹ

**ԹԵՄԱ՝ Թվային սխեմաների իրադարձային տրամաբանական մոդելավորման միջոցի մշակումը և հետազոտումը**

Կարապետյան Ռազմիկ Կարենի

|  |  |
| --- | --- |
| Ատենախոսության ղեկավար՝ | Ա․ Պետրոսյան  ֆ.-մ. գ. թ. |
| Մագիստրանտ՝ | Ռ. Կ. Կարապետյան  Բակալավր |
| Գրախոս՝ | Վ. Շ. Մելիքյան  տ. գ. դ., պրոֆեսոր |
| Ամբիոնի վարիչ՝ | Վ. Շ. Մելիքյան  տ. գ. դ., պրոֆեսոր |

ԿԵՆՍԱԳՐԱԿԱՆ ՏՎՅԱԼՆԵՐ

|  |  |
| --- | --- |
| Մագիստրանտ՝ | Կարապետյան Ռազմիկ Կարենի |
| Աստիճանը՝ | «Էլեկտրոնային նախագծման ավտոմատացում» մասնագիտության ճարտարագիտության մագիստրանտ |
| Տարեթիվը՝ | 2019 |
| Ծննդյան տարեթիվը՝ | 1995 |
| Մինչ մագիստրոսական որակավորումը՝ | Ճարտարագիտության բակալավրի աստիճան |
| Մասնագիտությունը՝ | Ինֆորմատիկա և հաշվողական տեխնիկա |
| Հրատարակված աշխատանքները՝ | Չկան |

**ՀԱՄԱՌՈՏԱԳԻՐ**

**ԹԵՄԱ՝ Թվային սխեմաների իրադարձային տրամաբանական մոդելավորման միջոցի մշակումը և հետազոտումը**

Կարապետյան Ռազմիկ Կարենի

Սույն մագիստրոսական ատենախոսության շրջանակներում ուսումնասիրվել են իրադարձային տրամաբանական մոդելավորում իրականացնող մինչև այժմ հայտնի ալգորիթմնեևը և միջոցները: Դրանց համատեղման արդյունքում մշակվել է նոր ծրագրավորման գրադարան, որը հնարավորություն է տալիս նկարագրել ինտեգրալ սխեման C++ լեզվի օբյեկտների տեսքով, կատարել իրադարձային տրամաբանական մոդելավորում և ստացված արդյունքների հիմման վրա տեսնել սխեմայի փոխանջատման բնութագծերը։ Գրաֆիկական ինտերֆեյսի միջոցով օգտագործողը կարող է հավաքել սխեման։  
Ատենախոսության ընթացքում հետազոտվել է մշակված գրադարանի հիմման վրա նկարագրված սխեմայի մոդելավորման ժամանակի կախումը սխեմայի input-vector- ից։

**Բովանդակությունը**

Նկարների ցանկ

Աղյուսակների Ցանկ

ԳԼՈՒԽ 1

**Ներածություն**

ՆԵՐԱԾՈՒԹՅՈՒՆ

Վաղ ժամանակներից սխեմայի դիզայնի ստուգումը թվային թվային սխեմաների նախագծման գործընթացի կարեւորագույն մասն է կազմում։  
 Պատճառը պարզ է։ Գերարդյունավետ է ստուգել դիզայնի ճշգրտությունը նախքան արտադրելը, քան վերանորոգել կամ վերակառուցել հազարավոր սխալ արտադրված սխեմաներ: Ոչ վաղ անցյալում ստուգումը կատարվել էր փաստացի նախատիպ կառուցելով արտաքին միացումներով փոխկապակցված բաղադրիչների միացումով: Այն ժամանակ նախատիպն էր օգտագործվում գնահատելու սխեմայի տրամաբանական ճշգրտությունը եւ ժամանակային բնութագրերը:Այս մեթոդը անհարմար դարձավ թվային սխեմաների չափերի պայթյունավտանգ աճով: Գերմեծ ինտեգրալ սխեմայի բաղադրիչների քանակը կարող է հասնել հարյուր միլիոնավոր տարրերի, միաժամանակ մեծացնելով սխեմայի բարդությունը:  
 Այն դարձել է շատ ծախսատար եւ ժամանակ սպառող կառուցելու համար նախատիպեր սխեմաների համար։ Այս գործոնները առաջ քաշեցին ավտոմատացված նախագծում անող գործիքներ ստեղծելու խնդիրը։ Դիզայնի ֆիզիկական նախատիպավորման համար կենսունակ փոխարինող գործիքը դարձավ սիմուլյատորը: Մոդելավորման միջոցը հնարավորություն է տալիս նախագծողին տեսնել, թե ինչպես կպահի նախագիծը իրականում, միաժամանակ հաստատելով դիզայնը հաճախորդի առաջադրած առանձնահատկությունների հետ: Այն թույլ է տալիս հայտնաբերել եւ չափել այն իրադարձությունները, որոնք կարող են շատ լինել դժվար է կամ անհնար է հայտնաբերել փաստացի համակարգում։ Սիմուլյատորը նաեւ հնարավորություն է տալիս միացում դիզայներին նախագծային գործընթացում տարբեր գաղափարներ կիրառել փորձարկելու եւ օպտիմալացնելու համար դիզայնը։ Էլեկտրոնային սարքերի բարդությունը հասել է այնպիսի մակարդակի, որը նույնիսկ ոչ մի սիմուլյատոր չի կարող կարգավորել մոդելավորման բոլոր ասպեկտները։ Արդյունքում, տարբեր տեսակի սիմուլյատորներ հայտնվեցին տարբեր ոլորտների խնդրիների լուծման համար: Սիմուլյատորները դասակարգելու ամենատարածված ձեւը հիմնված է նրանց թվային համակարգի աբսրակցիայի մակարդակի վրա։ Հիմնականում կարելի է դիտարկել հետեվյալ հինգ տեսակները։

* Վարքագծային սիմուլյատորը գտնվում է ամենաբարձր մակարդակով: Այս մակարդակում համակարգը սիմուլյացվում է կատարող ալգորիթմների առումով, եւ ընդգծում է ընդհանուր համակարգի կայունությունը:
* Հաջորդ մակարդակում ֆունկցիոնալ սիմուլյատոր է: Նաեւ կոչվում է ռեգիստր փոխանցման մակարդակ։ Այն օգտագործվում է տվյալների հոսքի եւ հսկողության ազդանշանների ներդաշնակեցման համար ֆունկցիոնալ բլոկների միջեւ, ինչպիսիք են ռեգիստրները, կոդավորիչները, ապակոդավորիչները, թվաբանական-տրամաբանությունը միավորները և այլն։
* Հիերարխիայում հաջորդը տրամաբանական սիմուլյատորն է, որը կոչվում է նաև տրամանական բանալիների մակարդակի սիմուլյատոր, որն արտացոլում է անջատման տարրերի կամ տրամաբանական փականների փոխկապակցումը համակարգում: Այստեղ ուշադրության կենտրոնում է սխեմայի տրամաբանական ճշգրտությունը ստուգելը։Այս տեսակի սիմուլյատորը կոչվում է դիզայնի ստուգման սիմուլյատոր:
* Տրանզիստոր / էլեկտրական մակարդակի սիմուլյատորը զբաղվում է տրամաբանական բանալին տրանզիստորների բաժանելով և նրա վարքագիծը ստուգելով։
* Ցածր մակարդակի վրա է երկրաչափական մակարդակի սիմուլյատորը, որը սիմուլացնում է սխեման ֆիզիկական մարմինների առումով։

Սիմուլյացիան ամենաբարձր մակարդակում պահանջում է ավելի մանրակրկիտ մշակում, հետեւաբար սիմուլյացիայի մեծ արագություն։ Այնուամենայնիվ, տեղեկատվության կորուստը կարող է դժվարացնել հասկանալը սխեմայի վարքագծի։

Ժամանակակից ավտոմատացված նախագծման համակարգերի թվային սխեմաների վերլուծության ենթահամակարգի հիմքը կազմում են տրամաբանական մոդելավորման ծրագրերը: Տրամաբանական մոդելավորումը թույլ է տալիս նախագծվող սխեմայի մոդելի օգնությամբ նրա տրամաբանական-ժամանակային վարքագծի պատկերի ստացումը: Այդ պատկերի հիման վրա կարելի է ստուգել սխեմայի տրամաբանական կառուցվածքը մինչև սխեմայի իրականացումը, որոշել օգտագործվող էլեմենտային բազային ներկայացվող պահանջները, համեմատել թվային սխեմայի իրականացման լուծումների տարբերակները հուսալիության, արագագործության և այլ տեսանկյուններից:

Ի վերջո կարելի է ասել, որ տրամաբանական մոդելավորումը բաղկացած է երկու փուլից.

* թվային սխեմայի տրամաբանական մոդելի կառուցում;
* տված մուտքային ազդանշանների համար սխեմայի տրամաբանական մոդելի ժամանակային դիագրամների հաշվարկում:

Գոյություն ունեցող մոդելավորման համակարգերը ստեղծված են եղել ավելի փոքր ինտեգրացման աստիճան ունեցող սխեմաների համար և չեն բավարարում ներկայիս պահանջներին՝ մոդելավորման արագության և անհրաժեշտ ճշտության տեսանկյունից, կամ վերը նշված պահանջների (արագագործություն, մոդելավորման ճշտություն) հետ, տրամաբանական մոդելավորման ժամանակ առաջ են գալիս նաև օժանդակ՝ այլ հնարավորությունների ավելացման խնդիրներ, որը բերում է նաև այդ պահանջների լուծմանը: Այդ խնդիրներից է թվային սխեմաների մեծ չափողականության հետևանքով նախագծման ընթացքում հնարավոր բաց թողումների առկայությունը: Այսինքն նախագծման ընթացքում հնարավոր է սխեմայում թողնվեն առանձին մասեր, որոնք չեն մասնակցում սխեմայում կատարվող փոխանջատումներին: Այդ մասերի սխեմայում թողնելը բերում է մասնավորապես տնտեսական կորուստների և մոդելավորման խնդրի բարդացմանը: Այս խնդիրը էլ ավելի կարևոր է դառնում, երբ էլեմենտների չափսերի փոքրացմանը զուգահեռ մոդելավորման մեջ հաշվի է առնվում կապի գծերի պարամետրերը, որը բերում է նախկին մեթոդների վերանայմանը, նոր մեթոդների մշակմանն ու իրագործմանը:

Տվյալ մագիստրոսական աշխատանքի շրջանակներում կուսումնասիրվեն տրամաբանական մոդելավորում անող առկա լուծումները, ու կմշակվի իրադարձային տրամաբանական մոդելավորում անող միջոց գրաֆիակակն ինտերֆեյսով։

ԳԼՈՒԽ 2

**Գրականության ակնարկ**

ԳՐԱԿԱՆՈՒԹՅԱՆ ԱԿՆԱՐԿ

2.1.1 Մոդելավորման ֆունկցիոնալ-տրամաբանական և սխեմատեխնիկական մակարդակները

Ֆունկցիոնալ-տրամաբանական նախագծումը հանդիսանում է հաշվողական համակարգերի նախագծման առանձին մակարդակ, որի ժամանակ, ինչպես վերը նշեցինք, մշակվում են սարքերի ֆունկցիոնալ և սկզբունքային սխեմաները (օր. պրոցեսորներ, հիշող սարքեր, և տարբեր տիպի այլ սարքեր):

Եթե հետևենք խնդրի լուծման ժամանակային հաջորդականությանը, ապա ֆունկցիոնալ-տրամաբանական նախագծումը հաջորդում է սխեմատեխնիկական նախագծման փուլին: Այդ պատճառով ֆունկցիոնալ-տրամաբանական նախագծման մուտքային տվյալներ են հանդիսանում սխեմատեխնիկական նախագծման արդյունքները. սարքերի համախումբը (կազմը) և նրանց ելքային պարամետրերի նկատմամբ եղած պահանջները, ալգորիթմների մասին ինֆորմացիան, տվյալ սարքերի ճարտարապետությունը և այլն:

Ֆունկցիոնալ-տրամաբանական նախագծման հիմնական խնդիրը ֆունկցիոնալ և սկզբունքային սխեմաների մշակման խնդիրն է: Նախագծման ժամանակ այդ սխեմաները պետք է արտացոլեն օգտագործվող միկրոսխեմաների կազմը, ինչպես նաև նրանց միջև կապերը:

Միկրոսխեմաների մեջ կարող են լինել ինչպես ցածր (օր. փականներ և տրիգերներ), այնպես էլ բարձր և միջին ինտեգրացման աստիճանի սխեմաներ (օր. հաշվիչներ, տարբեր ռեգիստրներ, կիսահաղորդչային հիշող սարքեր, տրամաբանական ծրագրավորվող մատրիցաներ և այլն ):       Մեծ ինտեգրալ սխեմաների մոդելավորման ժամանակ նախագծվող սխեմաների տարրեր են հանդիսանում ֆունկցիոնալ տարրերը ( փականները, որոնք իրականացնում են պարզ բուլյան ֆունկցիաներ, հիշող տարրերը, որոնցից են տարբեր տիպի տրիգերները, անալոգային սխեմաները, օր.` օպերացիոն ուժեղացուցիչները ):

Հաշվողական համակարգերի ֆունկցիոնալ և սկզբունքային սխեմաների նախագծման և մշակման գործընթացը իր մեջ ներառում է սինթեզի  և վերլուծության խնդիրների լուծումները.

 Սինթեզի խնդիրը , որպես կանոն, հնարավոր չէ լուծել ավտոմատ փաթեթով երկու պատճառով.

1)            ոչ բոլոր պրոցեդուրաներն են ձևայնացված ,

2)            մեծ թվով ձևայնացված պրոցեդուրաների համար հայտնաբերված չեն արդյունավետ ալգորիթմներ, որոնց վերջնական տեսքը մոտ կլինի գծային կախվածությամբ լուծման խնդիրներին:

Մեծ թվով սինթեզի ալգորիթմների համար մեքենայական ժամանակի ծախսերը աճում են ոչ գծային ( օր. էքսպոնենցիալ) ձևով` սինթեզվող օբյեկտի բարդության աստիճանին զուգահեռ: Այդպիսի ալգորիթմները կարող են օգտագործվել միայն հարաբերականորեն պարզ սխեմաների հատվածների սինթեզի ժամանակ:

Նրանց օգտագործման համար անհրաժեշտ է սխեման նախապես ենթարկել տարանջատման: Ֆունկցիոնալ տրամաբանական նախագծման հիմնական խնդիրները, որոնք համեմատաբար ավելի ձևայնացված են, դրանք վերլուծության խնդիրներն են:

Վերլուծության հիմնական նպատակը սխեմայի առաջարկվող տարբերակի որակի գնահատումն է: Այդ գնահատականը կարող է լինել բազմամակարդակային:

Նախապես նպատակահարմար է ստուգել սխեմայի համապատասխանությունը տրված ֆունկցիաներին` առանց հաշվի առնելու ազդանշանների հապաղումը, տարրային բազայի սահմանափակումը և այլ տարբեր ապակայունացնող գործոններ: Այդպիսի ստուգումը կարող է կատարվել առանց մեքենայական ժամանակի զգալի կորուստների և թույլատրում է հայտնաբերել սինթեզի ժամանակ կառուցվածքում թույլ տրված սխալները: Հայտնաբերված սխալների վերացումից հետո կարելի է շարունակել վերլուծությունը` անցնելով սխեմայի ավելի կոնկրետ մոդելների օգտագործմանը, որոնք հաշվի են առնում տարրերի հապաղումները, տարբեր ապակայունացնող գործոններ և այլն:

Վերլուծության խնդրի այս փուլի ժամանակ հնարավորություն է ընձեռնվում հայտնաբերելու ազդանշանների կրիտիկական շեղումը, որն առաջանում է ասինխրոն սխեմաներում, ինչպես նաև խափանման այլ պատճառներ: Քանի որ վերլուծությունը կարող է կատարվել սխեմաների մի քանի համեմատվող տարբերակների համար, հնարավորություն է ընձեռնվում ընտրելու դրանցից լավագույնը ( օպտիմալ ), օր. արագագործության տեսանկյունից: Այդպիսով վերլուծության հիման վրա որոշվում է սխեմայի սինթեզի խնդիրը: Ֆունկցիոնալ տրամաբանական նախագծման ուրույն խնդիրը թեստերի սինթեզի խնդիրն է: Թեստերը անհրաժեշտ են սարքավորումների հսկման համար, դրանց պատրաստման եւ շահագործման ընթացքում: Թեստային հսկումը պետք է տա հավաստի եւ ամբողջական արդյունքներ: Այդ պատճառով արդյունավետ թեստերի ստեղծումը պատկանում է քոմփյութերային համակարգերի վրա լուծում պահանջող բավականին բարդ խնդիրների դասին:

Տրամաբանական և սխեմատեխնիկական մոդելավորման մակարդակները սկսեցին զարգանալ իրարից անկախ 50-ական թվակաների վերջերում: Այդ ժամանակահատվածում ստեղծվեցին մեծ քանակությամբ մոդելներ, մեթոդներ, ալգորիթմներ այդ թվում նաև ծրագրային փաթեթներ, որոնք այդ ժամանակ լուծեցին թվային սխեմաների նախագծողների  համար կոնկրետ խնդիրներ: Անգամ ընդունված է առ այսօր թվային համակարգերի  ստեղծված միջոցները բաժանել երկու սերնդի: Թվային սխեմաների մոդելավորման միջոցների զարգացման ընթացքի վերլուծությունը ցույց է տալիս, որ հիմնական շարժիչ գործոնն էր հակասությունը հաշվարկների արդյունքների ճշտության պահանջների և մեքենայական միջոցների անհրաժեշտ ծախսերի միջև: Զարգացման ամեն փուլում ստեղծվում էր այն իրադրությունը, երբ օգտագործվող թվային սխեմաների վերլուծության միջոցները միայն մասնակիորեն էին բավարարում թվային սարքավորումների նախագծողների պահանջներին: Այդ պատճառով արդեն գոյություն ունեցող մոդելավորման համակարգերի համար փնտրվում էին կատարելագործման նոր ճանապարհներ: Այդ իրավիճակի պատճառն հանդիսացան թվային սխեմաների պատրաստման տեխնոլոգիաների առաջ ընկնող տեմպերը մոդելավորման միջոցների նախագծման ոլորտի նկատմամբ: Տվյալ միտումը դարձավ ավելի նկատելի ինտեգրալ տեխնոլոգիային անցման հետևանքով և առավել սրվեց մեծ ինտեգրացիայի աստիճան ունեցող սխեմաների առաջ գալուն պես:

Միկրոսխեմաների պատրաստման ամեն մի նոր տեխնոլոգիայի անցումը թվային սխեմաների սխեմատեխնիկական և տրամաբանական մոդելավորման միջոցների նախագծողների համար բերում էր երկու կարևորագույն պրոբլեմների.

1.            Նախագծվող սարքերի էլեմենտների քանակը կտրուկ, հաճախ միանգամից մի քանի կարգով, աճում էր և տվյալ պահին ներկա վերլուծության միջոցները անկարող էին գտնվում մոդելավորել տվյալ քանակության էլեմենտներով սխեմա:

2.  Տեղի էր ունենում տարբեր ֆիզիկական երևույթների որակական         վերագնահատում, որոնք տեղի էին ունենում թվային սխեմայում նրա   գործունեության ժամանակ: Երևույթները, որոնց կարելի էր անտեսել մինչև նոր տեխնոլոգիայի մշակումը, դառնում էին որոշիչ նոր տիպի սխեմաների գործունեության համար, իսկ գոյություն ունեցող վերլուծության համակարգերը նախատեսված չէին տվյալ հաշվարկների համար:

Վերլուծության տրամաբանական մակարդակի ժամանակ, հատկապես իրադարձային մոդելավորման մտքի առաջարկումից հետո, չափողականության պրոբլեմը ոչ այդքան սուր էր դրված, ինչպես սխեմատեխնիկական վերլուծության ժամանակ: Հիմնական պատճառը բուլյան հանրահաշվի վրա հիմնված թվային էլեմենտների պարզ տրամաբանական մոդելների օգտագործումն էր, որոնք հաշվարկի համար չէին պահանջում բարդ լուծումներ: Էլեկտրական վերլուծության համար միշտ առավել կարևոր պրոբլեմ էր հանդիսանում լուծվող դիֆերենցիալ հավասարումների համակարգի չափողականությունը: Հաշվարկների քանակի  ոչ գծային կախվածությունը սխեմայում էլեմենտների քանակից բերում է մեքենայական ժամանակի անթույլատրելի ծախսերի ժամանակակից ՄԻՍ-երի սխեմատեխնիկական մոդելավորման ժամանակ: Հաշվարկների թույլատրելի ժամանակի ընթացքում լավագույն սխեմատեխնիկական մոդելավորման ծրագրերի օգնությամբ կարելի է վերլուծել միայն սխեմաներ, որոնք պարունակում են ոչ ավել, քան 200-300 տրանզիստորներ: Եվ այդ ամենը տեղի է ունենում անգամ եթե այդ ծրագրերում կիրառվում են բոլոր առաջադեմ մտահաղացումները, որոնք առաջացել են սխեմատեխնիկական մոդելավորման զարգացման ընթացքում (դիֆերենցիալ հավասարումների համակարգերի, մակրոմոդելավորման, իրադարձային  և կառուցվածքային տարանջատման, ժամանակային և նիշային վերլուծության, զուգահեռ մոդելավորման  նոր, ավելի արագ թվային մեթոդներ):

Մոդելավորվող սխեմայի չափողականության նվազեցման նշված զարգացման ուղիներից առավել արդյունավետ հանդիսացավ խառը տրամաբանական-էլեկտրակական մոդելավորման մտահաղացումը: Այն հանգում է նրան, որ կախված օգտագործողի կողմից պահանջվող սխեմայում տեղի ունեցող պրոցեսների արտացոլման մանրակրկիտության աստիճանից, տարբեր դրվագները մանրամասնացվում են մինչև տարբեր մակարդակներ՝ էլեկտրական կամ տրամաբանական:

էլեկտրական մակարդակը ներկայացնում է դրվագներ, որոնք պահանջում են ազդանշանների ձևի առավել մանրակրկիտ հաշվարկներ, նկարագրվում է դիֆերենցիալ հավասարումների համակարգերի օգնությամբ և լուծվում է սխեմատեխնիկական վերլուծության հասարակ միջոցներով: Տրաբանական մակարդակը իր մեջ ներառում է սխեմաների դրվագներ, որոնցում պետք է հստակ մոդելավորել ժամանակային հարաբերակցությունները, իսկ ազդանշանի ձևը ունի երկրորդական նշանակություն: Տրաբանական մոդելավորումը տրվում է բուլյան հավասարումների համակարգով կամ էլ ռեգիստրային փոխանցումների տերմիններով և հաշվարկվում է տրամաբանական մոդելավորման մեթոդներով: Այդպիսով, մեկ ծրագրում համագործակցում են սկզբունքային տարբեր ալգորիթմներ՝  դիֆերենցիալ հավասարումների  համակարգերի լուծման թվային մեթոդներ, որոնք նկարագրում են էլեկտրական մակարդակը, և տրամաբանական հավասարումների լուծում տրաբանական մակարդակի համար: Տարբեր բնույթի ազդանշանների փոխադարձ վերաձևավորումները իրագործվում են մուտքի և ելքի մոդելների միջոցով: Ըստ էության, տեղի է ունենում փոխազդում մոդելավորման երկու մակարդակների (սխեմատեխնիկական և տրամաբանական) առավելությունների և թերությունների միջև: Տրաբանական մակարդակում իրադարձային տրամաբանական մոդելավորման օգտագործումը խառը վերլուծության ծրագրում ապահովում է մեքենայական միջոցների (ժամանակ և հիշողություն) էական կրճատում: Կրճատումը հասնում է մինչև 1-3 կարգի սխեմատեխնիկական վերլուծության համեմատ: Բայց դա նույնպես չի բավարարում ժամանակակից պահանջներին:

Գոյություն ունեցող տրամաբանական մոդելավորման միջոցները նախատեսված չեն ֆիզիկական  երևույթների գնահատմանը (օրինակ` ազդանշանների  պարազի­տային երևույթներ): Այդ պատճառով խառը տրամաբանական-էլեկտրական մոդելավորման ծրագրերի ստեղծման ժամանակ հիմնական ուշադրությունը ուղղված էր սխեմատեխնիկական մոդելների զարգացման վրա նրանց՝ տրամաբանականների հետ համատեղ գործունեության տեսանկյունից, հետևաբար այդ երևույթները այժմ հաշվի են առնվում:

Բացի այդ, տվյալ փաթեթներում հաճախ չեն նախատեսվում լրացուցիչ, մեծամասամբ կարևոր և օգտակար ֆունկցիաների կիրառում (օրինակ` սխեմաների նախորդ մոդելավորման արդյունքների մասին ինֆորմացիայի հավաքում ):

Սակայն այդ խնդիրները լուծելու համար նախ և առաջ ծանոթանանք մոդելավորման հայտնի մեթոդներին:

2.1. Թվային սխեմաների մոդելավորման մակարդակները

            Մոդելավորման համակարգային մակարդակում մշակվում է հաշվողական սարքը, որը բաղկացած է մեծ բլոկերից, օրինակ՝ պրոցեսոր, օպերատիվ հիշողություն, տվյալների հաղորդման միջոցներ և այլն: Այդ մակարդակում, որտեղ մոդելավորման բնորոշ խնդիրներ են հանդիսանում առանձին բլոկերի ինֆորմացիոն համաձայնությունը և այլն, կիրառվում է զանգվածային սպասարկման տեսությունը:

            Ֆունկցիոնալ-տրամաբանական մակարդակում մոդելավորվում է թվային սարքը, որը մանրամասնեցվում է մինչև բազմակարգային էլեմենտները (ռեգիստրներ, հաշվիչներ, վերծանիչներ և այլն) կամ էլ ներկայացվում է առանձին փականներով և տրիգերներով (տրամաբանական ենթահամակարգ): Այդ դեպքում վերլուծության խնդիրներից են հանդիսանում միկրոծրագրի գործունեության տրամաբանության ստուգումը, առանձին էլեմենտների աշխատանքի ժամանակային համաձայնեցումը, սխեմայի սահմանային արագագործության որոշումը, սխեմայի գործունեության տարբեր տիպի խափանումների հայտնաբերումը և այլն: Տրամաբանական մոդելավորման  համակարգի կառուցման հիմքում ընկած է բուլյան հանրահաշիվը: Թվային սխեմաների առավել  հայտնի տրամաբանական մոդելավորման ծրագրերից են՝ HILO, ADM, VERILOG և այլն:

Վերլուծության սխեմատեխնիկական մակարդակում ստուգվում է սխեմայի փոքր դրվագների գործունեությունը, որոնք մանրամասնեցվում են մինչև առանձին էլեկտրոնային բաղադրիչների՝ տրանզիստորների, դիոդների, դիմադրությունների և այլն: Ստացվում են սխեմայի հանգույցներում լարումների և հոսանքների փոփոխման մանրամասն ձևերը, որոնց հիման վրա կարելի է լուծել այնպիսի խնդիրներ, ինչպիսիք են առանձին էլեմենտների էլեկտրական համաձայնեցումը, զգայունության վերլուծությունը սխեմայի այս կամ այն բաղադրիչի արժեքների փոփոխության նկատմամբ, հաճախականային բնութագրերի ստացումը և այլն:

            Բաղադրիչային մակարդակի վրա մոդելավորվում են կիսահաղորդչային կառուցվածքի առանձին դրվագներ (հաճախ՝ առանձին տրանզիստորը), որոնք ներկայացվում են որպես ֆիզիկա-տոպոլոգիական օբյեկտներ: Կիսահաղորդչային  կառուցվածքների տեսության հիման վրա  հաշվարկվում են բաղադրիչների էլեկտրական պարամետրերը և նրանց մեջ կատարվող ֆիզիզկական պրոցեսները: Մոդելավորման բաղադրիչային մակարդակի մոդելների իրագործումը կատարվում է մասնակի ածանցյալներով դիֆերեցիալ հավասարումների համակարգերի միջոցով:

Նախագծողների մոտ հիմնական և առավել հայտնի թվային սխեմաների մոդելավորման մակարդակներից են համարվում տրամաբանականը և սխեմատեխնիկականը:

Այդ պատճառով ավելի մանրամասն անդրադառնանք այդ մակարդակներին:

2.2. Տրամաբանական մոդելավորման հայտնի մեթոդները

Տրամաբանական մոդելվորման ընթացքում կարող են լուծվել տարբեր խնդիրներ . սխեմաների աշխատանքի տրամաբանության ստուգում, անցողիկ  պրոցեսների վերլուծում, սխեմաների աշխատանքի հուսալիության որոշում էլեմենտների պարամետրերի ցրումից կախված, թեստերի գեներացիա և այլ: Դրված խնդրից կախված ընտրվում է մոդելավորման մեթոդը: Մեթոդները հիմնականորեն իրարից տարբերվում են  հետևյալով՝  սխեմայում ազդանշանի տարածման ժամանակի հաշվման ձևերով, ազդանշանի կոդավորման ձևերով, քոմփյութերում մոդելի կառուցման ձևերով, էլեմենտների մոդելավորման հերթականությամբ:

Սխեմայում ազդանշանի տարածման ժամանակի հաշվման ձևերից կախված մեթոդները լինում են սինխրոն (առանց սխեմայի էլեմենտների հապաղումները հաշվի առնման) և ասինխրոն (հապաղումները հաշվի առնելով): Ազդանշանի կոդավորման ձևերից կախված տարբերում են երկարժեք և բազմարժեք մոդելավորման մեթոդներ: Էլեմենտների մոդելավորման հերթականությունից կախված՝ միջանցիկ և իրադարձային: Մոդելավորման ծրագրի աշխատանքի կազմակերպման ձևերից կախված՝ կոմպիլյատիվ և ինտերպրետատիվ մեթոդներ:

Սինխրոն մոդել կոչվում է, այն մոդելը, որի ժամանակ հաշվի չեն առնվում էլեմենտների ֆիզիկական հատկությունները, օրինակ էլեմենտի հապաղման ժամանակը: Դա հաշվի է առնվում ասինխրոն մոդելների օգտագործման ժամանակ: Եթե սինխրոն մոդելում օգտագործվում են բուլյան փոփոխականներ, ապա մոդելը կոչվում է երկարժեք:

Երկարժեք մոդելները մեքենայական միջոցների տնտեսման տեսանկյունից ամենաշահավետն են, բայց այդ մոդելների օգնությամբ կարելի է լուծել խնդիրների նեղ շրջանակ, օրինակ ֆունկցիոնալ սխեմայի նախագծման ընթացքում կատարած կոպիտ սխալների հայտնաբերում:

   Անցողիկ պրոցեսների վերլուծության ընթացքում անհրաժեշտ է հաշվի առնել սխեմայում ազդանշանի տարածման վերջավոր արագությունը: Սխեմայի ցանկացած տրամաբանական էլեմենտի փոխանջատումը կատարվում է ոչ թե ակնթարթորեն, այլ որոշակի հապաղումից հետո նրա մուտքերի ազդանշանների փոփոխման պահի նկատմամբ: Հապաղումով է տարածվում նաև ազդանշանը սխեմայի կապի գծերով:

   Ազդանշանի հապաղումը տրամաբանական էլեմենտում կախված է տվյալ էլեմենտի ֆիզիկական իրագործման, սխեմայում միացման, մուտքային ազդանշանների համակցությունների, ջերմաստիճանի, պարամետրերի ցրումի ձևերից: Հետևաբար ազդանշանի հապաղումը ունի պատահական արժեք: Շղթաներով ազդանշանի տարածման տարբեր հապաղումները բերում են ազդանշանի ձևափոխությանը էլեմենտների փոխանջատման ընթացքում:

   Սխեմայի անցողիկ պրոցեսների վերլուծության նպատակները կարող են լինել տարբեր, բայց հիմնականում դա սխեմայի վարքագծի որոշումն է տրված մուտքային ազդանշանների դեպքում: Դա հատկապես կարևոր է ասինխրոն սխեմաների համար, որոնց վարքը կախված է ազդանշանների հապաղումներից:

   Ֆունկցիոնալ էլեմենտների և նյութերի, օգտագործվող կառուցվածքային հանգույցների պատրաստման համար,  պարամետրերի ցրման պատճառով սխեմայի արձագանքը նույն մուտքային համակցության նկատմամբ կարող է լինել տարբեր: Սխեմաները պետք է նախագծվեն այնպես, որ պատրաստման պարամետրերի ցրումից անկախ սխեմաները նույն մուտքային ազդանշանների դեպքում միշտ անցնեն նույն վիճակներին: Եթե օգտագործվող մուտքային ազդեցության համար կա ոչ 0-ական հավանականություն, որ սխեման կանցնի տարբեր վիճակների, ապա սխեման նախագծված է սխալ:

   Կառուցվածքային հանգույցների սխեմաների համաձայնեցման ընթացքում պետք է հայտնի լինի կառուցվածքային հանգույցի ելքային ազդանշանների ժամանակային շեղումը: Այդպիսի արդյունքներ վերլուծության ընթացքում ստանալու համար անհրաժեշտ է ֆիքսել էլեմենտի փոխանջատման պահը:

   Էլեմենտի մուտքային ազդանշանի անհամաձայնեցման դեպքում կարող է առաջանալ կեղծ ազդանշան տրամաբանական էլեմենտի ելքին: Կեղծ ազդանշանների առաջացման հնարավորությունը կոչվում է խափանման ռիսկ: Եթե ազդանշանները սխեմայի ելքին երկու կից A և B մուտքային ազդեցությունների հավաքածուի տակ մնուն են նույնը, բայց անցողիկ պրոցեսի ընթացքում հնարավոր է հակառակ արժեքի կեղծ ազդանշանի առաջացումը, ապա այդպիսի դրությունը կոչվում է ստատիկ խափանման ռիսկ:  Այսինքն խափանման ռիսկը կեղծ ազդանշանների հայտնաբերման միջոց է: Խափանման ռիսկի որոշման համար ևս օգտագործում են տարբեր մոդելներ` բազմարժեք մոդելներ

   Դինամիկ խափանման ռիսկը ենթադրում է ելքում ազդանշանի արժեքի բազմակի փոփոխման հնարավորությունը A մուտքային հավաքածուից B հավաքածու փոխվելու դեպքում, երբ սխեմայի ելքային ազդանշանը փոխվում է հակառակին:

2.2.1 Սինխրոն տրամաբանական մոդելավորում

Սինխրոն մոդելավորման ընթացքում էլեմենտների մոդելները ներկայացվում են իրենց տրամաբանական ֆունկցիաներով առանց ազդանշանների հապաղումները հաշվի առնելու, իսկ ազդանշանները՝ 0 և 1 արժեքներով: Սինխրոն մոդելավորումը օգտագործվում է  դիսկրետ  սարքերի տրամաբանական գործառման ճշգրտության գնահատման համար առանց անցողիկ պրոցեսների հաշվի առնման: Սխեմայի աշխատանքի մոդելավորումը բերվում է տրամաբանական էլեմենտների ելքերի ազդանշանների արժեքների հաշվմանը ըստ տրված մուտքային ազդանշանների:

Մոդելավորումը կատարվում է մուտքերում ազդանշանի ամեն մի փոփոխման համար: Ենթադրվում է, որ անցողիկ պրոցեսները սխեմայում վերջանում են ինչ-որ մի Dt  ժամանակի ընթացքում, որը փոքր է մի տակտի տևողությունից:

Սինխրոն մոդելավորման օգտագործումը առավելապես հարմար է  կոմբինացիոն սխեմաների աշխատանքի վերլուծության համար հաստատված ռեժիմում: Այդ դեպքում մոդելավորման արդյունքը ճշգրտորեն համապատասխանում է իրական սխեմային:

Սխեմայի ամեն մի էլեմենտը նկարագրվում է Y = f (X1, X2, ..., Xn) տրամաբանական բանաձևով, որտեղ Y - էլեմենտի ելքային, X1, X2, ..., Xn - էլեմենտի մուտքային ազդանշաններն են: Արդյունքում լրիվ սխեման նկարագրվում է այդպիսի բանաձևերի համակարգով: Սինխրոն մոդելավորումը բերվում է նրանց հերթականորեն լուծմանը: Տրամաբանական բանաձևի լուծման տակ հասկանում ենք Y տրամաբանական արժեքի հաշվումը ըստ հայտնի X1, X2, ..., Xn տրամաբանական արժեքների: Սինխրոն մոդելավորման առանձնահատկությունն է տրամաբանական բանաձևերի լուծումը ըստ տրված հերթականությանը, որը համապատասխանում է ազդանշանի էլեմենտների միջով անցման հերթականությանը: Այդ հերթականությունը որոշելու համար սխեման նախնականորեն պետք է ռանգավորել, որ ամեն մի Y = f (X1, X2, ..., Xn) բանաձևի լուծման պահին բոլոր X1, X2, ..., Xn  արժեքները լինեն հայտնի: Ռանգավորման ժամանակ ընդունվում է, որ սխեմայի մուտքային ազդանշանները  մոդելավորման սկսման պահին հայտնի են և նրանց վերագրվում է r = 0 ռանգ:

Փոքր-ինչ ավելի բարդ է մոդելավորումը հետադարձ կապերով տրամաբանական սխեմաների դեպքում: Այդպիսի սխեմաների մոդելավորման սկզբունքը դիտարկելու համար հետադարձ կապի շղթայի մեջ հարմար է մտցնել հապաղման էլեմենտ: Իրական սխեմաներում այդ հապաղման էլեմենտը որպես ֆիզիկական էլեմենտ բացակայում է, իսկ հապաղումը տեղի է ունենում ամեն մի էլեմենտի  գործարկման հապաղման պատճառով:

Հետադարձ կապերով տրամաբանական սխեմաների մոդելավորման ընթացքում սխեման ռանգավորվում ու մոդելավորվում է որպես հասարակ կոմբինացիոն սխեմա այն ենթադրությամբ, որ հապաղման էլեմենտի միացման տեղում  հետադարձ շղթան ժամանակավոր անջատվում է և էլեմենտի մուտքերին միացված հետադարձ կապի շղթաներին վերագրվում է 0 ռանգ: Այդ դեպքում նաև ընդունում ենք, որ հետադարձ կապի շղթաներով էլեմենտների մուտքերին տրվում են ազդանշաններ ՝ համապատասխանող նախկինում տրված 0 կամ 1 վիճակներին: Սխեման մոդելավորվում է և նոր հաշված հետադարձ կապի ազդանշանի արծեքները  տրվում են էլեմենտների մուտքերին ինչ որ Dt  ժամանակ անց, որը հավասար է հետադարձ կապի շղթայի հապաղման ժամանակին:

Սինխրոն մոդելավորման արդյունքն է ժամանակային դիագրամը, որը ներկայացվում է 0-երի և 1-երի հաջորդականությամբ: Մոդելավորման ամեն մի տակտի համար դիագրամում բերվում են մուտքային ազդեցությունների և սխեմայի էլեմենտների ելքերի ազդանշանների արժեքները: Նրանով վերլուծվում է սարքի աշխատանքի ճշգրտությունը: Սովորաբար սարքի մոդելավորումը կատարվում է ինչ-որ մի տեստային ազդեցությունների համախմբության համար, որոնց համար հայտնի է էտալոնային սխեմայի արձագանքը: Կոնկրետ դեպքում արդյունքները համատեղելով պահանջվող էտալոնային արձագանքի հետ, որոշում են սարքի աշխատանքի ճշգրտությունը:

Սինխրոն մոդելավորումը երկարժեք ազդանշանի ներկայացումով մոդելավորման հեշտագույն եղանակն է: Նրա կարևորագույն առավելությունը ՝ արագագործությունն է, սակայն սինխրոն մոդելավորունը չի թույլատրում սխեմայում վերլուծել անցողիկ պրոցեսները և բացահայտել սխալներ, որոնք առաջանում են սխեմայի էլեմենտներում աղդանշանների հապաղման պատճառով:

2.2.2 Ասինխրոն տրամաբանական մոդելավորում

Անցողիկ պրոցեսների վերլուծությունը տրամաբանական սխեմաներում կատարվում է մոդելավորման ասինխրոն մեթոդով, որտեղ հաշվի է առնվում ազդանշանի տարածման ժամանակը էլեմենտներում և սխեմայի միացնող շղթաներում:

Տրամաբանական էլեմենտի գործարկումը կատարվոմ է ինչ-որ հապաղումով ըստ մուտքային ազդանշանների, որը հաշվի է առնվում հապաղումով էլեմենտների մոդելներում: Ամեն մի էլեմենտը բնութագրվում է ինչ-որ մի միջին հապաղումով, որի արժեքը կարող է փոխվել կախված էլեմենտի աշխատանքի ռեժիմից, մուտքային ազդանշանների համակցությունից, ջերմաստիճանից և այլ:

Էլեմենտի մուտքային ազդանշանի ժամանակային ապահամաձայնեցումը կարող է բերել տրամաբանական էլեմենտի ելքին կեղծ ազդանշանի հայտնմանը:

Հիշողությունով տրամաբանական էլեմենտներում, այսինքն հետադարձ կապերով սխեմաներում, մուտքային ազդանշանների ազդեցության տակ մի քանի հիշողության էլեմենտեր միանգամից կարող են փոխել իրենց վիճակները: Մի քանի դեպքերում սխեմայի վերջնական վիճակը կախված է հիշողության էլեմենտների փոխանջատման հաջորդականությունից: Այդ դեպքերում սխեմայում կան հետադարձ կապերի ազդանշանների մրցակցություն: Եթե մուտքային ազդանշանի ազդեցության տակ սխեման մի վիճակից կարող է անցնել տարբեր վիճակների, կախված սխեմայի էլեմենտների հապաղումներից, ապա մրցակցությունները կոչվում են կրիտիկական:

Ասինխրոն մոդելավորումը թույլ է տալիս հայտնաբերել խափանման ռիսկերը և կրիտիկական մրցակցությունները:

Տրամաբանական էլեմենտի մոդելը ասինխրոն մոդելավորման համար ցուցաբերվում է ոչ իներցիոն տրամաբանական էլեմենտի, իրականացնող տվյալ ֆունկցիան, և հապաղման էլեմենտի հաջորդաբար միացման ձևով:

Տրամաբանական էլեմենտի ասինխրոն մոդելը կազմվում  է ելենելով նրանից, որ ամեն մի էլեմենտ հապաղում է ազդանշանը Dt ժամանակով: Անցողիկ պրոցեսը վերլուծելու համար անհրաժեշտ է սխեման դիտել որպես ասինխրոն նրա աշխատանքի մի տակտի ընթացքում: Այն նշանակում է, որ էլեմենտները փոխարկվում են ասինխրոն կերպով, որքան որ ազդանշանները փոխվում են նրանց մուտքերում: Սակայն անցողիկ պրոցեսի տևողությունը չպետք է գերազանցի սխեմայի աշխատանքի DT տակտի տևողությունից, որը պետք է պատիկ լինի Dt աբստրակտ հապաղմանը և գերազանցի սխեմայում ազդանշանի մաքսիմալ հապաղումից: Սխեմայի մուտքերին ազդանշանի փոխվելու պահին անցողիկ պրոցեսը ավարտվում է, որովհետև մուտքային ազդանշանների տևողությունը հավասար է սխեմայի աշխատանքի տակտին:

Սխեմայի վիճակի մոդելավորումը իրականացվում է ժամանակի դիսկրետ պահերին Dt աբստրակտ հապաղմանը հավասար քայլով: Մոդելավորման ամեն մի տակտի ընթացքում սխեմայում ազդանշանները  առաջ են մղվում Dt ժամանակով: Սխեմայի ամեն մի էլեմենտի համար մտցնվում են գրգռում և արձագանք հասկացողությունները: Ցանկացած ազդեցություն էլեմենտի մուտքին բերում է գրգռմանը, որը հասնում է էլեմենտի ելքին  Dt = k Dt ժամանակի ընթացքում: Էլեմենտը գտնվում է կայուն վիճակում, եթե նրա արձագանքը չի հակասում գրգռմանը: Եթե սխեմայի բոլոր էլեմենտները գտնվում են  կայուն վաճակում, ապա սխեման նաև գտնվում է կայուն վիճակում:

Մոդելավորման ընթացքում կրկնվում են տեսածրման և իրագործման ընթացակարգը: Տեսածրման ընթացքում բոլոր էլեմենտների մուտքերը դիտվում են  mDt ժամանակի պահերին, որտեղ m = 1, 2, ...: Տեսածրումը ընդհատվում է երբ գրգռումը  հասնում է էլեմենտի ելքին, այնուհետև կատարվում է իրագործման ընթացակարգը: Մոդելավորման ընթացքում որոշվում են ազդանշանների փոփոխությունները և սխեմայի վարքը անցողիկ պրոցեսի ընթացքում: Սխեմայի մոդելավորման արդյունքը ցուցաբերվում է էլեմենտների փոխարկման ժամանակային դիագրամների տեսքով, որոնց հիման վրա վերլուծվում է սխեմայի աշխատանքը:

Ասինխրոն մոդելավորումը պահանջում է մեքենայական ժամանակի մեծ ծախսեր, և այն իրականացնելու համար օգտագործվում է  պատահարային սկզբունքը. մոդելավորվում են միայն այն տրամաբանական էլեմենտները, որոնց մուտքերը փոխել են իրենց վիճակը ներկա ժամանակի պահին:

Ասինխրոն պատահարային մոդելավորման ամեն մի տակտում դիտվում են այն էլեմենտները, որոնց գոնե մեկ մուտքը փոխում է իր արժեքը: Սկզբում արժեք է տրվում սխեմայի մուտքային շղթաներին: Առաջին տակտի ընթացքում մոդելավորվում են  A1 բազմության էլեմենտները, որոնց մուտքը միացվում է սխեմայի  գոնե մի մուտքային շղթայի: Այն էլեմենտների բազմությունը, որոնք արժեք են ընդունում մոտակա տակտում, նշանակենք B1, որտեղ B1 ծ A1: Այնուհետև արժեք է տրվում B1 բազմության էլեմենտների ելքային շղթաներին: Այդ երկրորդ տակտում առաջացնում է A2 բազմության այն էլեմենտների մոդելավորումը, որոնց մուտքերը պարունակում են B1 բազմության էլեմենտների գոնե մեկ ելքային շղթա: Այդպիսի էլեմենտների մոդելավորումից հետո ընտրվում են B2 ծ (A1 / B1) Լ A2 բազմության այնպիսի էլեմենտները, որոնք արժեք են ստանում մոտակա տակտում: B2 բազմությունը որոշում է A3 էլեմենտների բազմությունը մոդելավորման հաջորդ տակտի համար և այլն:

Երբ էլեմենտը անցնում է անորոշ վիճակի, երկուական մոդելավորման ժամանակ իտերացիոն պրոցեսը չի զուգամիտվում: Այդ պատճառով մոդելավորման ալգորիթմի մեջ պետք է մտցնել սահմանափակում իտերացիաների մաքսիմալ քանակի վրա, ինչին հասնելուց սխեմայոմ պետք է ֆիքսվի գեներացիա:

Սինխրոն մոդելավորման համեմատ ասինխրոն մոդելավորումը պահանջում է բավականին մեծ քանակի հաշվարկների կատարում: Ծրագրի ծավալը նույնպես մեծանում է հապաղման էլեմենտների մոդելավորման անհրաժեշտության պատճառով:

2.2.3 Ասինխրոն իրադարձային մոդելավորում

Դիսկրետ սարքերի աշխատանքի վերլուծությունը ցույց է տալիս, որ ցանկացած պահին ակտիվ վիճակում գտնվում են սխեմայի բոլոր էլեմենտներից միայն 1...2,5 %: Հետևաբար, մոդելավորման ժամանակի էական պակասացման կարելի է հասնել, եթե ամեն անգամ մոդելավորել միայն այն էլեմենտները, որոնց մուտքային ազդանշանները փոխվել են: Այդպիսի մեդելավորման մեթոդը կոչվում է իրադարձային: Պատահարը իրադարձային մոդելավորման համակարգում՝  դա  ինչ-որ մի էլեմենտի վիճակի և նրա հետ կապված շղթաների փոփոխումն է:

Ասինխրոն իրադարձային մոդելավորման ծրագրերում հիմնական դերն են խաղում երկու զանգվածներ՝  մոդելավորվող սխեմայի հանգույցների վիճակների զանգված և ապագա իրադարձությունների հերթը: Հանգույցների վիճակների զանգվածը պահում է մոդելավորվող սխեմայի բոլոր շղթաների ընթացիկ վիճակները 0 և 1 տրամաբանական տեսքով: Մոդելավորման ընթացքում ապագա իրադարձությունների հերթում գրանցվում են այն իրադարձությունները, որոնք պետք է ապագայում առաջանան մոդելավորվող սխեմայում: Ապագա իրադարձությունների հերթի ամեն մի էլեմենտ պարունակում է շղթայի համարը, որտեղ պետք է առաջանա վիճակի փոփոխում, և իրադարձության առաջացման ժամանակի պահը: Պատահարները ապագա իրադարձությունների հերթում գրված են ժամանակի աճման կարգով, և առաջին տեղում գրված է այն իրադարձությունը, որը կկատարվի բոլորից առաջ:

Ասինխրոն իրադարձային մոդելավորումը կատարվում է հետևյալ ձևով: Մոդելավորման սկզբում սահմանվում է սխեմայի սկղզբնական վիճակը՝ հանգույցների վիճակների արժեքները զանգված գրանցելով: Սխեմային տրվող թեստային մուտքային ազդեցությունները մտցնվում են ապագա իրադարձությունների հերթի մեջ ըստ նրանց առաջացման ժամանակի: Ապա սկսվում է մոդելավորումը, որը բաղկացած է հետևյալ գործողություններից:

1. Ապագա իրադարձությունների հերթից ընտրվում է առաջին էլեմենտը: Նրանում նշված ժամանակը մտցվում է մոդելավորման ժամանակի հաշվիչի մեջ, իսկ շղթաների վիճակների զանգվածի մեջ ըստ էլեմենտում նշված համարի գրանցվում է ապագա իրադարձությունների հերթում նշված, շղթայի նոր վիճակը հնի փոխարեն:

2. Գտնվում են տրամաբանական էլեմենտներ, որոնց համար տրված շղթան մուտքային է, և հաշվվում են այդ էլեմենտների ելքային ազդանշանների արժեքները (այսինքն որոշվում են շղթաների նոր վիճակները) և նրանց հապաղումները:

3. Ամեն մի շղթայի համար ազդանշանի արժեքը համեմատվում է շղթաների վիճակների զանգվածի մեջ գտնվող արժեքի հետ, և եթե նրանք չեն համընկնում, ապա կատարվում է շղթայի վիճակի փոփոխում, և իրադարձությունը գրանցվում է ապագա իրադարձութունների հերթի մեջ: Եթե արժեքները համընկնում են, ապա գրանցումը չի կատարվում:

Այնուհետև գործողությունները կրկնվում են սկսած 1-ից:

Մոդելավորման պրոցեսը վերջանում է, երբ ապագա իրադարձությունների հերթը վերջանում է կամ վերջանում է տրված մոդելավորման ժամանակը:

Ասինխրոն մոդելավորման արդյունքները ներկայացվում են ժամանակային դիագրամների տեսքով, որոնցով վերլուծվում է սարքավորումների աշխատանքը, օրինակ ՝  հայտնաբերվում են կրիտիկական մրցակցությունները, ստատիկ և դինամիկ խափանման ռիսկերը:

Խափանման ռիսկերը և կրիտիկական մրցումները ավելի արագ են հայտնաբերվում տրամաբանական մոդելավորման բազմարժեք մեթոդի օգնությամբ:

2.2.4 Բազմարժեք մոդելավորման մեթոդը

Բազմարժեք մոդելավորման մեթոդներից ամենատարածվածը եռարժեք մոդելավորումն է: Այդ դեպքում օգտագործվում են ազդանշանի երեք արժեքներ 0, 1, X, որտեղ X-ը դա անցումն է մի վիճակից մյուսը կամ անորոշ վիճակը:

Եռարժեք մոդելավորման առավելությունը՝ իրագործման պարզությունն է և բարձր արագագործությունը:

Եռարժեք մոդելավորման ժամանակ էլեմենտների հապաղումները ընդունվում են հավասար 0: Տրամաբանական էլեմենտի մոդելը ներկայացվում է ճշմարտացիության աղյուսակների ձևով, որտեղ ամեն մի մուտքային ազդանշանը կարող է ընդունել երեք արժեքներ:

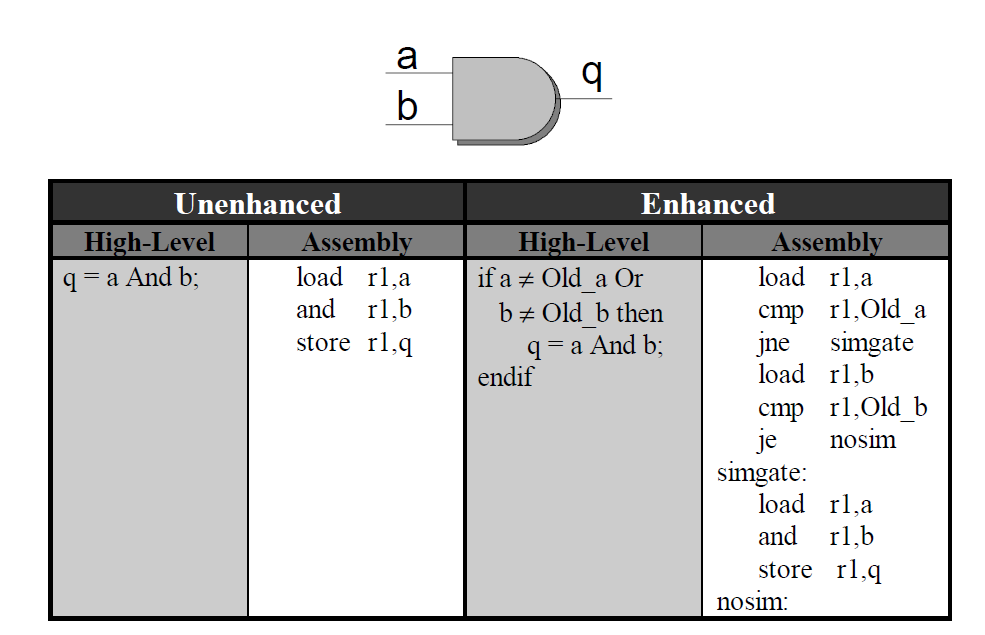
Հետադարձ կապերով սխեմաների եռարժեք մոդելավորման ժամանակ տրվում են հետադարձ կապերի ազդանշանների սկզբնական արժեքները (եթե նրանք անհայտ են, ապա ընդունվում են հավասար X արժեքին), հետադարձ կապերը խզվում են և իրագործվում է էլեմենտների ռանգավորումը: Այնուհետև հաշվվում են սխեմայի տրամաբանական էլեմենտների ելքերի արժեքները տրված մուտքային ազդեցության համար (անցողիկ կամ նոր ազդեցության) և որոշվում են հետադարձ կապերի ազդանշանների նոր արժեքները: Մոդելավորման պրոցեսը ամեն մի մուտքային ազդեցության համար իրականացվում է բազմակի անգամ մինչև կայուն վիճակի կամ գեներացիայի ռեժիմի հաստատումը:

Եթե եռարժեք մոդելավորման ժամանակ հայտնաբերվել է, որ սխեմայում հետադարձ կապի գոնե մի ազդանշան ընդունում է X արժեքը, ապա սխեմայում կան կրիտիկական մրցումներ:

Այսպիսով, բազմարժեք մոդելավորումը թույլ է տալիս հայտնաբերել հապաղման ռիսկի բոլոր հնարավոր տեղերը, նույնիսկ նրանց, որոնք իրական սխեմաներում չեն կարող պատահել:

2.3. Իրադարձային տրամաբանական մոդելավորում

Եթե տրամաբանկան փականի մուտքերը չեն փոխվում, կարիք չկա այն կրկին սիմուլացնելու։ Այս գաղափարը առաջ է քաշում սիմուլացիայի արագությունը մեծացնելու նոր մեթոդ։ Այն է քչացնել սիմուլացվող էլեմենտների քանակը մուտքային վեկտորով սիմուլացնելիս։ Ցավոք, հասարակ մոտեցումը չի բերի լուրջ կատարելագործման, քանի որ փականի մուտքի փոփոխության ստուգման ժամանակը գրեթե հավասար է փականը սիմուլացնելու ժամանակին։ Նկար 1-ը ներկայացնում է տարբերությունը՝

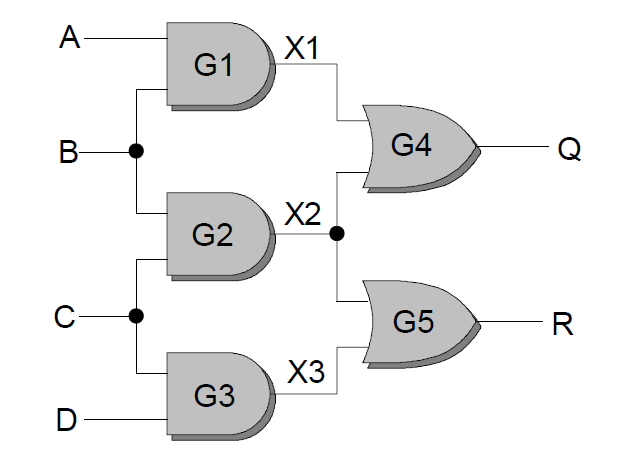


**Նկար 1**

Լավացրած կոդը աշխատեցնում է ամենաքիչը 6 հրաման, բայց կարող է հասնել նաև 9-ի, իսկ սովորականը՝ միշտ 3 հրաման է:

Եթե փականի մուտքեևը չեն փոխվում, ապա ելքը ևս չի փոխվի։ Նկար 1-ի պարագայում, եթե ոչ a-ն, ոչ b-ն չփոխվեն, q-ն չի փոխվի։ Եթե q-ն ինչ-որ ուրիշ սխեմայի մուտք է, ապա կարիք չկա այդ մուտքը ստուգել, այնքան ժամանակ, մինչև a-ն կամ b-ն չփոխվեն։

Գաղափարը ավելի լավ հասկանալու համար դիտարկենք նկար 2-ի սխեման՝

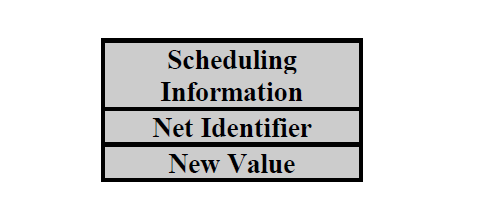


**Նկար 2**

Ենթադրենք սխեման սիմուլացվում է երկու տարբեր մուտքային վեկտորներով՝(0,0,0,0) և (0,0,0,1); Քանի որ A-ն, B-ն և C-ն չեն փոխվել, կարիք չկա սիմուլացնելու G1,G2, և G3 փականները։ G4-ի սիմուլացնելը կարելի է անտեսել, առանց ստուգելու x1 և x2-ը։ Ցավոք, կոդը այս ամենը իրականացնելու համար շատ բարդ կլինի գրեl, հատկապես B,C, և X1 ճյուղավորումների հատվածը։ Ոչ անհրաժեշտ փականների հեռացումը սիմուլացիայի պրոցեսից կարիք ունի հիմնովին այլ մոտեցման;

2.3.1 Հիմնական գաղափարները

Իրադարձային տրամաբանական մոդելավորումը նախատեսված է ավելորդ փականների սիմուլյացման հեռացման համար, առանց անընդունելի քանակությամբ լրացուցիչ թեստավորման: Իրադարձության գաղափարը ամենահիմնականն է այս պրոցեսի ընթացքում։ Իրադարձությունը իրենից ներկայացնում է կապակցող մետաղական միացման արժեքի փոփոխությունը։ Ւրադարձությունը կարելի է ներկայացնել հետեվյալ տվյալների կառուցվածքի միջոցով, ինչպես ներկայացված է նկար 3-ում ՝



**Նկար 3**

Իրադարձային տրամաբանական մոդելավորման հիմնական ֆունկցաին իրադարձությունը հայտնաբերելն է և դրա հիման վրա համապատասխան փականների սիմուլացիա կատարելը։

Եթե իրադարձություններ չկան, ուրեմն չկան փոփոխված միացումներ, ուստի նաև սիմուլացվող փականներ։

Իրադարձային տրամաբանական մոդելավորումը իրենից ներկայացնում է դինամիկ պլանամորման գործընթաց։ Այսինքն այն կարող է պլանավորել փականների սիմուլացիա, առանց իմանալու նախնական հաջորդակարգի սխեմայի վերլուծման ընթացքում։ Սրա պատճառով մեկ կամ մի քանի դինամիկ հերթեր կարող են օգտագործվել իրագործման ընթացում։

Երկու անգամ է անհրաժեշտ լինում ստուգել միացումների արժեքների փոփոխությունները՝ երբ նոր մուտքային վեկտոր է տրվում և անմիջապես փականի սիմուլյացիայից հետո։

Հիմնական ալգորիթմը կարելի է նկարագրել հետեվյալ կերպ՝

* Սկզբնարժեքավորել հիմնական մուտքերը մուտքային վեկտորի արժեքներով
* Միացումների հերթի մեջ ավելացնել հիմնական մուտքերը
* Քանի դեռ միացումների հերթը դատարկ չէ
  + Պտտվել միացումների հերթի վրայով, հանել մի էլեմենտ հերթից
  + Փականների ցուցակից վերցնել հերթական փականը
  + Եթե միացման էլեմենտը հանդիսանում է մուտք հանված փականի համար, փականը ավելացնել փականների հերթի մեջ
  + Քանի դեռ փականների հերթը դատարկ չէ, պտտվել փականների հերթի վրայով, հերթով սիմուլացնելով փականները
  + Սիմուլացվող փականի փոփոխված ելքը ավելացնել միացումների հերթի մեջ

Ծրագրավորման լեզուներին հարմար պսեվդո ներկայացումը հետեվյալն է՝

# setting primery\_inputs values from input\_vector

index = 0

for net in net\_list:

if net.name in prim\_ins:

net.old\_value = net.value

net.value = input\_vector[index]

index+=1

# putting changed nets to net\_queue

for net in net\_list:

if net.value != net.old\_value:

net\_queue.put(net)

# main process of simulation

while not net\_queue.empty():

#finding gates,whose inputs are the changed nets

while not net\_queue.empty():

temp\_net = net\_queue.get()

for gate in gate\_list:

if temp\_net in gate.input\_list:

if gate.in\_queue == False:

gate.in\_queue = True

gate\_queue.put(gate)

# simulation process

while not gate\_queue.empty():

temp\_gate = gate\_queue.get()

temp\_gate.in\_queue = False

temp\_gate.simulate()

if temp\_gate.output.old\_value != temp\_gate.output.value:

net\_queue.put(temp\_gate.output)