ՀԱՅԱՍՏԱՆԻ ՊԵՏԱԿԱՆ ՃԱՐՏԱՐԱԳԻՏԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

Հաշվողական տեխնիկայի ամբիոն

Էլեկտրոնային հաշվիչ մեքենաների կազմակերպում

Կուրսային աշխատանքի մեթոդական ցուցումներ

ԵՐԵՎԱՆ 2005

ՀՏԴ 681.3

Կազմողներ`   Թումանյան Ա.Կ.
Կիրակոսյան Գ.Տ.
Համազասպյան Ա.Հ.
Սաղաթելյան Ա.Կ.

Էլեկտրոնային հաշվիչ մեքենաների կազմակերպում: Կուրսային աշխատանքի մեթոդական ցուցումներ: Հեղինակներ` Ա.Կ. Թումանյան, Գ.Տ. Կիրակոսյան, Ա.Հ. Համազասպյան, Ա.Կ. Սաղաթելյան: Հայաստանի պետական ճարտարագիտական համալսարան. Երևան, 2005թ., 56 էջ:

Գրախոս`   Ռ.Պ. Պետրոսյան

“ԷՀՄ կազմակերպում” առարկայի կուրսային աշխատանքի մեթոդական ցուցումները նախատեսված են ՔՀ և Ի դեպարտամենտի երկրորդ կուրսի բոլոր ուսանողների համար:

Խմբագիր`   Ն.Ա. Խաչատրյան

Բ Ո Վ Ա Ն Դ Ա Կ ՈՒ Թ Յ ՈՒ Ն

1. Կուրսային աշխատանքի նկարագրությունը

Կուրսային աշխատանքը նվիրված է տրամաբանական սխեմաների նախագծմանը:

«Տրամաբանական նախագծում» թեմայով կուրսային աշխատանքի կատարման նպատակն է ուսանողին պարզ տրամաբանական սխեմաների նախագծման մեթոդիկային ծանոթացնելը և այս բնագավառում գործնական գիտելիքներ տալը:

Տրամաբանական սխեմաները բաժանվում են երկու դասի՝ «կոմբինացիոն» և «հաջորդականային (ավտոմատներ)»:

Կոմբինացիոն է համարվում այն տրամաբանական սխեման, որի ելքային ազդանշանները կախված են միայն մուտքային ազդանշանների ընթացիկ արժեքներից:

Յաջորդականային (ավտոմատ) է կոչվում այն տրամաբանական սխեման, որի ելքային ազդանշանները որոշվում են ոչ միայն մուտքային ազդանշանների ընթացիկ արժեքներով, այլև կախված են մուտքային ազդանշանների արժեքների նախկին հաջորդականությունից:

Կուրսային աշխատանքը կազմված է երկու մասից՝

- կոմբինացիոն սխեմայի սինթեզում,
- վերջավոր ավտոմատի սինթեզում:

Առաջադրանքների տարբերակները համարակալված են երեք նիշերով: Առաջին նիշը որոշում է հանձնարարության մասը. կոմբինացիոն սխեմայի սինթեզում (համապատասխանում է 1), թե վերջավոր ավտոմատի սինթեզում (համապատասխանում է 2): Երկրորդ նիշը որոշում է առաջադրանքների տեսակները: Երրորդ նիշը՝ առաջադրանքի տարբերակը տվյալ տեսակի սահմաններում: Օրինակ՝ առաջադրանք 1.3.7: Պահանջվում է նախագծել յոթ սեգմենտանի թվային ինդիկատորի ղեկավարման սխեմա, որի մուտքային կոդը 731(-2) է:

Ուսանողին առաջարկվում է ներքոհիշյալ առաջադրանքների տարբերակներից մեկը: Պահանջվում է որոշել տվյալ սարքը նկարագրող բուլյան ֆունկցիաների համակարգը, կատարել ստացված ֆունկցիաների մինիմացում և նախագծել կոմբինացիոն սխեման հետևյալ ձևերով՝

- սխեմայի կառուցումը "**ԵՎ**", "**ԿԱՄ**" և "**ՈՉ**" բազիսում;
- անցումը "**ԵՎ-ՈՉ**" ու "**ԿԱՄ-ՈՉ**" բազիսների և կոմբինացիոն սխեմաների կառուցումը այդ բազիսներում;
- իրագործվող ֆունկցիաների ժեգալկինի բազմանդամի որոշումը;
- սարքի սկզբունքային էլեկտրական սխեմայի կառուցումը SSS (տրանզիստորա-տրանզիստորային տրամաբանություն) կամ

ԿՄՕԿ (կոմպլեմենտար մետաղ-օքսիդ-կիսահաղորդիչ) սերիաների միկրոսխեմաների վրա;

- իրագործվող ֆունկցիաների կառուցումը վերծանիչի վրա;
- իրագործվող ֆունկցիաների կառուցումը մուլտիպլեքսորների վրա:

2. Կոմբինացիոն սխեմաների սինթեզում

Առաջադրանքների տեսակներ

Առաջադրանք 1.1.

Նախագծել սխեմա, որը երկուական-տասական կոդով տրված մուտքային տասական թիվը ձևափոխում է երկուական-տասական կոդով ներկայացված ելքային թվի:

Մուտքային և ելքային երկուական-տասական կոդերի դիրքային կշիռները բերված են աղյուսակ 1-ում:

Աղյուսակ 1

Տարբերակի համարը	Մուտքային կոդը p_1 p_2 p_3 p_4	Ելքային կոդը p_1' p_2' p_3' p_4'
1	8 4 2 1	7 4 2 1
2	7 4 2 1	5 4 2 1
3	7 4 2 1	8 4 2 1
4	7 4 2 1	2 4 2 1
5	5 4 2 1	7 4 2 1
6	5 3 2 1	8 4 2 1
7	5 3 2 1	7 4 2 1
8	5 3 2 1	2 4 2 1
9	5 2 1 1	5 3 2 1
10	7 5 3 -6	3 3 2 1
11	6 3 1 -1	5 3 2 -1
12	5 3 2 -1	2 4 2 1

Առաջադրանք 1.1-ի պարզաբանումը:

p_1 p_2 p_3 p_4 դիրքային կշիռներով երկուական-տասական կոդով ներկայացված նիշերի արժեքները որոշվում են

$X = x_1 \cdot p_1 + x_2 \cdot p_2 + x_3 \cdot p_3 + x_4 \cdot p_4 = \Sigma x_i \cdot p_i$ բանաձևով, որտեղ p_i–ն տետրադի (քառյակի) i-րդ կարգի կշիռն է:

Ելքային ազդանշանները հաշվարկվում են նույն սկզբունքով.

$Y = y_1 \cdot p_1' + y_2 \cdot p_2' + y_3 \cdot p_3' + y_4 \cdot p_4' = \Sigma y_i \cdot p_i'$

<u>Առաջադրանք 1.2.</u>

Նախագծել սարք, որը դիրքերի կշիռներով տրված թվերի մուտքային երկուական-տասական կոդը (աղյուսակ 2) կձևափոխի աղյուսակ 3-ում բերված որև է մեկ ելքային կոդի:

Աղյուսակ 2

Տարբերակի համարը	Մուտքային կոդ p_1 p_2 p_3 p_4	Ելքային կոդ
1	8 4 2 1	3-ի ավելցուկով կոդ
2	8 4 2 1	Կոդ 2-ը 5-ից
3	8 4 2 1	Երկուական–հնգական կոդ
4	2 4 2 1	3-ի ավելցուկով կոդ
5	2 4 2 1	Կոդ 2-ը 5-ից
6	2 4 2 1	Երկուական–հնգական կոդ
7	5 2 1 1	3-ի ավելցուկով կոդ
8	5 2 1 1	Կոդ 2-ը 5-ից
9	5 2 1 1	Երկուական–հնգական կոդ
10	4 3 1 1	3-ի ավելցուկով կոդ
11	4 3 1 1	Կոդ 2-ը 5-ից
12	4 3 1 1	Երկուական–հնգական կոդ

Աղյուսակ 3

Տասական թիվ	3-ի ավելցուկով կոդ	Երկուական – հնգական կոդ	Կոդ 2-ը 5-ից
0	0011	01 00001	00011
1	0100	01 00010	00101
2	0101	01 00100	00110
3	0110	01 01000	01001
4	0111	01 10000	01010
5	1000	10 00001	01100
6	1001	10 00010	10001
7	1010	10 00100	10010
8	1011	10 01000	10100
9	1100	10 10000	11000

Առաջադրանք 1.2-ի պարզաբանումը:

Տասական թվերի կոդավորման համար, բացի կշիռներով տրված կոդերից, օգտագործվում են նաև այլ կոդեր. 3-ի ավելցուկով, 2-ը 5-ից և երկուական–հնգական կոդերը: Այս կոդերը բերված են աղյուսակ 3-ում:

3-ի ավելցուկով կոդն ինքնալրացվող է (self-complemented): Դա նշանակում է, որ ցանկացած տասական թվանշանի հակադարձ կոդը կարելի է ստանալ՝ ինվերսելով երկուական-տասական կոդում տվյալ թվանշանի բիթերը: Օրինակ՝ 3 թվանշանի լրացումը մինչև 9 (այսինքն՝ լրացուցիչ կոդը) 6 թվանշանն է: 3-ի կոդը 0110 է, իսկ 6-ինը՝ 1001: Ինքնալրացվող կոդի ուրիշ օրինակ է 2421 կշիռներով կոդը՝ 3-ի կոդը – 0011, իսկ 6-ինը – 1100: Ինքնալրացվող կոդերը հարմար է օգտագործել՝ տասական համակարգում թվերի գումարում և հանում կատարելիս:

Երկուական-հնգական կոդը ձևավորվում է հետևյալ կերպ. կոդային բառում առաջին երկու բիթերը ցույց են տալիս, թե երկու միջակայքերից որին է պատկանում թիվը՝ 0-ից մինչև 4, թե՝ 5-ից մինչև 9: Վերջին հինգ բիթերը ցույց են տալիս, թե տվյալ միջակայքի հինգ թվերից որ թիվն է ներկայացված այդ բառով:

Երկուական-հնգական և 2-ը 5-ից կոդերն ունեն սխալների բացահայտման հնարավորության առավելությունը: Կոդային բառում ցանկացած մեկ բիթի պատահական փոփոխումը բերում է նրան, որ ստացված կոդային բառը իրենից ոչ մի տասական թիվ չի ներկայացնում: Այդպիսի բառը պետք է համարել սխալ:

<u>Առաջադրանք 1.3.</u>

Յոթսեգմենտանի թվային ինդիկատորի (seven-segment display) ղեկավարման սխեմայի նախագծումը (տարբերակները՝ աղյուսակ 4-ում):

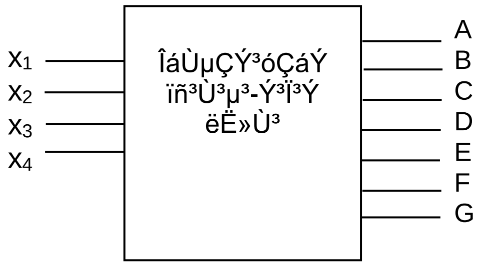
Աղյուսակ 4

Տարբերակի համարը	Երկուական-տասական կոդ p_1 p_2 p_3 p_4		Տարբերակի համարը	Երկուական-տասական կոդ p_1 p_2 p_3 p_4
1	8 4 2 1		7	7 3 1 -2
2	7 4 2 1		8	6 4 2 -1
3	5 3 2 1		9	5 3 2 -1
4	5 2 1 1		10	4 3 1 1
5	3 3 2 1		11	6 3 1 -1
6	2 4 2 1		12	7 5 3 -6

Առաջադրանք 1.3-ի պարզաբանումը:

Այս տեսակի ինդիկատորները, որոնք հիմնված են լուսադիոդների կամ հեղուկ բյուրեղների վրա, կիրառվում են հաշվիչներում, ժամացույցներում ու չափիչ սարքերում և նախատեսված են տասական համակարգում տվյալների արտապատկերման համար:

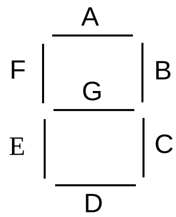
Յոթսեգմենտանի թվային ինդիկատորի ղեկավարման համար օգտագործվում է կոմբինացիոն տրամաբանական սխեմա, որն ունի 4 մուտք և 7 ելք (նկար 1):



Նկար 1. Յոթսեգմենտանի թվային ինդիկատորի ղեկավարման կոմբինացիոն տրամաբանական սխեմա

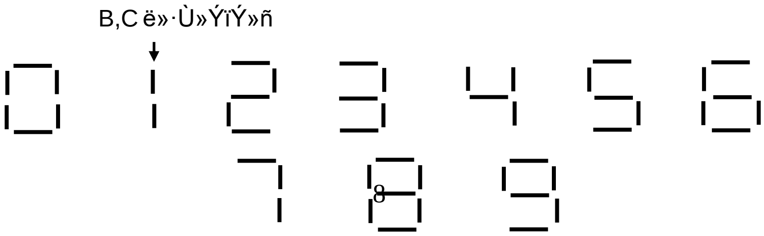
Սխեմայի յոթ ելքերը միացված են լուսադիոդներից կազմված սեգմենտների ղեկավարող էլեկտրոդներին, որոնք լուսավորվում են մուտքին տրամաբանական “1” ազդանշանի գալու դեպքում: Թիվը գոյանում է կոդի որոշակի սեգմենտների լուսավորման արդյունքում:

Յոթսեգմենտանի կոդը ներկայացված է նկար 2-ում:



Նկար 2. Յոթսեգմենտանի կոդի ներկայացում

Յուրաքանչյուր թիվ արտահայտող ակտիվ սեգմենտները ներկայացված են նկար 3-ում:



Նկար 3 . Յուրաքանչյուր թիվ արտահայտող ակտիվ սեգմենտների ներկայացում

<u>**Առաջադրանք 1.5.**</u>

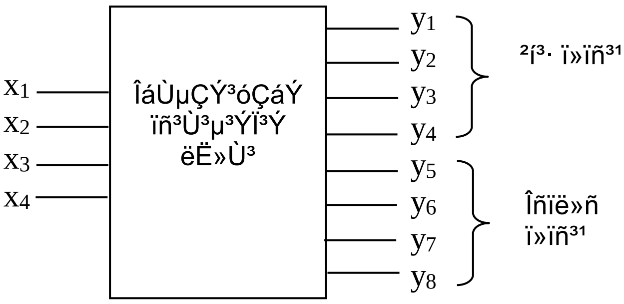
Նախագծել սխեմա ամբողջ երկկարգանի առանց նշանի թվերի բազմապատկման համար:

<u>**Առաջադրանք 1.6.**</u>

Նախագծել սխեմա, որը կձևափոխի քառաբիթ երկուական կոդը ելքային երկուական-տասական կոդի:

Առաջադրանք 1.6-ի պարզաբանումը:

Տասական կարգերի թիվը հավասար է երկուսի: Երկուական կոդը երկուական-տասականի ձևափոխող սարքն ունի 4 մուտք և ութ ելք: Այս սարքի կառուցվածքը բերված է նկար 4-ում:



Նկար 4. Մուտքային երկուական կոդը ելքային երկուական-տասական կոդի ձևափոխիչ

Տետրադ է կոչվում է քառաբիթ երկուական կոդը, որը համապատասխանեցված է յուրաքանչյուր տասական նիշին:

Մուտքային կոդի դիրքային կշիռները բերված են աղյուսակ 5-ում:

Աղյուսակ 5

Տարբերակի համարը	Կոդի նիշերի կշիռները p_1 p_2 p_3 p_4
1	8 4 2 1
2	7 4 2 1
3	5 3 1 1
4	5 2 1 1
5	7 3 1 -2
6	6 4 2 -1
7	5 3 2 -1
8	4 3 1 1
9	6 3 1 -1
10	7 5 3 -6
11	5 4 2 1
12	3 3 2 1

3. Վերջավոր ավտոմատի սինթեզում

Առաջադրանքների տեսակներ

Առաջադրանք 2.1.

Կառուցել առաջադրանքի տարբերակում նշված մուտքային հաջորդականության դետեկտորի սխեման: Միլի ավտոմատի մուտքին տվյալ հաջորդականության տրման դեպքում նրա ելքում գոյանում է «1» ազդանշանը: Մնացած բոլոր դեպքերում՝ «0» ազդանշանը: Առաջադրանքների տարբերակները բերված են աղյուսակ 6-ում:

Աղյուսակ 6

Տարբե-րակի համարը	Մուտքային հաջորդա-կանություն		Տարբե-րակի համարը	Մուտքային հաջորդա-կանություն
1	11011		7	000110
2	10001		8	01100
3	010101		9	100011
4	10010		10	001111
5	001010		11	101010
6	111011		12	101101

Առաջադրանք 2.2.

Կառուցել Մուրի ավտոմատի սխեման, որի վիճակների փոփոխ-ման հաջորդականությունները բերված են աղյուսակ 7-ում: «Ելքային ազդանշան» սյունում թվարկված են վիճակները, որոնցում գտնվելու դեպքում ավտոմատի ելքում ստացվում է «1» ազդանշանը, իսկ մնացած բոլոր դեպքերում՝«0» ազդանշանը:

Աղյուսակ 7

Տարբերակի համարը	Վիճակների հաջորդականություն	Ելքային ազդանշան
1	X=1. 0→1→3→5→4→0 † X=0. 0→5→4→3→1→0 †	1 v 5
2	X=1. 0→2→3→5→4→0 † X=0. 0→3→2→4→5→0 †	3 ∨ 4
3	X=1. 0→1→5→3→2→0 † X=0. 0→2→3→5→1→0 †	2 ∨ 3
4	X=1. 0→5→1→2→4→0 † X=0. 0→4→2→1→5→0 †	4 v 5
5	X=1. 0→7→6→2→1→0 † X=0. 0→6→1→7→2→0 †	2 v 6
6	X=1. 0→7→1→6→3→0 † X=0. 0→3→6→1→7→0 †	3 v 7
7	X=1. 0→6→4→5→1→0 † X=0. 0→1→5→4→6→0 †	4 v 6
8	X=1. 0→2→1→7→4→0 † X=0. 0→7→1→4→2→0 †	4 v 7
9	X=1. 0→2→5→4→1→0 † X=0. 0→5→1→2→4→0 †	1 v 5
10	X=1. 0→2→3→6→1→0 † X=0. 0→6→1→3→2→0 †	1 v 3
11	X=1. 0→4→5→6→1→0 † X=0. 0→6→1→5→4→0 †	1 v 4 v 5
12	X=1. 0→3→2→7→5→0 † X=0. 0→5→7→2→3→0 †	2 v 3 v 7

<u>Առաջադրանք 2.3.</u>

Երկուական-տասական թվերի տարբեր կոդավորման համար կառուցել երկուական-տասական սինքրոն հաշվիչի սխեման: Հաշվիչը գտնվում է **զրոյից - ինը** վիճակներից մեկում (տարբերակները աղյուսակ 8-ում):

Աղյուսակ 8

Տարբերակի համարը	Երկուական-տասական կոդ	Հաշվի ուղղությունը
1	3-ի ավելցուկով կոդ	Up
2	3-ի ավելցուկով կոդ	Down
3	Կոդ 2-ը 5-ից	Up
4	Կոդ 2-ը 5-ից	Down
5	Երկուական-հինգական կոդ	Up
6	Երկուական-հինգական կոդ	Down
	Կոդեր ըստ կշիռների	
7	2 4 2 1	Up
8	2 4 2 1	Down
9	6 3 1 -1	Up
10	6 3 1 -1	Down
11	6 4 2 -1	Up
12	7 5 3 –6	Up

<u>**Առաջադրանք 2.4.**</u>

Կառուցել կոդային փականը ղեկավարող, մեկ X մուտք և Y_1 ու Y_2 ելքեր ունեցող Միլի ավտոմատի սխեման:

Ելքային Y_1 ազդանշանը հավասար է մեկի այն դեպքում, երբ X=0, իսկ նախորդ հինգ տակտերի ընթացքում X մուտքին տրվել է աղյուսակ 10-ով առաջադրված մեկերի և զրոների հաջորդականությունը:

Ելքային Y_2 ազդանշանը պետք է հավասար լինի մեկի միայն այն դեպքում, երբ X-ի ընթացիկ արժեքը ճիշտ է ավտոմատի` դեպի փականը բաց լինելու վիճակին շարժման (առաջընթացի) տեսանկյունից:

Առաջադրանքների տարբերակները բերված են աղյուսակ 9-ում:

Աղյուսակ 9

Տարբերակի համարը	Մուտքային հաջորդակ.	Տարբերակի համարը	Մուտքային հաջորդակ.
1	11011	7	01001
2	01010	8	11001
3	10101	9	00100
4	00011	10	01100
5	10011	11	100111

6	10110		12	110001

3.1. Ավտոմատի սինթեզման հաջորդականությունը

Կուրսային աշխատանքում ավտոմատի սինթեզման հաջորդականու-թյունը պետք է իրականացնել ըստ հետևյալ քայլերի.

1. Տրված նկարագրությամբ կառուցել ավտոմատի անցումների գրաֆը:
2. Ընտրել ավտոմատի իրագործման համար տրիգերի տեսակը:
3. Կոդավորել ավտոմատի վիճակները:
4. Կառուցել ավտոմատի անցումների կոդավորված աղյուսակը:
5. Որոշել ավտոմատի գրգռման և ելքային ֆունկցիաները:
6. Մինիմացնել ստացված տրամաբանական ֆունկցիաները:
7. Կառուցել ավտոմատի սխեման ԵՎ, ԿԱՄ, ՈՉ բազիսում:
8. Կառուցել ավտոմատի սկզբունքային սխեման տրված սերիայի միկրոսխեմաներով և կազմել միկրոսխեմաների որակավորման աղյուսակը:

4. Համառոտ տեսական մաս

4.1. Բուլյան ֆունկցիաների հասկացությունը

Բուլյան են կոչվում այն ֆունկցիաները ($f(x_1, x_2, {\dagger} x_n)$), որոնք, ինչպես և նրանց արգումենտները, կարող են ընդունել երկու արժեք՝ «0» և «1»:

Բոլոր **n** արգումենտներից $f(x_1, x_2, {\dagger} x_n)$ֆունկցիան որոշված է $\mathbf{2^n}$ հավաքածուներում կամ կետերում: Երբեմն այդ հավաքածուների բազմությունը կոչվում է բուլյան տարածություն, իսկ այդ տարածության յուրաքանչյուր տարր՝ բուլյան վեկտոր:

Ֆունկցիան կոչվում է ամբողջովին որոշված, եթե ցանկացած հավաքածուի համար հայտնի է նրա արժեքը: Իսկ եթե որոշ հավաքածուներում ֆունկցիայի արժեքները որոշված չեն, ապա ֆունկցիան կոչվում է ոչ լրիվ կամ մասնակի որոշված:

Կան ընդամենը **n** փոփոխականների $\mathbf{2^n}$ աստիճանի 2 տարբեր ֆունկցիաներ, այդ թվում «0»-ի և «1»-ի հաստատունները և տրիվիալ ֆունկցիաները:

4.1.2. Բուլյան ֆունկցիաների տրման ձևերը

Բուլյան ֆունկցիաների տրման 2 հիմնական ձևերն են՝ **աղյուսակային** և **անալիտիկ**:

<u>Ֆունկցիայի տրման աղյուսակային ձև</u>

Ֆունկցիաների առաջադրման համար օգտագործվում է $\mathbf{2^n}$ տող պարունակող աղյուսակ: Յուրաքանչյուր տողին համապատասխանում է արգումենտների հավաքածուն և ֆունկցիայի արժեքը՝ ըստ տվյալ հավաքածուի (աջ սյունակում):

Հավաքածուները կարգավորված են աճման կարգով (աղյուսակ 10):

Աղյուսակ 10

$x_1\ x_2\ .\ .\ .\ x_n$	$f(x_1,x_2,\ .\ .\ .\ .x_n\ )$
0 0 . . . 0 0	0
0 0 . . . 0 1	1
0 0 . . . 1 0	1
. . .	. . .
1 1 . . ..1 1	1

<u>Ֆունկցիայի անալիտիկ առաջադրումը, Շեննոնի վերլուծումը</u>

<u>Վերլուծման վերաբերյալ թեորեմ</u>.

Ըստ x_1 մեկ փոփոխականի $f(x_1, x_2,\dagger\ x_n)$ ֆունկցիայի Շեննոնի վերլուծում կոչվում է նրա պատկերումը հետևյալ տեսքով.

$$f(x_1, x_2,\dagger x_n) = \overline{x_1}\ f(0,x_2, . . .x_n)\ V\ x_1\ f(1, x_2,. . . x_n)$$

Այս թեորեմն ապացուցվում է $x_i=0$ և $x_i=1$ արգումենտների արժեքների տեղադրման միջոցով:

Ըստ բոլոր փոփոխականների վերլուծումը (ընդհանրացված թեորեմ).

$$f(x_1, x_2,\dagger x_n) = \overline{x_1}\ \overline{x_2}\ .\ .\ .\ \overline{x_n}\ f\ (0,0, . . 0)\ \ v\ \ \overline{x_1}\ \overline{x_2}\ .\ .\ .\ \overline{x_n}\ f\ (0,0, . . .1)\ \ v$$
$$v\ .\ .\ .x_1\ x_2\ \ .\ .\ .x_n\ f\ (1,1, . . . 1)$$

Բուլյան ֆունկցիան անալիտիկ եղանակով նկարագրելու համար հաճախ օգտագործում են **ներկայացման նորմալ ձևերը**:

Ցանկացած բուլյան ֆունկցիա կարելի է ներկայացնել որպես տերմ-արտադրյալների դիզյունկցիա, որոնցից յուրաքանչյուրը բոլոր փոփոխականների կոնյունկցիան է (բացասումներով կամ առանց դրանց), և այդ ֆունկցիայի արժեքները՝ փոփոխականների համապատասխան հավաքածուում:

$\sigma \in \{0,1\}$-ի համար նշանակենք

$$x^\sigma = \begin{cases} \overline{x}, \text{ եթե } \sigma = 0 \\ x, \text{ եթե } \sigma = 1 \end{cases}$$

Այս դեպքում ցանկացած բուլյան ֆունկցիա կարելի է ներկայացնել հետևյալ տեսքով.

$$f(x_1, x_2, \dagger x_n) = \bigvee_{f(\sigma 1, \sigma 2, \ldots \sigma n)=1} x_1^{\sigma 1} \cdot x_2^{\sigma 2} \cdot \ldots \cdot x_n^{\sigma n}: \qquad (1)$$

σi – x_i արգումենտի արժեքն է արգումենտների հավաքածուում:

Քանի որ յուրաքանչյուր որոշակի հավաքածուում ֆունկցիայի արժեքը կամ «0» է, կամ «1», ապա բանաձևում կմնան միայն այնպիսի տերմեր, որոնք համապատասխանում են փոփոխականների այն հավաքածուներին, որոնց համար ֆունկցիան հավասար է «1»-ի: Ներկայացման այդպիսի ձևը կոչվում է կատարյալ դիզյունկտիվ նորմալ ձև (**ԿԴՆՁ**):

$$f(x_1, x_2, \dagger x_n) = K_1 \vee K_2 \vee \ \ldots \ \vee K_m, \ \ m \geq 1,$$

որտեղ յուրաքանչյուր տերմ K_i (կոնյունկտ) ներկայացնում է ֆունկցիայի փոփոխականների ժխտված (ինվերս) կամ ուղիղ ձևով վերցրած կոնյունկցիան: Բոլոր կոնյունկցիաներում տառերի քանակը հավասար է **n**: Երբեմն **K**$_i$–ն անվանում են մեկի կոնստիտուենտ կամ մինտերմ (ֆունկցիայի արժեքը որոշվում է առաջնային տերմերի նվազագույն արժեքով): Եթե որոշ կոնյունկցիաներ պարունակում են n-ից ավելի քիչ քանակով տառեր, ապա դա դիզյունկտիվ նորմալ ձև է (**ԴՆՁ**):

ԿԴՆՁ – ում բուլյան ֆունկցիաների ներկայացումը միակն է:

Նկարագրենք կոնյունկտիվ նորմալ ձևը (**ԿՆՁ**):

x_i^σ-ը առաջնային տերմն է:Առաջնային տերմերի սահմանումից բխում է, $x_i^1 = x_i^0 = x_i$ իսկ $\overline{x_i^0} = \overline{x_i^1} = x_i$

Դե Մորգանի բանաձևերը կիրառենք արտահայտություն (1)-ի նկատմամբ: Կստանանք.

$$f(x_1, x_2, \ldots, x_n) = \overline{\overline{\bigvee_{f(\sigma 1, \sigma 2, \ldots \sigma n)=1} x_1^{\sigma 1} \cdot x_2^{\sigma 2} \cdot \ldots \cdot x_n^{\sigma n}}} =$$

$$= \overline{\&_{f(\sigma 1, \sigma 2, \ldots \sigma n)=1} (\overline{x_1^{\sigma 1}} \vee \overline{x_2^{\sigma 2}} \vee \ldots \overline{x_n^{\sigma n}})} \qquad \text{կամ}$$

$$\overline{f(x_1, x_2, \ldots, x_n)} = \&_{f(\sigma 1, \sigma 2, \ldots \sigma n)=1} (x_1^{\overline{\sigma 1}} \vee x_2^{\overline{\sigma 2}} \vee \ldots \vee x_n^{\overline{\sigma n}}).$$

$$f(x_1, x_2, \ldots, x_n) = \&_{f(\sigma 1, \sigma 2, \ldots \sigma n)=0} (x_1^{\overline{\sigma 1}} \vee x_2^{\overline{\sigma 2}} \vee \ldots \vee x_n^{\overline{\sigma n}}).$$

Ֆունկցիայի ներկայացումը հետևյալ տեսքով՝

$f(x_1,x_2,\ldots,x_n) = D_1 \cdot D_2 \cdot \ldots \cdot D_k$, որտեղ $k \geq 1$, կոչվում է կատարյալ կոնյունկտիվ նորմալ ձև (**ԿԿՆՁ**): Բոլոր դիզյունկտիվ տերմերը պարունակում են **n** լիտերալներ:

Դրանք կոչվում են զրոյի կոնստիտուենտ կամ մաքստրիմ (ֆունկցիայի արժեքը որոշվում է առաջնային տերմերի առավելագույն արժեքով): Եթե որոշ կոնյունկցիաներ պարունակում են n-ից ավելի պակաս լիտերալների քանակ, ապա դա **ԿՆՁ**– ն է:

Ֆունկցիաների ներկայացումը բաղմանդամների տեսքով

Ժեգալկինի թեորեմը: Ցանկացած բուլյան ֆունկցիա՝ $\mathbf{f(x_1, x_2, \ldots x_n)}$ կարելի է ներկայացնել բազմանդանի տեսքով:

$$\mathbf{f(x_1,x_2,\ldots x_n) = a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus \ldots \oplus a_nx_n \oplus a_{n+1}\, x_1x_2 \oplus \ldots}$$

$\mathbf{\oplus\, a_Nx_1x_2\ldots x_n}$, որտեղ $a_0, \ldots a_n$ – որոշ հաստատուններ են, որոնք հավասար են զրոյի կամ մեկի: a_0, a_1, †..a_N-ը$\in \{0,1\}$.

Որպես ապացույց, ենթադրենք, որ ֆունկցիան տրված է **ԿԴՆՁ**-ի տեսքով: Այս դեպքում մենք կարող ենք ԿԱՄ գործողությունը փոխարինել ըստ մոդուլ երկուսի գումարումով: Դիտարկենք օրինակի վրա:

$$\mathbf{f(x_1,x_2,x_3) = \overline{x}_1\overline{x}_2\overline{x}_3 \vee \overline{x}\,\overline{x}_2x_3 \vee \overline{x}_1x_2\overline{x}_3 \vee x_1x_2\overline{x}_3 =}$$

$$\mathbf{= (1 \oplus x_1)(1 \oplus x_2)(1 \oplus x_3) \oplus (1 \oplus x_1)(1 \oplus x_2)\,x_3 \oplus (1 \oplus x_1)\,x_2\,(1 \oplus x_3) \oplus}$$

$$\mathbf{\oplus\, x_1\,x_2\,(1 \oplus x_3) = 1 \oplus x_1 \oplus x_1x_2 \oplus x_2x_3.}$$

Ժեգալկինի հանրահաշիվը երկու բինար բուլյան ֆունկցիաների (ԵՎ,$\oplus$) բազմության և 1 հաստատունի հանրահաշիվն է:

$$\mathbf{x_1 \oplus x_2 = x_2 \oplus x_1,}$$

$$\mathbf{x_1(x_2 \oplus x_3) = x_1x_2 \oplus x_1x_3;}$$

$$\mathbf{x \oplus x = 0; \quad x \oplus 0 = x; \quad x \oplus 1 = \overline{x};}$$

4.1.3. Բուլյան ֆունկցիաների ֆունկցիոնալ լիակատար համակարգերը

Ֆունկցիոնալ-լիակատար հավաքածու (կամ բազիս) կոչվում է այն բուլյան ֆուկցիաների բազմությունը, որոնց վերադրումներով կարող են արտահայտված լինել ցանկացած բուլյան ֆունկցիաներ:

Վերադրում է կոչվում մի ֆունկցիայի՝ որպես արգումենտ մի ուրիշ ֆունկցիայում տեղադրումը:

Բուլյան ֆունկցիաների բազիսների որոշման հիմնախնդիրը լուծել է Պոստը: Պոստը առանձնացրել է փոխարկումային ֆունկցիաների հինգ դասեր, որոնցից յուրաքանչյուրը փակ է, այսինքն՝ տրված դասից յուրաքանչյուր ֆունկցիայի նկատմամբ կիրառելով վերադրման գործողությունը՝ ստանում ենք ֆունկցիա այդ նույն դասից:

Բուլյան ֆունկցիաների հինգ դասերն են.

1. Զրոն պահպանող ֆունկցիաներ:

f(x_1, x_2, †.x_n) ֆունկցիան պահպանում է զրոն, եթե f(0,0,††0)=0, զրոն պահպանող: Ֆունկցիայի օրինակ է կոնյունկցիան:

2. *Մեկը պահպանող ֆունկցիաներ:*

f(x_1, x_2,†.x_n) ֆունկցիան պահպանում է մեկը, եթե f(1, 1,†..1) =1:

3. Մոնոտոն ֆունկցիաներ:

f(x_1, x_2,†..x_n) ֆունկցիան կոչվում է մոնոտոն, եթե արգումենտների հավաքածուների աճման դեպքում նրա արժեքը չի նվազում: Կոնյունկցիան և դիզյունկցիան համարվում են մոնոտոն ֆունկցիայի օրինակ:

4. Ինքնաերկակի ֆունկցիաներ:

f(x_1, x_2,†..x_n) ֆունկցիան կոչվում է ինքնաերկակի, եթե հակառակ հավաքածուների յուրաքանչյուր զույգի համար այն ընդունում է հակառակ արժեքներ, այսինքն՝ ցանկացած հավաքածուի համար (σ_1, σ_2, †.σ_n), f(σ_1,σ_2,†.σ_n): Ինքնաերկակի ֆունկցիայի օրինակ է բացասումը:

5. *Գծային ֆունկցիաներ:*

Բուլյան ֆունկցիան կոչվում է գծային, եթե նրա՝ Ժեգալկինի բազմանդամն ունի հետևյալ տեսքը.

f(x_1, x_2, . . . x_n) = $a_0 \oplus a_1x_1 \oplus a_2x_2 \oplus \ldots \oplus a_nx_n$..

Ապացույցն այն է, որ այդ բոլոր դասերը փակ են, հիմնված են վերադրման և տեղադրման գործողությունների կիրառման վրա:

Պոստի թեորեմը.

Որպեսզի N բուլյան ֆունկցիաների բազմությունը լինի բազիս, անհրաժեշտ և բավարար է, որ.

1. N-ը պարունակի գոնե մեկ ֆունկցիա, որը չի պահպանում զրոն:
2. N-ը պարունակի գոնե մեկ ֆունկցիա, որը չի պահպանում մեկը:
3. N-ը պարունակի գոնե մեկ ոչ մոնոտոն ֆունկցիա:
4. N-ը պարունակի գոնե մեկ ոչ ինքնաերկակի ֆունկցիա:
5. N-ը պարունակի գոնե մեկ ոչ գծային ֆունկցիա:

Այս պայմանների անհրաժեշտությունն ու բավարարությունն ակնհայտ է:

Բուլյան ֆունկցիաների ֆունկցիոնալ լիարժեք բազիսների օրինակներ են.

- **"ԵՎ", "ԿԱՄ", "ՈՉ",**
- **"ԵՎ-ՈՉ"** (Շեֆֆերի գծիկ ֆունկցիան),
- **"ԿԱՄ-ՈՉ"** (Պիրսի սլաք ֆունկցիան),
- **"ԵՎ", ⊕ , հաստատուն 1**:

Բազիսում բուլյան ֆունկցիաների նվազագույն քանակը մեկ է ("**ԵՎ-ՈՉ**" կամ "**ԿԱՄ-ՈՉ**" բազիս):

Նվազագույն բազիսում բուլյան ֆունկցիաների առավելագույն քանակը չորս է:

4.1.4. Կոմբինացիոն սխեմաների սինթեզը

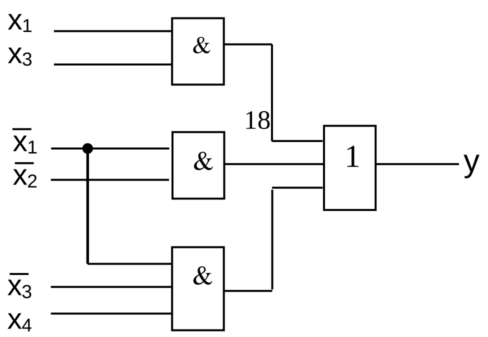
Կոմբինացիոն սխեման իրագործում է $F(x_1, x_2, \ldots x_n)$ բուլյան ֆունկցիան, եթե նրա ելքը մեկ է մեկին համապատասխան յուրաքանչյուր մուտքային հավաքածուի համար և զրո՝ զրոյին համապատասխան յուրաքանչյուր մուտքային հավաքածուի համար:

<u>Կոմբինացիոն սխեմայի իրագործումը</u>

ա) "**ԵՎ–ԿԱՄ**" երկկարգանի իրագործում: Եթե բուլյան ֆունկցիան առաջարկված է դիզյունկտիվ նորմալ ձևով, ապա նրա սխեմատիկ իրականացումը հետևյալն է. յուրաքանչյուր կոնյունկցիա իրագործվում է տրամաբանական "**ԵՎ**" տարրով, որի մուտքերին տրվում են ուղիղ կամ ժխտված մուտքային փոփոխականներ: "**ԵՎ**" տարրերի ելքային ազդանշանները տրամաբանորեն գումարվում են "**ԿԱՄ**" տարրի միջոցով:

բ) "**ԿԱՄ-ԵՎ**" երկկարգանի իրագործում: Եթե ֆունկցիան առաջարկված է կոնյուկտիվ նորմալ ձևով, ապա նրա սխեմատիկ իրականացումը հետևյալն է. յուրաքանչյուր դիզյունկցիա իրագործվում է տրամաբանական "**ԿԱՄ**" տարրով, իսկ դիզյունկցիաների տրամաբանական արտադրյալը՝ "**ԵՎ**" տարրով:

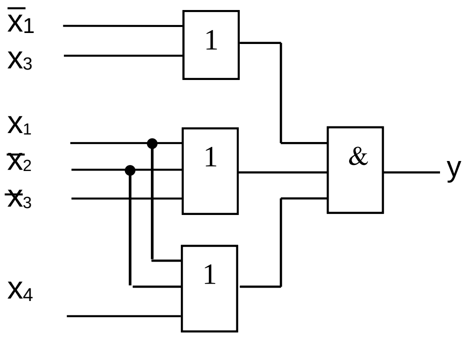
Օրինակ՝ $y = x_1x_3 \vee \bar{x}_1\bar{x}_2 \vee \bar{x}_1\bar{x}_3x_4$, որի իրագործումը "**ԵՎ**" և "**ԿԱՄ**" տարրերի օգնությամբ բերված է նկար 5-ում:



Նկար 5. **y** ֆունկցիայի իրագործումը “ԵՎ” և “ԿԱՄ” տարրերի օգնությամբ

“**ԿԱՄ-ԵՎ**” իրագործման համար դիտարկենք վերը նշված ֆունկցիան՝

$y = (\bar{x}_1 \vee x_3) \cdot (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \cdot (x_1 \vee \bar{x}_2 \vee x_4)$, որի սխեման բերված է նկար 6-ում:



Նկար 6. **y** ֆունկցիայի իրագործումը “ԿԱՄ” և “ԵՎ” տարրերի օգնությամբ

Սխեմաները նախագծվում են բուլյան ֆունկցիաների տրամաբանական (անալիտիկ) արտահայտությունների հիման վրա:

Բուլյան Ֆունկցիաների կառուցման հերթականությունն է.

1. Սխեմայի գործառության բառային նկարագրումից անցում դեպի բուլյան ֆունկցիայի տրման աղյուսակային ձևին:
2. Բուլյան ֆունկցիայի մինիմացում:
3. Անալիտիկ արտահայտությունից անցում դեպի սխեմա:

Բուլյան ֆունկցիաների մինիմացման խնդիրները լուծում են կլանման և սոսնձման օրենքների կիրառման միջոցով: Բուլյան ֆունկցիաների մինիմացումը կատարում են հետևյալ ձևով.

1. Բոլոր պարզ իմպլիկանտների որոշում,

2. Պարզ իմպլիկանտներով ֆունկցիայի բոլոր «1» արժեքների կետերի ծածկում (ընդգրկում):

F ֆունկցիայի P իմպլիկանտ է կոչվում այն կոնյունկցիան, որն ընդունում է «1» արժեք միայն ֆունկցիայի մեկական կետերում (թեկուզ մեկ կետում) և չի ընդունում «1» արժեք ֆունկցիայի զրոյական կետերում: P իմպլիկանտը կոչվում է պարզ, եթե նրանից չի կարելի հեռացնել ոչ մի տառ: Մինիմալ բուլյան ֆունկցիան հավասար է պարզ իմպլիկանտների դիզյունկցիային:

Դիզյունկտիվ նորմալ ձևերի դասում բուլյան ֆունկցիաների մինիմացումը

Բուլյան ֆունկցիան իրացնող սխեմայի բարդությունը որոշվում է նրա վերլուծական (անալիտիկ) բանաձևի բարդությամբ: Քանի որ միևնույն բուլյան ֆունկցիան կարելի է ներկայացնել տարբեր բանաձևերով, ապա առավել պարզ բանաձևի ընտրության խնդիրը անմիջականորեն բերում է առավել պարզ սխեմայի նախագծմանը:

Որոշ բազիսում բուլյան ֆունկցիայի նվազագույն ձևը կարելի է համարել այն, որը պարունակում է բազիսի ֆունկցիաների նվազագույն քանակով վերադրումներ՝ թողնելով նաև փակագծերը: Սակայն կառուցել նվազարկման արդյունավետ (էֆեկտիվ) ալգորիթմ, ստանալով նվազագույն փակագծային ձև, շատ դժվար է:

Դիտարկենք նվազարկման ավելի պարզ խնդիր, որի դեպքում որոնվում են բուլյան ֆունկցիայի նվազագույն **ԴՆՁ**-երը:

$\mathbf{f_{min}}$ ֆունկցիայի նվազագույն ձև է, որը **f** ֆունկցիայի պարզ իմպլիկանտների դիզյունկցիան է:

Այս թեորեմը հեշտությամբ ապացուցվում է հակադարձից:

Հասարակ իմպլիկանտների որոշումը Կառնոյի քարտերի միջոցով

Մինիմացումը Կառնոյի քարտերի օգնությամբ հարմար է, եթե փոփոխականների թիվը 6 - ից ավելի չէ:

Կառնոյի քարտը կառուցվում է այնպես, որ նրա հարևան վանդակները տարբերվում են միայն մեկ փոփոխականով:

3 փոփոխական ունեցող բուլյան ֆունկցիայի համար կազմված Կառնոյի քարտն ունի հետևյալ տեսքը՝

$x_1 \backslash x_2x_3$	00	01	11	10
0	000	001	011	010
1	100	101	111	110

Բուլյան ֆունկցիայի յուրաքանչյուր կետին համապատասխանում է մեկ վանդակ: Փոփոխականների հավաքածուները ցուցադրված են վանդակների ներսում: Ընդունենք, որ 3 փոփոխական ունեցող բուլյան ֆունկցիան ներկայացված է նկար 7-ում բերված աղյուսակի տեսքով:

Աղյուսակ 11

X_1	X_2	X_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

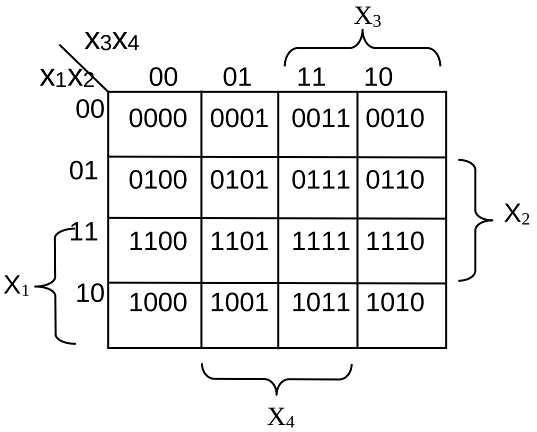
$x_1 \backslash x_2x_3$	00	01	11	10
0		1	1	
1	1	1	1	

$P_1 = x_1 \cdot \overline{x_2}$; $P_2 = x_3$;

$f_{min} = P_1 \vee P_2 = x_1\overline{x_2} \vee x_3$

Նկար 7. Երեք փոփոխական ունեցող բուլյան ֆունկցիայի նկարագրումը Կառնոյի քարտի միջոցով

Կառնոյի քարտը չորս փոփոխական ունեցող բուլյան ֆունկցիայի համար բերված է նկար 8-ում:



Նկար 8. Չորս փոփոխական ունեցող բուլյան ֆունկցիայի նկարագրումը Կառնոյի քարտի միջոցով

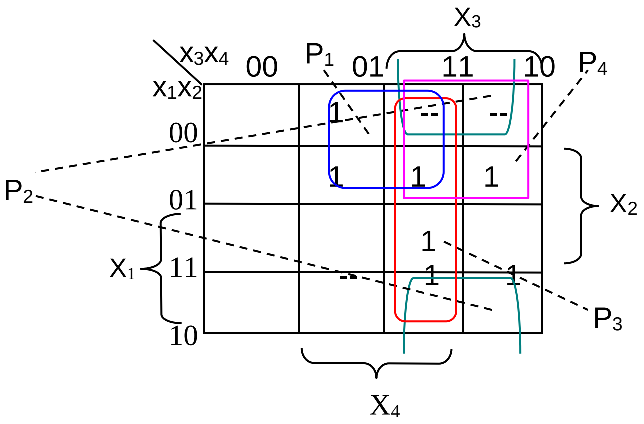
Դիտարկենք 4 փոփոխականից կախված բուլյան ֆունկցիայի $F(x_1, x_2, x_3, x_4)$ օրինակ, որը ներկայացված է աղյուսակ 12– ում:

Աղյուսակ 12

X_1	X_2	X_3	X_4	$F(X_1,X_2,X_3,X_4)$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	--
0	0	1	1	--
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	-
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

$F_{min} = P_1 \vee P_2 \vee P_3 \vee P_4 = \overline{X_1} X_4 \vee \overline{X_2} X_3 \vee X_3 X_4 \vee \overline{X_1} X_3$

Կարնոյի քարտերի միջոցով որոշում ենք պարզ իմպլիկանտները (նկար 9):



Նկար 9. Պարզ իմպլիկանտների որոշումը Կարնոյի քարտի միջոցով

Բուլյան ֆունկցիան ունի երեք անորոշ արժեքներ 0010, 0011 և 1001 կետերում:

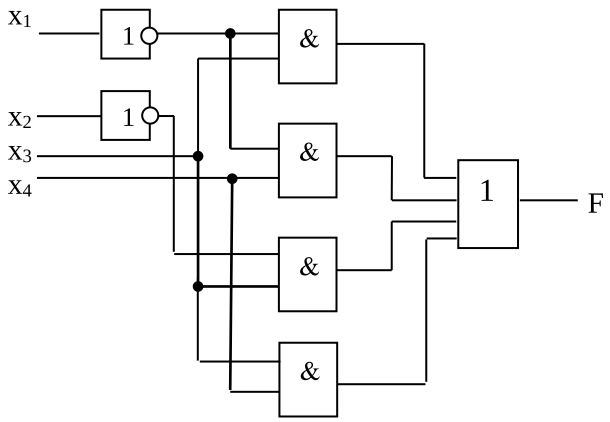
Երկու հարևան մեկերը խմբավորելով ստացվում է կոնյունկցիա n-1 փոփոխականից (տվյալ դեպքում n -1 = 3):

Չորս հարևան մեկերը խմբավորելիս ստանում ենք կոնյունկցիա n-2 փոփոխականից (n-2=2):

Ութ հարևան մեկերին համապատասխանում է n-3 փոփոխականի կոնյունկցիա (n-3=1):

Այն կետերը, ուր գրանցված են անորոշություններ, ընդունում են 1 կամ 0 արժեքներ, ըստ անհրաժեշտության: Մինիմալ արտահայտությունը ստանալու համար 0010 և 0011 կետերի անորոշությունները ընդունում ենք որպես 1 , իսկ 1001 կետում` որպես 0:

Նախագծենք սխեման “**ԵՎ**”, “**ԿԱՄ**”, “**ՈՉ**” տարրերի հիման վրա (նկար 10):



Նկար 10. Սխեմայի նախագծումը “ԵՎ”, “ԿԱՄ”, “ՈՉ” տարրերի օգնությամբ

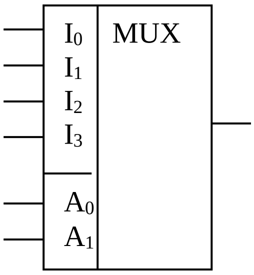
4.1.5. Մուլտիպլեքսորներ

Մուլտիպլեքսորն ունիվերսալ տրամաբանական տարր է:

Ենթադրենք՝ մուլտիպլեքսորն ունի **2^r** ինֆորմացիոն մուտքեր և r ղեկավարող մուտքեր: Կախված ղեկավարող մուտքերին տրված ազդանշանների արժեքների համակցումից՝ մուլտիպլեքսորում իրագործվում է ազդանշանների փոխանցումը ինֆորմացիոն մուտքերից մեկից նրա ելքին: **2^r** ինֆորմացիոն մուտքեր (i_0, i_1, + i_{2^r-1}) և **r** ղեկավարող (հասցեական) մուտքեր (a_0, a_1 . . . a_{r-1}) ունեցող մուլտիպլեքսորը սովորաբար նշանակում են MS (MUX կամ MX) (r), որի աշխատանքը նկարագրվում է հետևյալ բանաձևով՝

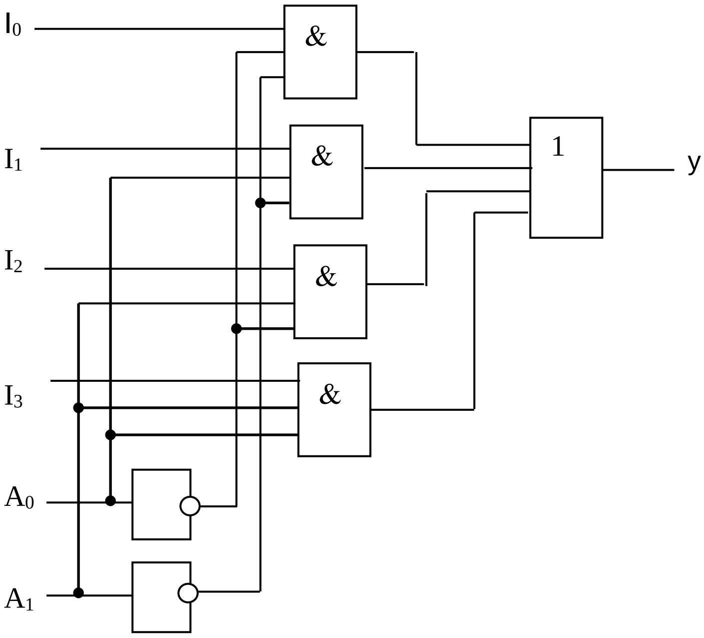
$$Y = i_0 \cdot \overline{a}_0 \cdot \overline{a}_1 \cdot \ldots \cdot \overline{a}_{r-1} \vee i_1 \cdot a_0 \cdot \overline{a}_1 \cdot \ldots \cdot \overline{a}_{r-1} \vee i_{2^r-1} \cdot a_0 \cdot a_1 \cdot \ldots \cdot a_{r-1}$$

Տեղադրելով i_0, i_1 . . .i_{2^r-1} փոխարեն «1» և «0» տարբեր համակցումները՝ կարելի է ստանալ r արգումենտներից կախված ցանկացած բուլյան ֆունկցիա:



Նկար 11. Մուլտիպլեքսորի սխեմայի պայմանական նկարագրումը

4 - 1 մուլտիպլեքսորի ֆունկցիոնալ սխեման բերված է նկար 12-ում:



Նկար 12. 4-1 մուլտիպլեքսորի սխեման

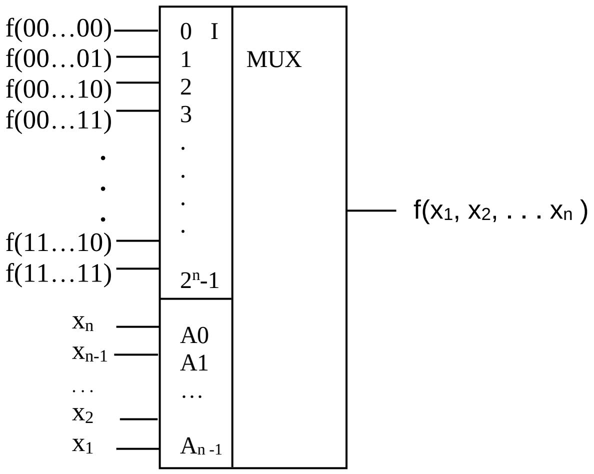
Տրամաբանական սխեմաների նախագծումը մուլտիպլեքսորի միջոցով

Մուլտիպլեքսորները հաճախ օգտագործում են որպես ունիվերսալ տրամաբանական տարրեր: Դա բացատրվում է նրանով, որ մուլտիպլեքսորի միջոցով իրականացվող ֆունկցիան կառուցվածքով համընկնում է ԿԴՆՁ-ի հետ, այդ պատճառով էլ մուլտիպլեքսորը առանց դժվարության կարելի է սարքաբերել ցանկացած տրամաբանական ֆունկցիայի:

1. Հաստատուն «0» և «1» սարքաբերում:

Ունենք $2^n - 1$ մուլտիպլեքսոր:

n փոփոխականից $y = f(x_1, x_2, \ldots x_n)$ ֆունկցիայի իրականացման համար պահանջվում է մուլտիպլեքսորի հասցեական մուտքերին տալ ֆունկցիայի $x_1, x_2, \ldots x_n$ փոփոխականները, իսկ $I0, I1, \ldots I2^n-1$ ինֆորմացիոն մուտքերին ՝ $0, 1 \ldots 2^n-1$ հավաքածուներին համապատասխանող ֆունկցիայի արժեքները (նկար 13):



Նկար 13. Ֆունկցիայի ($y = f(x_1, x_2, \ldots x_n)$) իրականացումը մուլտիպլեքսորի օգնությամբ

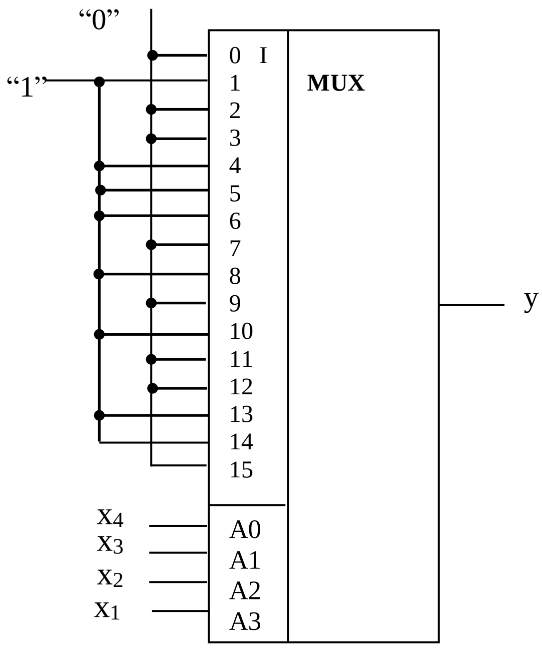
Փոփոխականների յուրաքանչյուր հավաքածուին համապատասխան ելք է փոխանցվում ֆունկցիայի արժեքը տվյալ հավաքածուի վրա: Տարբեր ֆունկցիաներին կհամապատասխանեն սարքաբերման տարբեր կոդեր: Սարքաբերման այբուբենը կլինի {«0», «1»}:

Օրինակ՝ տրված է աղյուսակի տեսքով ներկայացված 4 փոփոխականից ֆունկցիա (աղյուսակ 13)։

Աղյուսակ 13

x_1 x_2 x_3 x_4	y		x_1 x_2 x_3 x_4	y
0 0 0 0	0		1 0 0 0	1
0 0 0 1	1		1 0 0 1	0
0 0 1 0	0		1 0 1 0	1
0 0 1 1	0		1 0 1 1	0
0 1 0 0	1		1 1 0 0	0
0 1 0 1	1		1 1 0 1	1
0 1 1 0	1		1 1 1 0	1
0 1 1 1	0		1 1 1 1	0

Իրականացնենք աղյուսակ 13-ում տրված ֆունկցիան 16–1 մուլտիպլեքսորի օգնությամբ (նկար 14)։



Նկար 14. Աղյուսակ 12-ում նկարագրված ֆունկցիայի իրականացումը 16–1 մուլտիպլեքսորի միջոցով

Հնարավոր է պակասեցնել սարքաբերման մուտքերի քանակը՝ մեծացնելով մուլտիպլեքսորի սարքաբերման այբուբենը: Դիտարկենք դա օրինակով: Իրականացնենք աղյուսակ 14-ի տեսքով ներկայացված ֆունկցիան 8-1 մուլտիպլեքսորի հիման վրա:

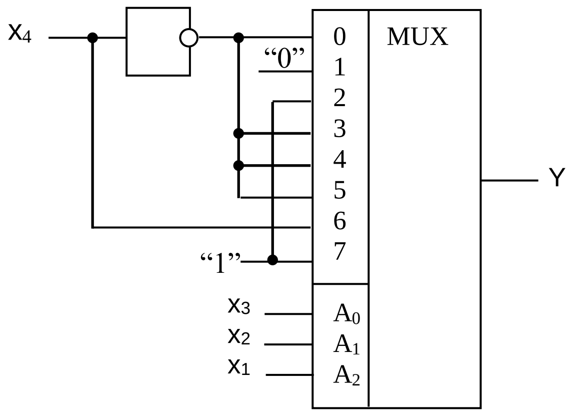
Աղյուսակ 14

x_1 x_2 x_3 x_4	y	$F_{մնաց}$
0 0 0 0	0	$I_0=x_4$
0 0 0 1	1	
0 0 1 0	0	$I_1=0$
0 0 1 1	0	
0 1 0 0	1	$I_2=1$
0 1 0 1	1	
0 1 1 0	1	$I_3=\overline{x_4}$
0 1 1 1	0	
1 0 0 0	1	$I_4=\overline{x_4}$
1 0 0 1	0	
1 0 1 0	1	$I_5=\overline{x_4}$
1 0 1 1	0	
1 1 0 0	0	$I_6=x_4$
1 1 0 1	1	
1 1 1 0	1	$I_7=1$
1 1 1 1	1	

Մուլտիպլեքսորի հասցեական մուտքերին տրվում են x_1, x_2, x_3 փոփոխականները: Փոփոխական x_4-ը ընդգրկում ենք սարքաբերման այբուբենի մեջ: Այս դեպքում սարքաբերման ֆունկցիան կլինի մեկ փոփոխականի ֆունկցիա: Սովորաբար սարքաբերման ֆունկցիան կոչում են մնացորդային ֆունկցիա: Մնացորդային ֆունկցիան տրվում է մուլտիպլեքսորի ինֆորմացիոն (սարքաբերման) մուտքերին:

Տվյալ ֆունկցիայի իրականացումը մուլտիպլեքսորի հիման վրա ներկայացված է նկար 15 -ում:

Եթե սարքաբերման ազդանշանների թվին միացվում է n փոփոխականով ֆունկցիայի մեկ փոփոխական, ապա ստացվում է n տարբերակ (տվյալ դեպքում՝ 4): Ինչպե՞ս որոշել, թե որ փոփոխականն է անհրաժեշտ ընտրել սարքաբերման համար: Խորհուրդ է տրվում վերցնել այն փոփոխականը, որը նվազագույն քանակով է առկա ֆունկցիայի նվազագույն տեսքում: Այս դեպքում ավելանում է հաստատուն զրոների և մեկերի թիվը:



Նկար 15. Աղյուսակ 14-ում նկարագրված ֆունկցիայի իրականացումը 8–1 մուլտիպլեքսորի միջոցով

Եթե սարքաբերման ազդանշանների թվին միացվեն 2 փոփոխական, ապա լրացուցիչ տրամաբանական սխեմաները իրենցից կներկայացնեն երկու մուտքանի «եվ» տրամաբանական տարրեր:

4.1.6. Վերծանիչներ

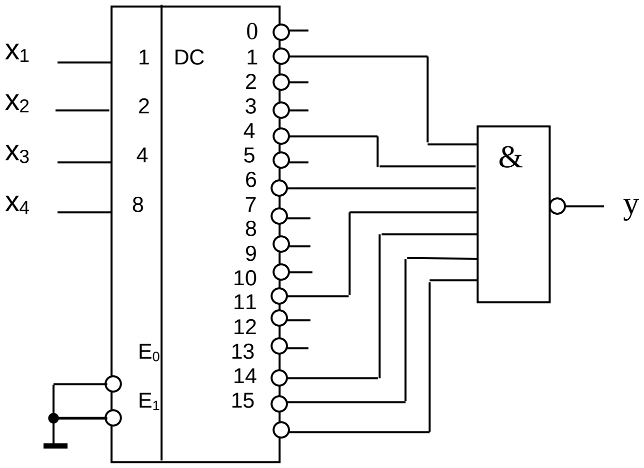
Ինֆորմացիայի մեծ մասը համակարգիչներում պահվում և մշակվում է կոդավորված տեսքով:

Վերծանիչը մի քանի մուտքեր և ելքեր ունեցող տրամաբանական սխեմա է, որը կոդավորված մուտքային ազդանշանները ձևափոխում է ելքային կոդավորված ազդանշանների: Մուտքային կոդը սովորաբար ունի ավելի փոքր կարգայնություն, քան ելքայինը:

Սխեման, որը **n**-կարգանի երկուական կոդը ձևափոխում է $\mathbf{2^n}$-կարգանի ($\mathbf{n \times 2^n}$) կոդի, կոչվում է լրիվ վերծանիչ:

Քանի որ $\mathbf{n - 2^n}$ վերծանիչը իրականացնում է **n** փոփոխականով ֆունկցիայի բոլոր մինիթերմերը, դրա հիման վրա կարելի է իրականացնել **n** փոփոխականով ցանկացած ֆունկցիա, որի համար պետք է “**ԿԱՄ**” տարրով միավորել վերծանիչի այն ելքերը, որոնք համապատասխանում են հավաքածուներին, որոնց վրա ֆունկցիայի արժեքը «1»: է

Իրականացնենք աղյուսակ 16 -ում բերված ֆունկցիան 4-16 ժխտած ելքերով վերծանիչի վրա (Լժ3 միկրոսխեմա):



Նկար 16. Աղյուսակ 14 -ում բերված ֆունկցիայի իրականացումը 4-16 ժխտած ելքերով վերծանիչի օգնությամբ

4.2. Ավտոմատի սխեմայի սինթեզում

Կոմբինացիոն սխեմաներից բացի, գոյություն ունեն ինֆորմացիայի ավելի բարդ ձևափոխիչներ, որոնց ռեակցիան կախված է տվյալ պահին ոչ միայն մուտքի վիճակից, այլև նրանից, թե ինչ կար մուտքին մինչ այդ: Այդպիսի ձևափոխիչները կոչվում են ավտոմատներ:

Ավտոմատ է կոչվում ինֆորմացիայի դիսկրետ ձևափոխիչը, որը մուտքային ազդանշանների ազդեցության տակ ընդունակ է անցնել մեկ վիճակից մյուսը և ձևավորել ելքային ազդանշաններ:

Վերացական ավտոմատն առաջադրվում է հետևյալ հինգ բազմությունների օգնությամբ՝ **A=** $\{X,Y,S,\delta,\lambda\}$, որտեղ $X=\{X_1,X_2, \dagger, X_M\}$ - ավտոմատի մուտքային այբուբենն է, $Y=\{Y_1,Y_2, \dagger,Y_N\}$ - ավտոմատի ելքային այբուբենն է, $S=\{S_0,S_1, \dagger, S_{k-1}\}$ - ավտոմատի ներքին վիճակների բազմությունն է կամ այբուբենը, δ - անցումների ֆունկցիան է, λ - ավտոմատի ելքերի ֆունկցիան է:

Ավտոմատը կոչվում է վերջավոր, եթե **X, Y, S** բազմությունները վերջավոր են:

Ավտոմատը գործառում է ժամանակի դիսկրետ պահերին t=0,1, 2,†n: Ժամանակի յուրաքանչյուր պահին ավտոմատը գտնվում է S բազմության որևէ մի վիճակում: S_0 -ն ավտոմատի սկզբնական վիճակն է (ժամանակի t=0 պահին):

δ անցումների ֆունկցիան ժամանակի յուրաքանչյուր t պահին որոշում է ավտոմատի հաջորդ վիճակը՝ կախված ավտոմատի ընթացիկ վիճակից և մուտքային ազդանշանից: Այլ կերպ ասած, δ ֆունկցիան «վիճակ - մուտքային ազդանշան» յուրաքանչյուր զույգին համապատասխանեցնում է հաջորդ վիճակը:

$\delta: \mathbf{S}\times\mathbf{X}\rightarrow\mathbf{S}$ δ-ն **S**×**X** դեկարտյան արտադրյալի ար-տապատկերումն է **S** բազմության մեջ: Անցումների ֆունկցիան կարելի է գրել հետևյալ տեսքով՝ $\mathbf{S_{t+1} = \delta (S_t, x_t)}$:

λ ելքային ֆունկցիան ժամանակի յուրաքանչյուր t պահին որոշում է ավտոմատի ելքային ազդանշանը:

Գոյություն ունի ավտոմատների 2 մոդել՝ Միլիի և Մուրի ավտոմատներ: Դրանք տարբերվում են ելքերի ֆունկցիաներով, որոնք որոշվում են հետևյալ կերպ՝

$\mathbf{Y_t = \lambda (S_t, X_t)}$ – Միլիի ավտոմատի համար կամ $\lambda: \mathbf{S}\times\mathbf{X}\rightarrow\mathbf{Y}$

$\mathbf{Y_t = \lambda (S_t)}$ – Մուրի ավտոմատի համար:

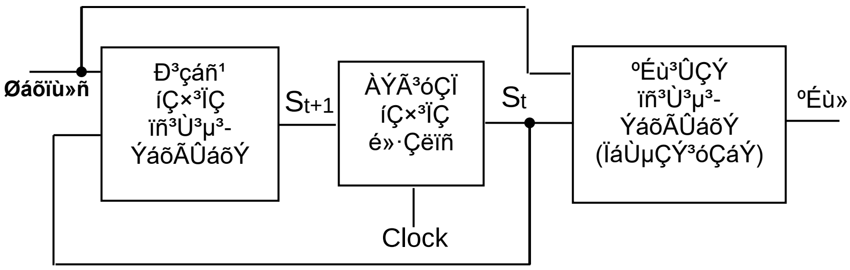
Միլիի ավտոմատում ելքային ազդանշանը ժամանակի t պահին կախված է ինչպես մուտքային ազդանշանից ժամանակի t պահին, այնպես էլ ընթացիկ վիճակից:

Մուրի ավտոմատում ելքային ազդանշանը բացահայտ կախված չէ մուտքային ազդանշանից, այլ որոշվում է միայն ընթացիկ վիճակով:

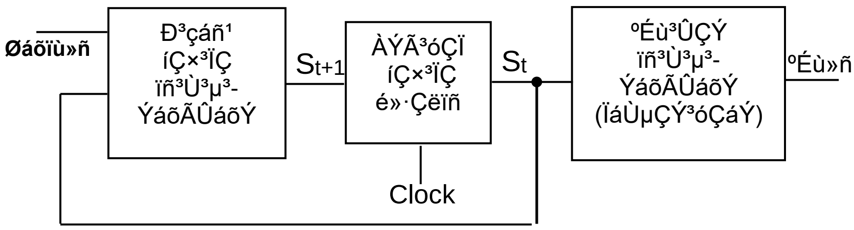
Վիճակը ավտոմատի հիշողությունն է անցյալ մուտքային ազդե-ցությունների մասին:

Ավտոմատները կոչում են նաև հաջորդականային մեքենաներ (Sequential Machines), քանի որ մուտքային հաջորդականությունը ձևափոխվում է վիճակների հաջորդականության և ելքային հաջորդականության: Անգլալեզու գրականությունում օգտագործվում է FSM (Finite State Machine) անվանումը:

Միլի և Մուրի ավտոմատների կառուցվածքային սխեմաները բերված են համապատասխանորեն նկար 17 և 18-ում:



Նկար 17. Միլի ավտոմատի կառուցվածքային սխեման



Նկար 18. Մուրի ավտոմատի կառուցվածքային սխեման

Սովորաբար վերջավոր ավտոմատը բաղկացած է 3 մասից.

1. Ընթացիկ վիճակի ռեգիստր: Որպես կանոն, դա տակտավորված D տրիգերների հավաքածու է, որոնք սինքրոնացվում են նույն սինքրոազդանշանով:
2. Անցումների տրամաբանություն: Յաջորդ վիճակը ընթացիկ վիճակից և մուտքային ազդանշանից կախված ֆունկցիա է:
3. Ելքի ձևավորման տրամաբանություն:

Անխափան աշխատանքի համար միշտ ապահովվում է ավտոմատի տեղակայումը սկզբնական վիճակում: Այսպիսով ապահովում է ավտոմատի ինիցալացումը նախօրոք որոշված վիճակ առաջին իսկ տակտային իմպուլսի պահին:

Այն դեպքում, երբ տեղակայումը սկզբնական վիճակ նախատեսված չէ, անհնար է կանխատեսել, թե որ սկզբնական վիճակից է սկսում ավտոմատի գործառնությունը: Սա շատ կարևոր է սնուցումը միացնելու դեպքում: Սովորաբար կիրառում են տեղակայման ասինքրոն սխեմաներ:

4.2.1. Ավտոմատների կառուցվածքային սինթեզի փուլերը

Ավտոմատի կառուցվածքային սինթեզի խնդիրը ավտոմատի սխեմայի կառուցումն է ըստ նրա վերացական առաջադրանքի:

Կառուցվածքային սինթեզի հաջորդականությունը հետևյալն է. ավտոմատի մուտքային և ելքային ազդանշանների կոդավորում, ներքին վիճակների կոդավորում, տրիգերների գրգռման ֆունկցիաների և ավտոմատի ելքերի ֆունկցիաների որոշում, ավտոմատի տրամաբանական սխեմայի կառուցում:

Ավտոմատի մուտքային և ելքային ազդանշանների կոդավորումը

Այս փուլում որոշվում է ավտոմատի մուտքերի և ելքերի քանակը: Ավտոմատի մուտքային ազդանշանների թիվը՝ m-ը որոշվում է հետևյալ հարաբերակցությամբ՝

$$\lceil \log_2 M \rceil \leq m \leq M,$$

որտեղ M –ը մուտքային այբուբենի տառերի քանակն է: Այնուհետև մուտքային այբուբենի յուրաքանչյուր տառին համապատասխանեցվում են զրոներից և մեկերից կազմված $x_1, x_2 \ . \ . \ . \ x_m$ ազդանշանների հավաքածուները:

Նույնապես ավտոմատի ելքային ազդանշանների քանակը՝ n-ը, որոշվում է հետևյալ հարաբերակցությամբ՝

$$\lceil \log_2 N \rceil \leq n \leq N,$$

որտեղ N – ը ելքային այբուբենի տառերի թիվն է: Ելքային այբուբենի ամեն տառին համապատասխանեցվում է $y_1, y_2 \ . \ . \ . \ y_n$ ելքային ազդանշանների կոդը: Մուտքային (ելքային) այբուբենի բոլոր տառերը կոդավորվում են տարբեր հավաքածուներով:

Ավտոմատի վիճակների կոդավորում

Ներքին վիճակների կոդավորման եղանակը էապես ազդում է ինչպես անցումների ֆունկցիաների, այնպես էլ ելքերի ֆունկցիաների բարդության վրա, իսկ արդյունքում՝ ավտոմատի կոմբինացիոն մասի բարդության վրա:

Այս փուլում ընտրվում է տրիգերի տիպը ավտոմատի հիշողությունը իրականացնելու համար: Տրիգերների քանակը որոշվում է հետևյալ կերպ՝

$$\lceil \log_2 K \rceil \leq k \leq K,$$

որտեղ K - ն ավտոմատի ներքին վիճակների քանակն է: Տրիգերների ելքերը նշենք $q_1, q_2 \ . \ . \ . \ q_k$: Ավտոմատի յուրաքանչյուր վիճակին համապատասխանեցվում է զրոներից և մեկերից k կարգանի հավաքածու այնպես, որ ոչ մի երկու վիճակներին չվերագրվի նույն հավաքածուն: Այս գործընթացը կոչվում է կոդավորում:

Տարբեր եղանակով կոդավորման արդյունքում ստացվում են ավտոմատների ելքերի և հաջորդ վիճակի տարբեր ֆունկցիաներ: Այն բանից, թե ինչպես են կոդավորված վիճակները, կախված է սխեմայի բարդությունը: Դժվար է գտնել կոդավորման օպտիմալ տարբերակը, որն ապահովում է ավտոմատի ամենապարզ սխեմայի իրագործումը: Դա բացատրվում է առաջին հերթին կոդավորման տարբերակների մեծ քանակով, նաև նրանով, որ հաջորդ վիճակի ֆունկցիան կախված է տրիգերի տիպից: Տրիգերներից նախապատվությունը տրվում է RS- կամ JK-տրիգերներին, քանի որ մուտքային ազդանշանների անորոշությունների առկայությունը թույլ է տալիս ստանալ ավելի պարզ սխեմաներ:

Կոդավորման ամենապարզ եղանակը` երկուական (բինար) կոդավորումն է, երբ տրիգերների քանակը վերցվում է նվազագույնը $\lceil \log_2 K \rceil$ կամ “1 K-ից” (ունար) կոդավորումը, երբ տրիգերների քանակը առավելագույն է` ամեն վիճակին համապատասխանեցվում է տրիգեր: Կոդավորման այս եղանակը անգլերեն կոչվում է One-Hot Encoding. Այս եղանակով կոդավորման ժամանակ տրիգերների մուտքային շղթաներում օգտագործվում են փոքր քանակով տրամաբանական սխեմաներ, իսկ տրիգերների հավաքածուն տեղաշարժող ռեգիստրի տիպի կառուցվածք է կազմում:

4.2.2. Գրգռման ֆունկցիաների և ավտոմատի ելքերի ֆունկցիաների որոշումը

Գրգռման ֆունկցիաների որոշման համար օգտագործվում է անցումների կոդավորված աղյուսակը: Յուրաքանչյուր տիպի տրիգերի համար գրգռման ֆունկցիան որոշվում է նրա գրգռման ֆունկցիայի մատրիցից: Կատարվում է ընթացիկ վիճակի կոդի կարգ առ կարգ համեմատումը հաջորդ վիճակի կոդի հետ: Յուրաքանչյուր անցումի համար որոշվում է ֆունկցիայի արժեքը: D տրիգերի գրգռման ֆունկցիան որոշելիս համեմատություն չի արվում, քանի որ ֆունկցիայի արժեքը համընկնում է հաջորդ վիճակի կոդի հետ:

Թվային ավտոմատի սխեմայի նախագծման համար հարմար է օգտվել հետևյալ հաջորդականությամբ.

1. Ելնելով ավտոմատի գործառման պայմաններից` որոշվում է նրա վիճակների քանակը:
2. Ավտոմատը ներկայացվում է իր ստանդարտ ձևերից որևէ մեկով (օրինակ, գրաֆի կամ աղյուսակի միջոցով):
3. Կազմվում է անցումների կոդավորված աղյուսակը:

4. Այդ աղյուսակով որոշվում են տրիգերների ելքերի և գրգռման ֆունկցիաները: Կատարվում է ֆունկցիաների համատեղ մինիմացումը:
5. Ստացված ֆունկցիաների հիման վրա կառուցվում է ավտոմատի տրամաբանական սխեման:

Դիտարկենք չօգտագործված վիճակների կիրառումը: Եթե ավտոմատի վիճակների քանակը k տրիգերների առկայության դեպքում ավելի քիչ է, քան 2^k, ապա կախված ներկայացված պահանջներից, կիրառվում են երկու մոտեցումներ`

- Նվազագույն ռիսկ,
- Նվազագույն արժեք:

Նվազագույն ռիսկ. ենթադրվում է, թե ավտոմատը ինչ-որ կերպ կարող է ընկնել չօգտագործվող վիճակի մեջ, օրինակ` անսարքությունների, անսպասելի մուտքային ազդեցությունների կամ նախագծման սխալների պատճառով: Բոլոր չօգտագործվող վիճակները գրանցվում են վիճակների աղյուսակում, և ակնհայտ նշվում են դրանցից անցումներն այնպես, որ ցանկացած մուտքային ազդեցության տակ ավտոմատը անցնի նախնական վիճակի, չզբաղվածության վիճակի կամ որևէ այլ անվտանգ վիճակի:

Նվազագույն արժեք. ենթադրվում է, թե ավտոմատը երբեք չի ընկնում չօգտագործվող վիճակի մեջ, որի պատճառով այդ վիճակները կարելի է համարել «անտարբեր»: Շատ դեպքերում դա պարզեցնում է գրգռման ֆունկցիաները: Սակայն եթե ավտոմատը, այնուամենայնիվ, ընկնում է չօգտագործվող վիճակի մեջ, նրա գործառումը կարող է տարօրինակ լինել:

RS - տրիգեր

RS- տրիգերը կոչվում է առանձացած տեղակայման մուտքերով տրիգեր: R (Reset) մուտքը - «0» տեղակայման մուտքն է , S (Set) մուտքը - «1» տեղակայման մուտքը:

RS- տրիգերի իսկության աղյուսակը լրիվ(աղյուսակ 15) և կրճատ (աղյուսակ 16) տեսքերով`

Աղյուսակ 15 Աղյուսակ 16

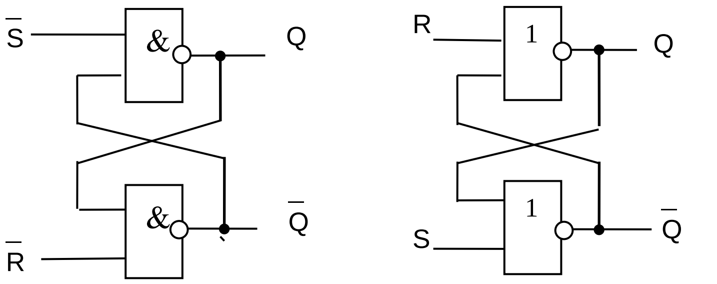
R	S	Q	Q*
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	–
1	1	1	–

R	S_t	Q*
0	0	Q
0	1	1
1	0	0
1	1	–

որտեղ՝ R, S, Q – մուտքերի և ելքի վիճակն է ժամանակի ընթացիկ t պահին: Q*- ելքի նոր վիճակն է (ժամանակի t+1 պահին):

RS- տրիգերի բնութագրային հավասարումը որոշվում է իսկության աղյուսակով՝ $Q^* = S \vee \overline{R}Q$:

“**ԵՎ-ՈՉ**” ու “**ԿԱՄ-ՈՉ**” տարրերի վրա կառուցված RS- տրիգերի սխեման ներկայացված է նկար 19-ում:



Նկար 19. RS-տրիգերի սխեման “ԵՎ-ՈՉ” ու “ԿԱՄ-ՈՉ” տարրերի վրա

RS - տրիգերի գրգռման ֆունկցիաների մատրիցը տրված է աղյուսակ 17-ում:

Ասինքրոն RS-տրիգերը հիմք է բոլոր տիպի տրիգերների կառուցման համար:

Աղյուսակ 17

Անցումներ	R	S

$0 \to 0$	-	0
$0 \to 1$	0	1
$1 \to 0$	1	0
$1 \to 1$	0	-

Սինքրոն RS-տրիգերի իսկության աղյուսակն ունի հետևյալ տեսքը (աղյուսակ 18)՝

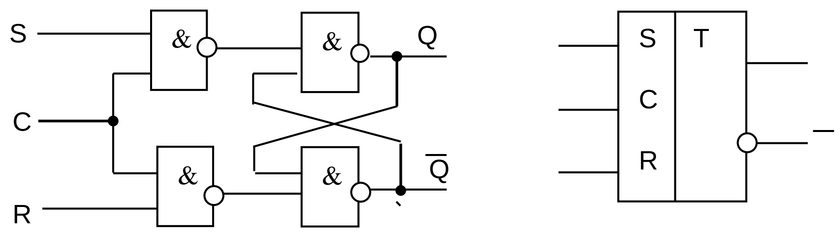
Աղյուսակ 18

R	S	C	Q	Q*
x	x	0	Q	Q
0	0	1	Q	Q*
0	1	1	X	1
1	0	0	X	0
1	1	1	X	-

Սինքրոն RS-տրիգերը կարող է ստացվել ասինքրոն RS-տրիգերից՝ ավելացնելով լրացուցիչ տրամաբանական սխեմա, որը տրիգերի մուտքերին կձևավորի ակտիվ տրամաբանական մակարդակ միայն սինքրոազդանշանի առկայության դեպքում:

Տրիգերը կառավարվում է սինքրոազդանշանի մակարդակով՝ C=1 դեպքում մուտքային ազդանշանի փոփոխումը կարող է բերել ելքային ազդանշանի փոփոխման:

Սինքրոն RS-տրիգերի պայմանական նշանակումը և սխեման ներկայացված են նկար 20-ում:



Նկար 20. Սինքրոն RS-տրիգերի սխեման և պայմանական նշանակումը

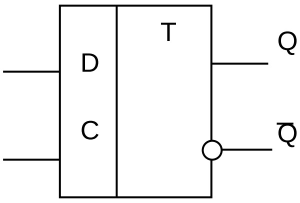
D - տրիգեր

D-տիպի (Delay բառից) տրիգերն ունի մեկ ինֆորմացիոն մուտք: Նրա վիճակը կրկնում է մուտքային ազդանշանը, բայց սինքրոազդանշանով որոշվող հապաղումով: Այստեղից հետևում է, որ D-տրիգերը հիմնականում սինքրոն է լինում:
D-տրիգերի իսկության աղյուսակը ունի հետևյալ տեսքը (աղյուսակ 19)՝

Աղյուսակ 19

D	Q	Q*
0	0	0
0	1	0
1	0	1
1	1	1

Տրիգերի բնութագրային հավասարումը՝ Q* =D:
D-տրիգերի պայմանական նշանակումը ներկայացված է նկար 21-ում:



Նկար 21. D-տրիգերի պայմանական նշանակումը

D-տրիգերի գրգռման ֆունկցիաների մատրիցը տրված է աղյուսակ 20-ում:

Աղյուսակ 20

Անցումներ	D
$0 \rightarrow 0$	0
$0 \rightarrow 1$	1
$1 \rightarrow 0$	0
$1 \rightarrow 1$	1

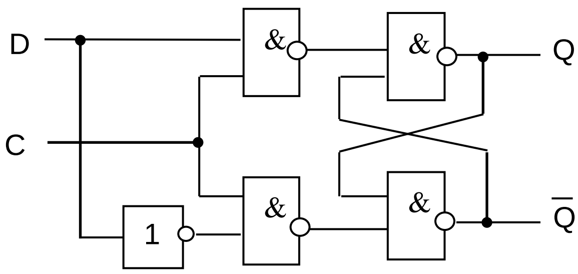
Սինքրոն D-տրիգերի իսկության աղյուսակը հետևյալն է (աղյուսակ 21)՝

Աղյուսակ 21

D	C	Q	Q*

x	0	0	0
x	0	1	1
0	1	x	0
1	1	x	1

Սինքրոն RS-տրիգերի վրա կառուցված D-տրիգերը ներկայացված է նկար 22-ում:



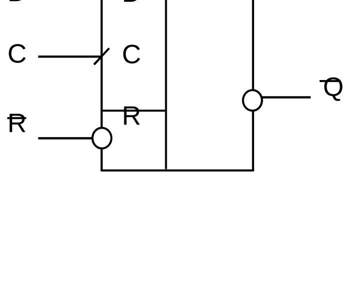
Նկար 22. D տրիգերի ներկայացումը սինքրոն RS-տրիգերի օգնությամբ

Սինքրոազդանշանի մակարդակով կառավարվող միաստիճան D-տրիգերը հաճախ անվանում են Latch-տիպի տրիգեր:

Քանի դեռ C=1, ինֆորմացիոն մուտքի ցանկացած փոփոխություն իսկույն հայտնվում է ելքերին: D-տրիգերի օգտագործման սովորական ռեժիմում նրանում ֆիքսվում է D ազդանշանի արժեքը սինքրոազդանշանի «1»-ից «0»-ի անցման պահին:

Ավտոմատի կառուցման ժամանակ որպես հիշող տարր հարմար է օգտագործել ճակատով ղեկավարվող D-տրիգերներ (դինամիկ կառավարումով տրիգերներ):

Դիտարկենք սինքրոազդանշանի առջևի ճակատով ղեկավարվող D-տրիգերի աշխատանքը: D-տրիգերի պայմանական գրաֆիկական նշանակումը (TM2 միկրոսխեմա) բերված է նկար 23-ում:



Նկար 23. D- տրիգերի պայմանական գրաֆիկական նշանակումը

Եթե $\overline{R}$ և $\overline{S}$ մուտքերին ունենք «1», ապա տրիգերն աշխատում է D և C մուտքերով: C=0 դեպքում տրիգերն աշխատում է հիշման ռեժիմում: Սինքրոազդանշանի 0-ից 1 փոխվելու ժամանակ տրիգերի ելքին կունենանք D արժեքին համապատասխանող ազդանշաններ՝ երբ D =1, Q = 1, երբ D = 0, Q = 0:

C=1 սինքրոազդանշանը տալուց հետո D մուտքի ազդանշանի փոփոխումը չի ազդում ելքի վրա: Տրիգերի հաջորդ փոխանջատման համար նախապես անհրաժեշտ է տալ C=0: Յաջորդ սինքրոիմպուլսի ճակատով տրիգերը կանցնի նոր վիճակի, որը կորոշվի D ինֆորմացիոն մուտքին տրված ազդանշանի արժեքով: Առջևի ճակատով ղեկավարվող D-տրիգերի աշխատանքի նկարագրումը բերված է աղյուսակ 22-ում:

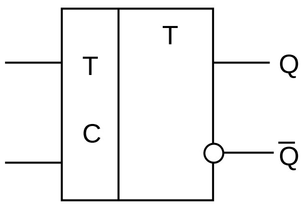
Աղյուսակ 22

$\overline{R}$	$\overline{S}$	D	C	Q*	è»ÅÇÙÝ»ñ
0	1	x	x	0	“0”-Ç ï»Õ³Ï³ÛáõÙ
1	0	x	x	1	“1”- Ç ï»Õ³Ï³ÛáõÙ
0	0	x	x	–	³ñ·»Éí³Í íÇ×³Ï
1	1	0	↑	0	“0” ·ñ³ÝóáõÙ
1	1	1	↑	1	“1” ·ñ³ÝóáõÙ

T- տրիգեր

T- տրիգերը կամ հաշվային տրիգերը (toggle անգլերեն բառից) ունի, ինչպես և D- տրիգերը, մեկ ինֆորմացիոն մուտք:

T- տրիգերի պայմանական նշանակումը ներկայացված է նկար 24-ում:



Նկար 24. T- տրիգերի պայմանական նշանակումը

T- տրիգերի գործառման աղյուսակը (աղյուսակ 23)՝

Աղյուսակ 23

T	Q	Q*
0	0	0
0	1	1
1	0	1
1	1	0

T- տրիգերի բնութագրիչ հավասարումը՝ $Q^* = T \oplus Q$:

T- տրիգերի գրգռման ֆունկցիաների մատրիցը

	T
$0 \to 0$	0
$0 \to 1$	1
$1 \to 0$	1
$1 \to 1$	0

Սովորաբար T- տրիգերը կառուցվում է JK- կամ D-տրիգերի հիման վրա:

JK-տրիգեր

JK-տրիգերը ունիվերսալ է, քանի որ դրա հիման վրա կարելի է կառուցել բոլոր դիտարկված տրիգերները:

JK-տրիգերի իսկության աղյուսակը (աղյուսակ 24)՝

Աղյուսակ 24

J	K	Q	Q*
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

JK-տրիգերի գործառման կրճատ աղյուսակը (աղյուսակ 25)՝

Աղյուսակ 25

J	K	Q*
0	0	Q
0	1	0
1	0	1
1	1	$\overline{Q}$

JK-տրիգերի գրգռման ֆունկցիաների մատրիցն ունի հետևյալ տեսքը՝

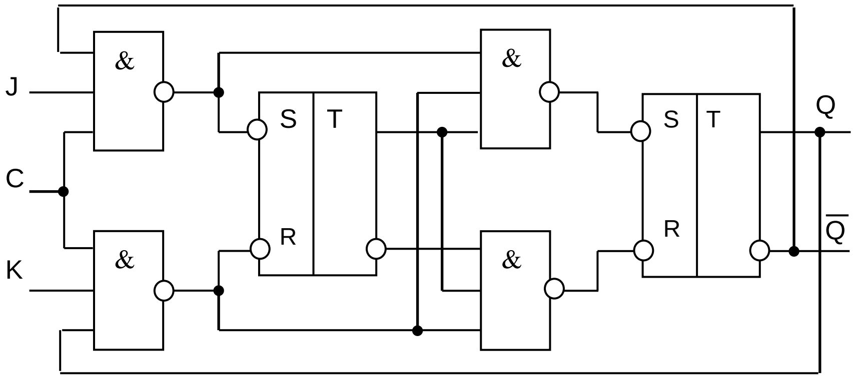
Աղյուսակ 26

Անցում	J	K
$0 \to 0$	0	–
$0 \to 1$	1	–
$1 \to 0$	–	1
$1 \to 1$	–	0

Ըստ աղյուսակ 24-ի կազմելով Կառնոյի քարտը՝ կարելի է ստանալ JK-տրիգերի բնութագրային հավասարումը՝

$Q^* = J\,\overline{Q} \vee \overline{K}Q$:

Արգելող կապերով երկաստիճան JK-տրիգերի սխեման բերված է նկար 25-ում: Այն կառուցվում է T-տրիգերի սխեմային համանմանորեն:



Նկար 25. Երկաստիճան JK- տրիգերի սխեման

Երբ C=1, ըստ տրիգերի վիճակի և J և K մուտքերին տրված արժեքների, տեղակայվում է մուտքային տրիգերը (Master): C=0 դեպքում ինֆորմացիան արտագրվում է ելքային (Slave) տրիգեր: Երբ C=1, առաջին խմբի ԵՎ-ՈՉ տարրերի ելքերին կլինեն հակառակ արժեքներ ունեցող ազդանշաններ (բացառությամբ J=K=0 դեպքից), որոնք արգելում են ինֆորմացիայի փոխանցումը երկրորդ տրիգեր: C=0 դեպքում առաջին խմբի տարրերի ելքերին կունենանք «1», հետևաբար ինֆորմացիայի փոխանցումը «M» տրիգերի ելքերից «S» տրիգեր կլինի թույլատրված: Երբ J=K=0 տրիգերի վիճակը չի փոփոխվում: Մուտքային J և K ազդանշանները պետք է մնան անփոփոխ այնքան ժամանակ, քանի C=1: Եթե այդ պայմանը չի կատարվում, առկա է այնպիսի իրավիճակ, որը կոչվում է մեկերի (կամ զրոների) զավթում:

Ներկայումս JK-տրիգերների համար առավելապես կիրառվում են ներքին հապաղումով և դինամիկ կառավարումով միաստիճան սխեմաները:

Ճակատով ղեկավարվող JK -տրիգեր

Մեկերի և զրոների զավթման հիմնախնդիրը լուծված է սինքրոազդանշանի ճակատով փոխանջատվող JK-տրիգերում: Ճակատով ղեկավարվող JK-տրիգերը կառուցվում է ճակատային D- տրիգերի հիման վրա:

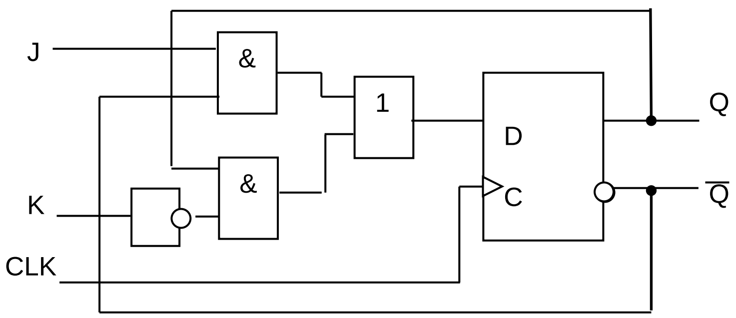
JK-տրիգերը կառուցելու համար օգտվենք նրա իսկության աղյուսակից: Յարկավոր է որոշել, թե ինչ ֆունկցիա պետք է տրվի D-տրիգերի մուտքին, որպեսզի ապահովվի նրա ճիշտ գործառումը:

Աղյուսակ 27

J_1	K_t	Q_t	Q_{t+1}	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

$D_t = Q_{t+1} = J\,\bar{Q}_t \vee \bar{K}Q_t$

Տրիգերի սխեման բերված է նկար 26-ում:



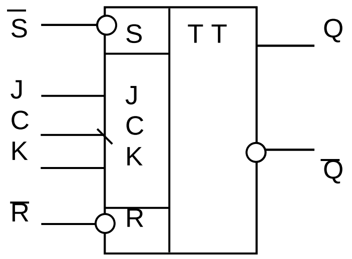
Նկար 26. D-տրիգերի հիման վրա JK տրիգերի սխեման

JK- տրիգերի իսկության աղյուսակը բերված է աղյուսակ 28-ում:
Q_N –ը տրիգերի ինվերսային ելքն է:

Աղյուսակ 28

J	K	CLK	Q*	Q_N *
x	x	0	Last Q	Last Q_N
x	x	1	Last Q	Last Q_N
0	0	↑	Last Q	Last Q_N
0	1	↑	0	1
1	0	↑	1	0
1	1	↑	Last Q_N	Last Q

Սինքրոազդանշանի ետին ճակատով ղեկավարվող JK-տրիգերի պայ-մանական գրաֆիկական նշանակումը (TB9 միկրոսխեմ) բերված է նկար 27-ում:



Նկար 27. JK-տրիգերի պայմանական գրաֆիկական նշանակումը

Ետին ճակատով ղեկավարվող JK- տրիգերի աշխատանքի նկարագրումը բերված է աղյուսակ 29-ում:

Աղյուսակ 29

$\overline{R}$	$\overline{S}$	J	K	C	Q*
0	1	x	x	x	0
1	0	x	x	x	1
0	0	x	x	x	–
1	1	0	0	↓	Last Q
1	1	0	1	↓	0
1	1	1	0	↓	1
1	1	1	1	↓	Last Q_N
1	1	x	x	0	Last Q
1	1	x	x	1	Last Q

5. Կոմբինացիոն սխեմայի սինթեզի օրինակ

Առաջադրանք: Կառուցել 8421 կշիռներով երկուական տասական կոդից 2421 կշիռներով երկուական տասական կոդի ձևափոխիչի սխեման:
Ձևափոխիչը կոմբինացիոն սխեմա է, որն ունի 4 մուտք և 4 ելք, ու իրագործում է 4 ֆունկցիա պարունակող համակարգ:

1. Մուտքային և ելքային կոդերի որոշումը:

Մուտքային և ելքային կոդերը որոշվում են հետևյալ կերպ`

$\mathbf{X = p_1 \cdot x_1 + p_2 \cdot x_2 + p_3 \cdot x_3 + p_4 \cdot x_4,,}$

$\mathbf{Y = p'_1 \cdot y_1 + p'_2 \cdot y_2 + p'_3 \cdot y_3 + p'_4 \cdot y_4,}$ որտեղ **X** և **Y** մուտքային և ելքային կոդերի նիշերի արժեքներն են, p_1,p_2,p_3,p_4 –ը մուտքային կոդի, իսկ p'_1, p'_2, p'_3, p'_4 –ը ելքային կոդին կշիռներն են:

Մուտքային և ելքային կոդերը բերված են աղյուսակ 30-ում:

Աղյուսակ 30

Նիշը	Մուտքային կոդ	Ելքային կոդ
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0
3	0 0 1 1	0 0 1 1
4	0 1 0 0	0 1 0 0
5	0 1 0 1	1 0 1 1
6	0 1 1 0	1 1 0 0
7	0 1 1 1	1 1 0 1
8	1 0 0 0	1 1 1 0
9	1 0 0 1	1 1 1 1

Իրագործվող ֆունկցիաների կոմբինացիաները բերված են աղյուսակ 31-ում:

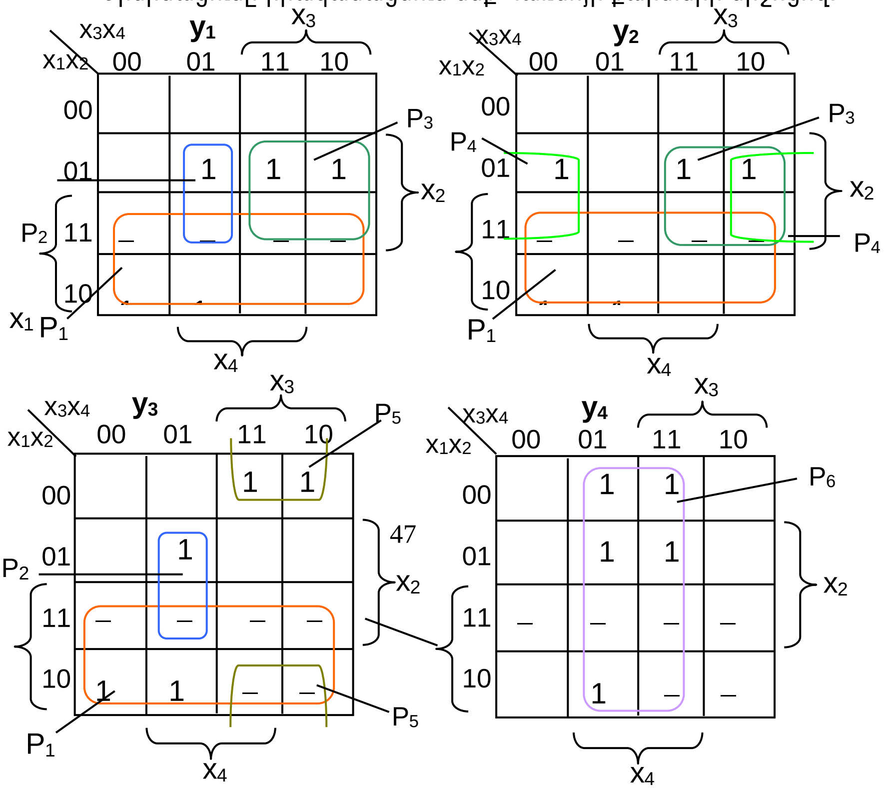
Սկսած 1010 հավաքածուից` y_1,y_2,y_3,y_4 ֆունկցիաները որոշված չեն:

Աղյուսակ 31

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4	x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4
0	0	0	0	0	0	0	0	1	0	0	0	1	1	1	0
0	0	0	1	0	0	0	1	1	0	0	1	1	1	1	1
0	0	1	0	0	0	1	0	1	0	1	0	–	–	–	–
0	0	1	1	0	0	1	1	1	0	1	1	–	–	–	–
0	1	0	0	0	1	0	0	1	1	0	0	–	–	–	–
0	1	0	1	1	0	1	1	1	1	0	1	–	–	–	–
0	1	1	0	1	1	0	0	1	1	1	0	–	–	–	–
0	1	1	1	1	1	0	1	1	1	1	1	–	–	–	–

Բուլյան ֆունկցիաների մինիմացումը

Մինիմացումը իրականացնում ենք Կառնոյի քարտերի միջոցով:



$y_1 = P_1 \vee P_2 \vee P_3 = x_1 \vee x_2 \cdot \overline{x_3} \cdot x_4 \vee x_2 \cdot x_3;$

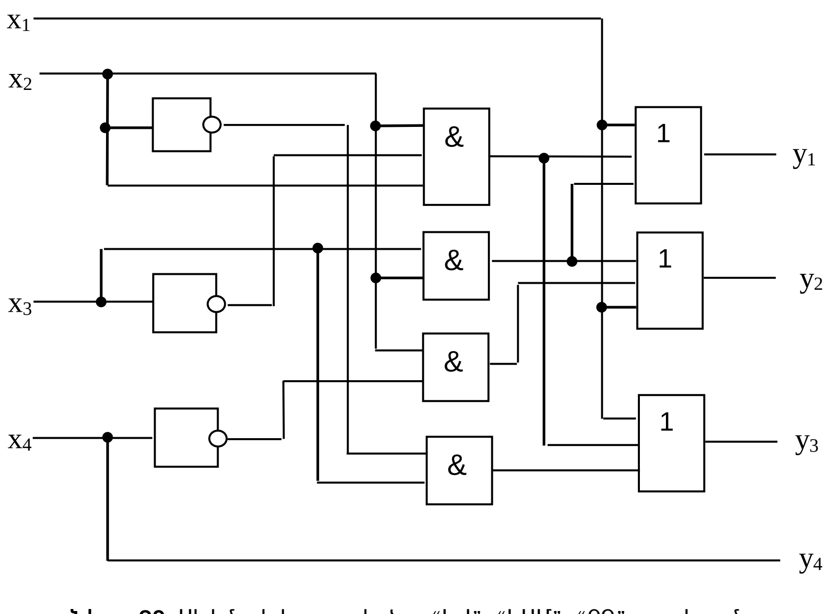
$y_2 = P_1 \vee P_3 \vee P_4 = x_1 \vee x_2 \cdot x_3 \vee x_2 \cdot \overline{x_4};$

$y_3 = P_1 \vee P_2 \vee P_5 = x_1 \vee x_2 \cdot \overline{x_3} \cdot x_4 \vee \overline{x_2} \cdot x_3;$

$y_4 = P_6 = x_4.$

Կատարվում է ֆունկցիաների համատեղ մինիմացում: P_2 իմպլիկանտը համարվում է ընդհանուր y_1 և y_3 ֆունկցիաների համար, իսկ P_3 իմպլիկանտը՝ y_1 և y_2 ֆունկցիաների համար:

Սխեմայի կառուցումը “ԵՎ”, “ԿԱՄ”, “ՈՉ” բազիսում



Նկար 28. Սխեմայի կառուցվածքը “ԵՎ”, “ԿԱՄ”, “ՈՉ” բազիսում

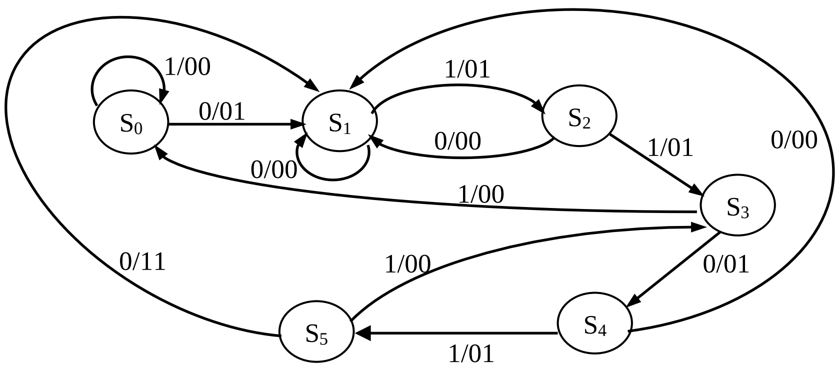
6. Ավտոմատի սինթեզի օրինակ

Կառուցել կոդային փականը ղեկավարող՝ մեկ X մուտք և երկու Y_1, Y_2 ելքեր ունեցող տակտավորվող սինքրոն ավտոմատի սխեման:

Ելքային Y_1 ազդանշանը հավասար է մեկի այն դեպքում, երբ X=0, իսկ նախորդ հինգ տակտերի ընթացքում X մուտքին տրվել է 01101 հաջորդականությունը:

Ելքային Y_2 ազդանշանը պետք է հավասար լինի մեկի միայն այն դեպքում, երբ X-ի ընթացիկ արժեքը ճիշտ է ավտոմատի՝ դեպի փականը բաց լինելու վիճակին շարժման (առաջընթացի) տեսանկյունից:

Կառուցենք կոդային փականի գրաֆը (նկար 29):



Նկար 29. Կոդային փականը ղեկավարող ավտոմատի գրաֆ

S_0 – հաշիվը բաց չէ;
S_1 – ընդունված է 0;
S_2 – ընդունված է 01;
S_3 – ընդունված է 011;
S_4 – ընդունված է 0110;
S_5 – ընդունված է 01101:

Սկզբնական S_0 վիճակում պահանջվող մուտքային հաջորդականության ոչ մի ազդանշան ընդունված չէ: Ավտոմատը սպասում է հաջորդականության առաջին գրոյի գալուն և, երբ այդ զրոն ընդունվում է, անցնում է S_1 վիճակի: Գտնվելով S_1 վիճակում՝ ավտոմատը սպասում է մեկի հայտնվելուն: Այն հայտնվելուն պես ավտոմատն անցնում է S_2 վիճակի: S_2 վիճակից ավտոմատն անցնում է S_3 վիճակի, եթե ընդունվում է «1» ազդանշանը, և վերադառնում է S_1 վիճակի, եթե մուտքին տրվում է «0»: S_3 վիճակից ավտոմատն անցնում է S_0 վիճակի, եթե մուտքին գալիս է 1: Եթե ավտոմատի մուտքին տրվում է «0», ապա այն անցնում է S_4 վիճակի: Եթե ավտոմատը գտնվում է S_4 վիճակում, և նրա մուտքին տրվում է սխալ ազդանշան (այսինքն «0»), ապա ավտոմատն անցնում է S_1 վիճակի: Ճիշտ արժեքի ընդունման դեպքում (1), ավտոմատը կանցնի S_5 վիճակի: Այս վիճակից ավտոմատը «0»-ի ազդեցության տակ կանցնի S_1 վիճակի, իսկ «1»-ի ազդեցության տակ կանցնի S_3 վիճակի, քանի որ նախորդ երեք մուտքային ազդանշանները կարող են հանդիսանալ պահանջվող մուտքային հաջորդականության 011 առաջին երեք բիթերը:

Ավտոմատի կառուցվածքային սինթեզ

Որոշենք ավտոմատի սխեմայի կառուցման համար պահանջվող տրիգերների **k** քանակը`

$\lceil \log_2 6 \rceil \leq k \leq 6$:

k-ն ընդունենք հավասար 3:

Վիճակները կոդավորենք հետևյալ կերպ`

S_0 – 000; S_1 – 001; S_2 – 010; S_3 – 011; S_4 – 100; S_5 – 110:

Կազմենք անցումների կոդավորման աղյուսակը:

Աղյուսակ 32

Ընթացիկ վիճակ	$q_1\ q_2\ q_3$	Մուտքային ազդ. X	Հաջորդ վիճակ	$q_1\ q_2\ q_3$	Ելքային ազդանշաններ $y_1\ y_2$	Գրգռման ֆունկցիաներ $D_1\ D_2\ D_3$
S_0	0 0 0	0	S_1	0 0 1	0 1	0 0 1
		1	S_0	0 0 0	0 0	0 0 0
S_1	0 0 1	0	S_1	0 0 1	0 0	0 0 1
		1	S_2	0 1 0	0 1	0 1 0
S_2	0 1 0	0	S_1	0 0 1	0 0	0 0 1
		1	S_3	0 1 1	0 1	0 1 1
		0	S_4	1 0 0	0 1	1 0 0

S_3	0 1 1	1	S_0	0 0 0	0 0	0 0 0
S_4	1 0 0	0	S_1	0 0 1	0 0	0 0 1
		1	S_5	1 1 0	0 1	1 1 0
S_5	1 1 0	0	S_1	0 0 1	1 1	0 0 1
		1	S_3	0 1 1	0 0	0 1 1

Սարքի կողմից իրագործվող ֆունկցիաների Կառնոյի քարտերը բերված են ստորև՝

D_1

q_1q_2 \ q_3x	00	01	11	10
00				
01				1
11			--	--
10		1	--	--

$$D_1 = q_2 \cdot q_3 \cdot \overline{x} \vee q_1 \cdot \overline{q_2} \cdot x;$$

D_2

q_1q_2 \ q_3x	00	01	11	10
00			1	
01		1		
11		1	--	--
10		1	--	--

$$D_2 = q_1 \cdot x \vee q_2 \cdot \overline{q_3} \cdot x \vee \overline{q_2} \cdot q_3 \cdot x;$$

D_3

q_1q_2 \ q_3x	00	01	11	10
00	1			1
01	1	1		
11	1	1	--	--
10	1		--	--

$$D_3 = \overline{q_2}\cdot\overline{x} \vee q_2\cdot\overline{q_3};$$

y_1

q_1q_2 \ q_3x	00	01	11	10
00				
01				
11	1		--	--
10			--	--

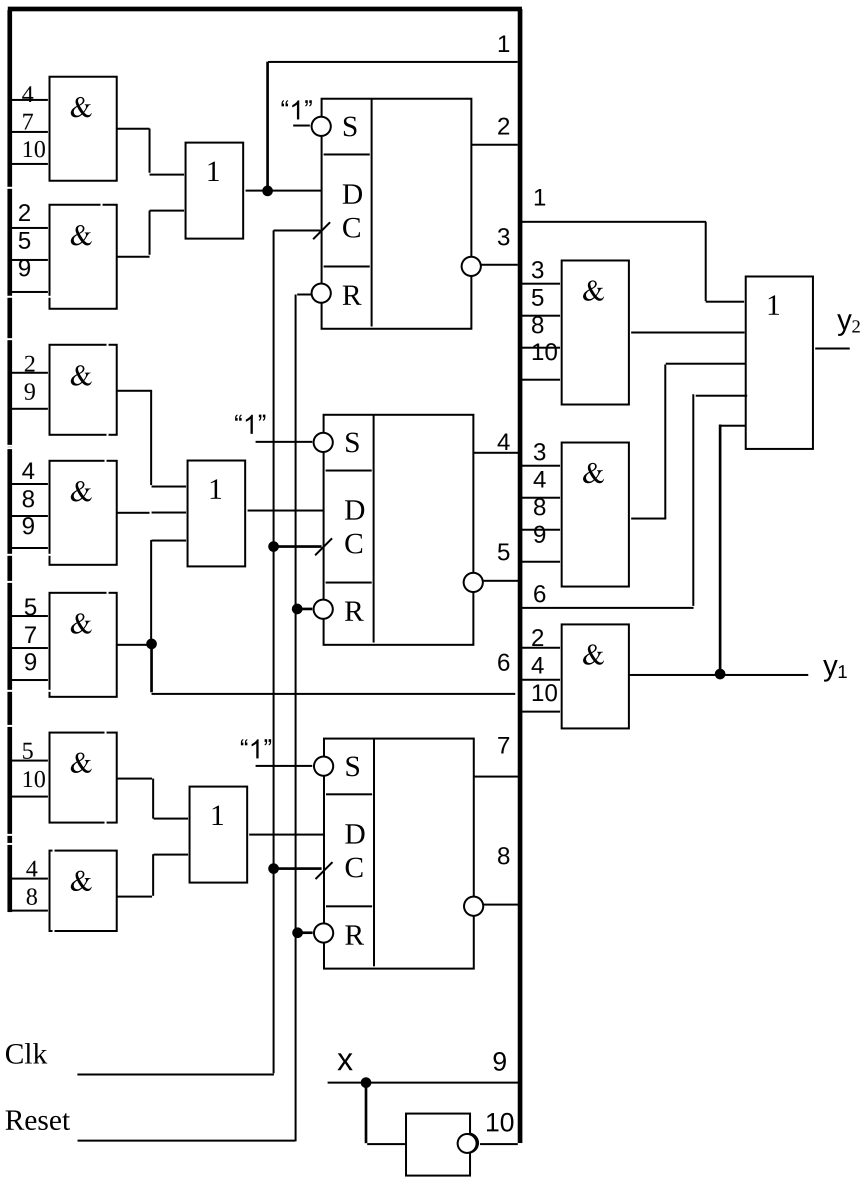
y_2

q_1q_2 \ q_3x	00	01	11	10
00	1		1	
01		1		1
11	1		--	--
10		1	--	--

$$y_1 = q_1\cdot q_2\cdot\overline{x};$$

$$y_2 = \overline{q_2}\cdot q_3\cdot x \vee q_1\cdot q_2\cdot\overline{x} \vee q_1\cdot\overline{q_2}\cdot x \vee q_2\cdot q_3\cdot\overline{x} \vee \overline{q_1}\cdot\overline{q_2}\cdot\overline{q_3}\cdot\overline{x} \vee \overline{q_1}\cdot q_2\cdot\overline{q_3}\cdot x =$$
$$= D_1 \vee \overline{q_2}\cdot q_3\cdot x \vee y_1 \vee \overline{q_1}\cdot\overline{q_2}\cdot\overline{q_3}\cdot\overline{x} \vee \overline{q_1}\cdot q_2\cdot\overline{q_3}\cdot x ;$$

Ավտոմատի սխեման ներկայացված է նկար 30-ում: Ավտոմատը կառուցված է սինքրոազդանշանի առջևի ճակատով ղեկավարվող D տրիգերների վրա:



Նկար 30. Ավտոմատի սխեման D- տրիգերների վրա

7. Կուրսային աշխատանքի կատարման հաջորդականությունը և ձևավորումը

7.1. Կոմբինացիոն սխեմայի նախագծման հաջորդականությունը և բացատրագրի ձևավորումը

Կուրսային աշխատանքում կոմբինացիոն սխեմայի նախագծման հաջորդականությունը և բացատագրի բովանդակությունը հարկավոր է իրականացնել ըստ հետևյալ քայլերի.

- Առաջադրանքի բովանդակությունը:
- Մուտքային և ելքային կոդերի աղյուսակի կազմումը:
- Իրագործվող ֆունկցիաների աղյուսակի կազմումը:
- Ֆունկցիաների մինիմացումը Կառնոյի քարտերի միջոցով:
- Անցում **"ԵՎ-ՈՉ"** և **"ԿԱՄ-ՈՉ"** բազիսների:
- Սարքի սխեմաների նախագծումը հետևյալ բազիսներում.
    - "**ԵՎ**", "**ԿԱՄ**", "**ՈՉ**",
    - "**ԵՎ-ՈՉ**",
    - "**ԿԱՄ-ՈՉ**":
- Իրագործվող ֆունկացիաների ժեգալկինի բազմանդամների որոշումը:
- SSL սերիայի միկրոսխեմաների վրա սարքի սկզբունքային էլեկտրական սխեմայի նախագծումը: Ներկայացնել միկրոսխեմաների որակավորման աղուսակը:
- Սարքի սխեմայի նախագծումը վերծանիչի միջոցով:
- Սարքի սխեմայի նախագծումը մուլտիպլեքսորների միջոցով:
  Այս դեպքում հարկավոր է ղեկավարվել ֆունկցիաների նվազագույն արժեքներով: Եթե ֆունկցիայի փոփոխականների քանակը **k** է, ապա պետք է օգտագործել **k-1** հասցեական մուտքերով մուլտիպլեքսոր:

7.2. Ավտոմատի նախագծման հաջորդականությունը և բացատրագրի ձևավորումը

Կուրսային աշխատանքում ավտոմատի նախագծման հաջորդականությունը և բացատագրի բովանդակությունը պետք է իրականացնել ըստ հետևյալ քայլերի.

- Առաջադրանքի բովանդակությունը,
- Ավտոմատի անցումային գրաֆի կառուցումը,

- Կառուցվածքային ավտոմատի սինթեզում,
  - Ավտոմատի վիճակների կոդավորում,
  - Ավտոմատի տրիգերի տեսակի ընտրում,
  - Ավտոմատի անցումների կոդավորված աղյուսակի ձևավորում,
  - Ավտոմատի գրգռման և ելքային ֆունկցիաների որոշումը,
  - Ավտոմատի գրգռման և ելքային ֆունկցիաների մինիմացումը Կառնոյի քարտերի միջոցով,
  - Ավտոմատի սխեմայի կառուցումը ԵՎ, ԿԱՄ, ՈՉ բազիսում,
  - Ավտոմատի սկզբունքային սխեմայի կառուցումը տրված սերիայի միկրոսխեմաներով և միկրոսխեմաների որակավորման աղյուսակի կազմումը:

8. ԳՐԱԿԱՆՈՒԹՅՈՒՆ

1. Ժ.Ճ¿Ր’ՎՏՉ. ՃՌԽՐՏՉՈԴ րււպՎՏՑպւվՌՍՈ. – ըղԹ – hՈվՍՑ- կպՑպՐոցՐ¿, 2004.
2. ԺՋ.Ճֆռնպրսռ. կՐՏպՍՑՌՐՏՉՈվՌպ ՓՌԽՐՏՉօւ ցրՑՐՏռրՑՉ. 2002.