ბათუმის შოთა რუსთაველის სახელმწიფო
უნივერსიტეტი

თენგიზ ხინიკაძე

კომპიუტერის არითმეტიკული და ლოგიკური საფუძვლები

დამტკიცებულია სახელმძღვანელოდ
რსუ აკადემიური საბჭოს მიერ
27.07.2012 დადგენილება №127

ბათუმი - 2012

სახელმძღვანელოში განხილულია ციფრული ტექნიკის თეორიული ბაზის - ორობითი არითმეტიკისა და ბულის ალგებრის - საფუძვლები, კომპიუტერის კომბინაციური და მიმდევრობითი ლოგიკისა და ძირითადი ოპერაციული ელემენტების სინთეზისა და ანალიზის საკითხები. დიდი ადგილი აქვს დათმობილი ლოგიკის ალგებრის ფუნქციათა მინიმიზაციას. იგი წარმოადგენს ბაზისს კომპიუტერის არქიტექტურისა და ორგანიზაციის შესასწავლად და განკუთვნილია კომპიუტინგის სპეციალობის ბაკალავრიატის საფეხურის სტუდენტებისა და კომპიუტერის საფუძვლების შესწავლის ყველა მსურველისათვის.

რედაქტორი:
გრიგოლ კახიანი - ბათუმის შოთა რუსთაველის სახელმწიფო უნივერსიტეტის განათლებისა და მეცნიერებათა ფაკულტეტის კომპიუტერულ მეცნიერებათა დეპარტამენტის ასისტენტ პროფესორი

რეცენზენტები:
გ. გოგიჩაიშვილი - ტექნიკურ მეცნიერებათა დოქტორი, საქართველოს ეროვნული აკადემიის წევრ კორესპონდენტი, საქართველოს პოლიტექნიკური უნივერსიტეტის ორგანიზაციული მართვის დეპარტამენტის უფროსი, სრული პროფესორი
ი. დიდმანიძე - ფიზიკა-მათემატიკის მეცნიერებათა დოქტორი, ბათუმის შოთა რუსთაველის სახელმწიფო უნივერსიტეტის კომპიუტერულ მეცნიერებათა დეპარტამენტის უფროსი, სრული პროფესორი

წინასიტყვაობა

თანამედროვე კომპიუტერები და სისტემები წარმოადგენენ სამეცნიერო და საინჟინრო აზრის ერთ-ერთ უმნიშვნელოვანეს მიღწევას, რომლის გავლენაც ადამიანის საქმიანობის ნებისმიერი სფეროს პროგრესზე უდიდესია. ამიტომაც ცხადია ის ყურადღება, რომელიც ექცევა გამოთვლითი სისტემების შესწავლას უმაღლეს პროფესიონალურ განათლებაში.

ციფრული ტექნიკის თეორიულ ბაზას ბულის ალგებრა, ორობითი არითმეტიკა და სასრულო ავტომატების თეორია წარმოადგენს, ამიტომ კომპიუტერის მუშაობის არითმეტიკული და ლოგიკური საფუძვლების გაგება მნიშვნელოვანია ინფორმატიკის სპეციალისტის მომზადებისათვის. ამასთან მისაწვდომი ხდება იმ პროცესების გაგება, რომლებიც წარმოიშობა კომპიუტერის არქიტექტურის ქვედა დონეზე.

უცხოურ ენებზე, ქართულისაგან განსხვავებით, ასეთი ლიტერატურა საკმაოდ ბევრია. ქართულ ენაზე გამოცემული ლიტერატურა, სამწუხაროდ, უკვე ბიბლიოგრაფიულ იშვიათობად იქცა, ამიტომ წინამდებარე ნაშრომი არსებული ხარვეზის შევსების მოკრძალებული ცდაა.

სახელმძღვანელო სამი ნაწილისაგან შედგება. პირველ ნაწილში განხილულია თვლის სისტემები, არითმეტიკული ოპერაციების ჩატარება თვლის ორობით სისტემაში, რიცხვების გადაყვანა თვლის ერთი სისტემიდან მეორეში, ორობით კოდირებული სისტემები, რიცხვების წარმოდგენა ფიქსირებული და მცურავი მძიმით, მათი კოდირება კომპიუტერში, თანრიგთა ბადის გადავსება, მოდიფიცირებული კოდები.

მეორე ნაწილში მოცემულია ბულის ლოგიკის ოპერაციები და კანონები, ლოგიკის ალგებრის წარმოდგენის ფორმები. დიდი ადგილი აქვს დათმობილი ლოგიკის ალგებრის ფუნქციების მინიმიზაციის მეთოდების განხილვას.

მესამე ნაწილში განხილულია კომბინაციური და მიმდევრობითი ლოგიკის ელემენტები, ციფრული ტექნიკის ძირითადი ოპერაციული ელემენტები, მათი სინთეზისა და ანალიზის საკითხები.

წინამდებარე ნაშრომში დიდი ადგილი აქვს დათმობილი პრაქტიკული მაგალითების განხილვას.

სახელმძღვანელოს საფუძვლად დაედო ავტორის მიერ შოთა რუსთაველის სახელმწიფო უნივერსიტეტში წლების განმავლობაში წაკითხული კურსები, პრაქტიკული და მეთოდური მუშაობის გამოცდილება.

ნაწილი I
კომპიუტერის არითმეტიკული საფუძვლები

1.1. თვლის სისტემები.

რაოდენობრივად განსხვავებული სიდიდის გამოსახვისათვის შემოღებულია სიმბოლო, რომელსაც ციფრი ეწოდება. ციფრი ან ციფრთა მიმდევრობა კი წარმოადგენს რ ი ც ხ ვ ს .

რიცხვების გამოსახვისათვის გვაქვს რომაული: **I** - ერთი, **V** - ხუთი, **X**- ათი, **L**- ორმოცდაათი, **C**- ასი , **D** - ხუთასი, **M** - ათასი და ინდური: **0, 1, 2, 3, 4, 5, 6, 7, 8, 9** ციფრები, მათ საკვანძო რიცხვები ეწოდება.

ადგილს რომელსაც იკავებს ციფრი, რიცხვის გამოსახვის დროს ეწოდება თანრიგი, თანრიგების ნუმერაცია ნატურალურ რიცხვში წარმოებს მარჯვნიდან მარცხნივ, ხოლო არაწესიერ წილადში მძიმიდან მარცხნივ (მთელი ნაწილისათვის) და მარჯვნივ (წილადი ნაწილისათვის).

წესების ერთობლიობას, რომელიც საშუალებას გვაძლევს სიმბოლოების (ციფრების) საშუალებით დავასახელოთ და ჩავწეროთ რიცხვები, თვლის სისტემა ეწოდება.

არსებობს თვლის პ ო ზ ი ც ი უ რ ი და ა რ ა პ ო ზ ი ც ი უ რ ი სისტემები.

არაპოზიციური ეწოდება თვლის ისეთ სისტემას, რომელშიც „წონა“ (მნიშვნელობა) დამოკიდებული არ არის იმ ადგილზე, ანუ პოზიციაზე, რომელიც მას უკავია ციფრთა მიმდევრობაში.

თვლის რომაული სისტემა არის არაპოზიციური, ვინაიდან ციფრები **I, V, X** და ა. შ. ინარჩუნებენ თავის მნიშვნელობას მიუხედავად იმისა, თუ რა ადგილი უკავია მათ ციფრთა ჯგუფში. მაგალითად: **III, XXX, VL , XXV.** აქ ციფრები **I, V და X** ყველა პოზიციაში

ინარჩუნებენ თავის მნიშვნელობას. შესაბამისად - ერთს, ხუთს, და ათს. თვლის რომაულ სისტემაში ყველა რიცხვი განისაზღვრება ორი არითმეტიკული ოპერაციის - შეკრებისა და გამოკლების საშუალებით:

1. როდესაც მიმდევრობით დაწერილია რამდენიმე ერთნაირი ციფრი, მაშინ რიცხვის მნიშვნელობა ამ ციფრების ჯამის ტოლია.

მაგალითად: **XX = X + X = 20, CCC = C+ C+ C = 300** და ა. შ.

2. თუ ციფრს მარცხნიდან მიუწერთ უფრო პატარა (ნაკლები სიდიდის აღმნიშვნელ) ციფრს, მაშინ რიცხვის მნიშვნელობა ამ ციფრების სხვაობის ტოლი იქნება; მაგალითად: **IV = V – I= 4, IX= X-I = 9, LM = M- L = 950.**

3. თუ ციფრის მარჯვნივ წერია უფრო პატარა (ნაკლები სიდიდის აღმნიშვნელი) ციფრი, მაშინ რიცხვის მნიშვნელობა ამ ციფრების ჯამის ტოლია; მაგალითად: **MCMLXXX = 1980.**

თვლის რომაული სისტემა თავისი სირთულის გამო არითმეტიკული მოქმედებებისათვის არ გამოიყენება.

პ ო ზ ი ც ი უ რ ი ეწოდება თვლის ისეთ სისტემას, რომელშიც ციფრი იცვლის თავის „წონას“ (მნიშვნელობას) იმ ადგილის, ანუ პოზიციის მიხედვით, რომელიც მას უკავია ციფრთა მიმდევრობაში.

თვლის პოზიციურ სისტემაში გამოყენებული ციფრების სიმრავლეს ხშირად უწოდებენ ანბანს და ისინი შეადგენენ თ ვ ლ ი ს ს ი ს ტ ე მ ი ს ბ ა ზ ა ს. ბაზის მიხედვით თვლის პ ო ზ ი ც ი უ რ ი სისტემები შეგვიძლია დავყოთ ორ ჯგუფად:

I - თვლის პოზიციური სისტემები ა რ ა უ ა რ ყ ო ფ ი თ ი ბ ა ზ ი თ;

II - თვლის პოზიციური სისტემები ს ი მ ე ტ რ ი უ ლ ი ბ ა - ზ ი თ, რომელშიც უარყოფითი და დადებითი ციფრები სიმეტრიულადაა განლაგებული ნულის მიმართ.

1.2. თვლის პოზიციური სისტემები არაუარყოფითი ბაზით

არაუარყოფითი ბაზის მქონე თვლის სისტემაში გამოიყენება ნული (0) და სხვა არაუარყოფითი ციფრები, ამიტომ თვლის ამ სის-ტემაში უარყოფითი რიცხვის ჩასაწერად შემოგვაქვს სპეციალური სიმბოლო „ - “.

ჩვენთვის ცნობილი თვლის ათობითი სისტემა არის პოზიცი-ური, არაუარყოფითი ბაზით, რომელიც შეიცავს ათ რიცხვს: 0, 1, 2, 3, 4, 5, 6, 7, 8 და 9.

ნებისმიერი ნატურალური A რიცხვი ათობით სისტემაში ჩა-იწერება შემდეგი სახით:

$$A = a_m a_{m-1} \dots a_2 a_1 a_0 = a_m \cdot 10^m + a_{m-1} \cdot 10^{m-1} + \cdots + a_2 \cdot 10^2 + \\ + a_1 \cdot 10^1 + a_0, \qquad (1.1)$$

სადაც a_1 ციფრები წარმოადგენენ კოეფიციენტებს და ისინი აკმაყო-ფილებენ შემდეგ პირობას: $0 \le a_i \le 9; \quad (i = 0 \div m)$.

პირველი თანრიგის a_0 ციფრი გვიჩვენებს რიცხვში შემავალი ერთეულების რაოდენობას, მეორე თანრიგის a_1 ციფრი - ათეულების რაოდენობას, მესამე თანრიგის a_2 ციფრი - ასეულების რაოდენობას და ა. შ.

როგორც ვხედავთ, ყოველი თანრიგის ერთეული ათჯერ მე-ტია მარჯვნივ მეზობლად მდებარე თანრიგის ერთეულზე, ამიტომ თვლის ამ სისტემას ათობითი ეწოდება, ხოლო რიცხვს ათი - თვლის სისტემის ფუძე. იგი ჩაიწერება ორი რიცხვის კომბინაციით „10“ და გამოსახავს ათ ერთეულს, ე. ი. იმდენ ერთეულს, როგორიც არის თვლის სისტემის დასახელება.

თვლის სისტემის ფუძე გვიჩვენებს, თუ რამდენჯერ მეტია რომელიმე თანრიგის „წონა“ მის მარჯვნივ მეზობლად მდებარე

თანრიგის წონაზე, ანუ რამდენი ციფრი გამოიყენება „თვლის აღებულ სისტემაში“.

თუ n - თანრიგა ათობითი A რიცხვი მოცემულია არაწესიერი ათწილადის სახით, სადაც რიცხვის მთელი ნაწილი შეიცავს m - თანრიგს, წილადი ნაწილი კი r -თანრიგს, მაშინ ეს რიცხვი ასე ჩაიწერება:

$$A = a_{m-1}a_{m-2} \dots a_2 a_1 a_0, a_{-1}a_{-2} \dots a_{-r} =$$
$$= a_{m-1} \cdot 10^{m-1} + a_{m-2} \cdot 10^{m-2} + \cdots + a_2 \cdot 10^2 + a_1 \cdot 10^1 + a_0 +$$
$$+ a_{-1} \cdot 10^{-1} + a_{-2} \cdot 10^{-2} + \cdots + a_{-r} \cdot 10^{-r} \qquad (1.2)$$

ადვილი შესამჩნევია, რომ თანრიგების წონას განსაზღვრავს ფუძის შესაბამისი ხარისხი (a -კოეფიციენტების ინდექსი ისეა შერჩეული, რომ იგი გვიჩვენებს თანრიგის შესაბამისი ფუძის ხარისხის მაჩვენებელს).

მაგალითად, ათობითი რიცხვი 535,05 შეგვიძლია ჩავწეროთ ჯამის სახით:

$$535{,}05 = 500 + 30 + 5 + 0{,}0 + 0{,}05 =$$
$$= 5 \cdot 10^2 + 3 \cdot 10^1 + 5 \cdot 10^0 + 0 \cdot 10^{-1} + 5 \cdot 10^{-2}$$

ციფრების წონა განსაზღვრულია იმ პოზიციით, რომელიც მათ უკავიათ, ე. ი. ციფრების წონას განსაზღვრავს შესაბამისი თანრიგების წონა.

თვლის პოზიციური სისტემის ფუძედ შეიძლება მივიღოთ ნებისმიერი ნატურალური k რიცხვი, რომელიც აკმაყოფილებს შემდეგ პირობას: $k \geq 2$. თვლის სისტემას, რომლის ფუძეა k, ვუწოდებთ k-ობითს. მასში გამოყენებულია k სიმბოლო: 0-დან $k-1$-ის ჩათვლით. როდესაც k-ობითი სისტემის რომელიმე თანრიგში ერთეულების რაოდენობა მიაღწევს k-ს, ამ თანრიგში ვწერთ ნულს და ერთიანი გადაგვაქვს უფროს თანრიგში, ამიტომ ყოველი თანრიგის წონა k-ჯერ მეტია მის მარჯვნივ მეზობლად მდებარე თანრიგის წონაზე, ე. ი. თანრიგების წონა რიცხვის მთელ ნაწილში ტოლია $1, k, k2, k3 \dots$, ხოლო წილად ნაწილში $k,\ k-1,\ k-2,\ k-3 \dots$ და ა. შ. თანრი-

გების ამ მიღებულ წონებს ბუნებრივი ეწოდება, მაგრამ შესაძლებელია თანრიგების წონების ხელოვნური მიმდევრობაც.

ისევე, როგორც ათობითი სისტემის ფუძე, k -ობითი სისტემის ფუძეც გამოისახება ორი ციფრის კომბინაციით „10“, მხოლოდ შეიცავს k ერთეულს.

გამოთვლით ტექნიკაში არაუარყოფითი ბაზის მქონე თვლის პოზიციური სისტემებიდან ძირითადად გამოიყენება: ათობითი, რვაობითი, სამობითი, თექვსმეტობითი, და ორობით-კოდირებლი სისტემები.

როდესაც $k \leq 10$, ვსარგებლობთ ჩვეულებრივი სიმბოლოებით 0, 1, 2, ... 9. მაგალითად, რ ვ ა ო ბ ი თ სისტემაში გამოიყენება რვა ციფრი: 0, 1, 2, 3, 4, 5, 6, 7; ს ა მ ო ბ ი თ სისტემაში სამი ციფრი: 0, 1, 2; ო რ ო ბ ი თ სისტემაში კი მხოლოდ ორი ციფრი: 0 და 1.

როდესაც $k > 10$, მაშინ საჭიროა შემოვიღოთ სპეციალური სიმბოლო 10-ის ტოლი და 10-ზე მეტი რიცხვების აღსანიშნავად. მაგალითად, თექვსმეტობით სისტემაში ჩვენთვის ცნობილი ათი სიმბოლოს (0, 1, 2, ..., 9) გარდა დამატებით შემოღებულია ექვსი ახალი სიმბოლო (ლათინური ანბანის დიდი ასოები):

A- ათი; B- თერთმეტი; C- თორმეტი;

D - ცამეტი; E - თოთხმეტი; F - თხუთმეტი.

თუ მთელ რიცხვს მარცხნიდან მივუწერთ ნებისმიერი რაოდენობის ნულებს ამით რიცხვის მნიშვნელობა არ შეიცვლება. მაგალითად: 15=015= 0015 და ა. შ.

თვლის სისტემის ფუძეებისათვის შგვიძლია შევადგინოთ ცხრილი 1.1.

ზოგადად შეგვიძლია ჩავწეროთ

$$k_{10} = 10_k^1 \tag{1.3}$$

მაგალითად: $2_{10}=10_2$; $3_{10}=10_3$; $5_{10}=10_5$; $8_{10}=10_8$ და $16_{10}=10_{16}$.

ცხრილ 1.2-ში მოცემულია ათობითი სისტემის საწყისი მთელი და წილადი რიცხვების ექვივალენტური გამოსახულება თვლის სხვადასხვა სისტემაში.

ცხრილი 1.1

ორობითი სისტემა	სამობითი სისტემა	რვაობითი სისტემა	ათობითი სისტემა	თექვსმე-ტობითი სისტემა
1 + 1 10_2 - ორ ერთეულს	2 + 1 10_3 - სამ ერთეულს	7 + 1 10_8 - რვა ერთეულს	9 + 1 10_{10} - ათ ერთეულს	F + 1 10_{16} - თექვსმეტ ერთეულს

ცხრილი 1.2

ათობითი სისტემა	სამობითი სისტემა	რვაობითი სისტემა	თექვსმეტობითი სისტემა
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C

13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14
21	10101	25	15
1/2	0,1	0,4	0,8
1/4	0,01	0,2	0,4
3/4	0,11	0,6	0,C
1/8	0,001	0,1	0,2
3/8	0,011	0,3	0,6
7/8	0,111	0,7	0, E
1/16	0,0001	0,04	0,1
3/16	0,0011	0,14	0,3
7/16	0,0111	0,34	0,7
15/16	0,1111	0,74	0, F

1.3. არითმეტიკული ოპერაციები თვლის ორობით სისტემაში.

ძალზე ადვილია არითმეტიკული ოპერაციების წარმოება თვლის ორობით სისტემაში. ამაში ადვილად დავრწმუნდებით ორობითი შეკრების, გამოკლების და გამრავლების ცხრილების განხილვით, რომელთაგან თითოეული შედგება ოთხი სტრიქონისგან:

ორობითი შეკრების ცხრილი	ორობითი გამოკლების ცხრ.	ორობითი გამრავლების ცხრ.
0 + 0=0	0 - 0=0	0 · 0=0
0 + 1=1	1 - 0=1	0 · 1=0
1 + 0=1	1 - 1=0	1 · 0=0
1 + 1=10	10 - 1=1	1 · 1=1

მ ა გ ა ლ ი თ ე ბ ი :

ათობითი ექვივალენტი

$$\begin{array}{r} 1101 \\ +\ \underline{11101} \\ 101010 \end{array} \qquad \begin{array}{r} 13 \\ +\ \underline{29} \\ 42 \end{array}$$

$$\begin{array}{r} 1000 \\ -\ \underline{1} \\ 111 \end{array} \qquad \begin{array}{r} 8 \\ -\ \underline{1} \\ 7 \end{array}$$

$$\begin{array}{r} 11001 \\ -\ \underline{1011} \\ 1110 \end{array} \qquad \begin{array}{r} 25 \\ -\ \underline{11} \\ 14 \end{array}$$

ვინაიდან მამრავლის ციფრები შეიძლება იყოს მხოლოდ 0 ან 1, ამიტომ კერძო ნამრავლები ნულის ან სამრავლის ტოლია.

გამრავლება შეიძლება ვაწარმოოთ სამრავლის როგორც მარცხნივ ისე მარჯვნივ ძვრით. მაშასადამე, გამრავლების ოპერაცია დაიყვანება, მამრავლის თანრიგების წონების შესაბამისად, მარცხნივ ან მარჯვნივ დაძრული სიმრავლის შეკრების ოპერაციამდე.

სამრავლის მარჯვნივ ძვრის დროს გამრავლების ოპერაცია იწყება სამრავლის გამრავლებით მამრავლის უფროსი თანრიგის ციფრზე.

სამრავლის მარჯვნივ ძვრა		სამრავლის მარცხნივ ძვრა		ათობითი ექვივალენტი
1101		1101		13
x		x		x
101		101		5
1101	კერძო	1101	I კ. ნ.	65
+	ნამრავ-	+		
1101	ლები	1101	II კ. ნ.	
1000001		1000001		
110111		110111		55
x		x		x
1111		1111		15
110111		110111		275
110111		110111		+ 55
+ 110111		+ 110111		825
110111		110111		
1100111001		1100111001		

ორობითი რიცხვების გაყოფისას განაყოფის თითოეული ციფრი არის 0 ან 1, ამიტომ ამ ციფრების დადგენა მალზე ადვილია.

```
_110100111110  | 101
 101           |-----------
               | 1010100110
  _110
   101
    _101
     101
      _111
       101
        _110
         101
          _101
           101
             0
```

```
3390 | 5
     |----
  39 | 678
   40
    0
```

როგორც ვხედავთ, არითმეტიკული მოქმედებები ორობით სისტემაში წარმოებს იმავე წესებით, რომლებითაც ათობით სისტემაში, მხოლოდ გაცილებით უფრო ადვილად.

ორი n - თანრიგა ორობითი წილადი რიცხვის გამრავლების შედეგად ნამრავლის თანრიგთა რაოდენობა იქნება $2n$ - ის ტოლი, მაგრამ კომპიუტერის მეხსიერების უჯრედი კონსტრუქციულად წინასწარ განსაზღვრულია და შეზღუდული; იგი ხშირად შეიცავს $2n$ თანრიგს, რის შედეგად, როდესაც გამრავლებას ვიწყებთ მამრავლის

უმცროსი თანრიგიდან, უჯრედში ვერ ჩაიწერება თანრიგის ციფრე-ბი, რაც გამოიწვევს საკმაოდ დიდ ცდომილებას.

ვთქვათ, გვაქვს ორი სამთანრიგა ორობითი წილადი, უჯრედი კი შეიცავს 5 თანრიგს (როგორც რიცხვების, ასევე უჯრედის თან-რიგთა რაოდენობა გაცილებით მეტია).

$$\begin{array}{r} 0,111 \\ \times \quad \\ \underline{0,101} \\ 111 \\ + \quad \\ \underline{111} \\ 0,100011 \end{array}$$

თუ ნამრავლს ჩამოეკვეთება წინა ერთი თანრიგი და უჯრედში ჩაიწერება 0,00011 აბსოლუტური ცდომილება ტოლი იქნება:

$$\begin{array}{r} 0,100011 \\ - \quad \\ \underline{0,00011} \\ 0,011101 \end{array}$$

აბსოლუტური ცდომილების შეფარდება თვით რიცხვთან გვაძლევს ფარდობით ცდომილებას:

$$\frac{0,011101}{0,100011}\cdot 100\% = \left(\frac{29}{64}:\frac{35}{64}\right)_{10}\cdot 100\% = \frac{29}{35}\cdot 100\% \approx 82,8\%$$

იმისათვის, რომ ნამრავლში დაგვრჩეს უფროსი თანრიგები და მას ჩამოეკვეთოს უმცროსი თანრიგები, მიმართავენ გამრავლებას, სამრავლის მარჯვნივ ძვრით:

$$\begin{array}{r} 0,111 \\ \times \quad\quad \\ \underline{0,101} \\ 111 \\ + \quad\quad \\ \underline{111} \\ 0,100011 \end{array}$$

თუ მიღებულ ნამრავლს ჩამოეკვეთება უმცროსი თანრიგი, დაგვრჩება დაგვრჩება 0,10001. ამ შემთხვევაში აბსოლუტური ცდომილება ტოლი იქნება:

$$\begin{array}{r} 0{,}100011 \\ -\;\underline{\;0{,}10001\;} \\ 0{,}000001 = \left(\frac{1}{64}\right)_{10} \end{array}$$

ფარდობითი ცდომილება კი -

$$\frac{1}{64}:\frac{35}{64}\cdot 100\% = \frac{1}{35}\cdot 100\% \approx 2{,}8\% \; .$$

1.4. რიცხვების გადაყვანა არაუარყოფითი ბაზის მქონე თვლის ერთი სისტემიდან მეორეში.

1.4.1. მთელი რიცხვების გადაყვანა თვლის ერთი სისტემიდან მეორეში.

მიმდევრობითი გაყოფის ხერხი

ვთქვათ k – ობითი სისტემის A_k მთელი რიცხვი გვინდა გადავიყვანოთ q - ობით სისტემაში.

დავუშვათ, რომ ნაპოვნია A_k რიცხვის q-ობითი ექვივალენტი და იგი წარმოადგენს რაღაც n+1 თანრიგა რიცხვს:

$$A_k = (a_n a_{n-1} \dots a_2 a_1 a_0)_q \qquad (1.4)$$

წარმოვიდგინოთ (2.4) ტოლობა ჯამის სახით, სადაც მარჯვენა მხარის q -ობითი კოეფიციენტები და სისტემის q - ფუძე გამოვსახოთ k- ობით სისტემაში:

$$A_k = (a_n)_k \cdot q_k^n + (a_{n-1})_k \cdot q_k^{n-1} + \cdots + (a_2)_k \cdot q_k^2 + (a_1)_k \cdot q_k + (a_0)_k \quad (1.5)$$

(1.5) ტოლობის ორივე მხარე გავყოთ ძველ სისტემაში გამოსახული ახალი სისტემის q_k ფუძეზე; არითმეტიკული მოქმედებები უნდა ვაწარმოოთ k - ობით (ძველ) სისტემაში. მივიღებთ პირველ განაყოფს A_{k_1}-ს, $A_{k_1} = (a_n)_k \cdot q_k^{n-1} + (a_{n-1})_k \cdot q_k^{n-2} + \cdots + (a_2)_k \cdot q_k$ და ნაშთს $(a_0)_k$, რომელიც წარმოადგენს A_k რიცხვის q - ობითი ექვივალენტის უმცროსი თანრიგის ციფრს გამოსახულს k – ობით სისტემაში. ცხადია

$$A_k = A_{k_1} \cdot q_k + (a_0)_k \, .$$

პირველი განაყოფი გავყოთ q_k-ზე, მივიღებთ მეორე განაყოფს $A_{k_2} = (a_n)_k \cdot q_k^{n-2} + (a_{n-1})_k \cdot q_k^{n-3} + \cdots + (a_2)_k$ და ნაშთს $(a_1)_k$, რომელიც A_k რიცხვის q - ობითი ექვივალენტის მეორე თანრიგის ციფრია. თუ გავაგრძელებთ მიღებული განაყოფების მიმდევრობით გაყოფას q_k-ზე, გაყოფის ოპერაციის n-ჯერ ჩატარების შემდეგ მივიღებთ ნაშთს $(a_{n-1})_k$ და განაყოფს $A_{k_n} = (a_n)_k$, რომელიც ნაკლებია q_k- ზე და წარმოადგენს A_k რიცხვის q - ობით ექვივალენტის უფროსი მე-(n+1) თანრიგის ციფრს გამოსახულს k -ობით სისტემაში. თვლის ერთი სისტემიდან მეორეში რიცხვების გადაყვანის დროს შეიძლება ადგილი ქონდეს ორ შემთხვევას:

1. რიცხვები დიდი ფუძის (მეტი სიდიდის) მქონე სისტემიდან გადაგვყავს პატარა ფუძის (ნაკლები სიდიდის) მქონე სისტემაში, ე.ი. $k > q$;

2. რიცხვები პატარა ფუძის მქონე სისტემიდან გადაგვყავს დიდი ფუძის მქონე სისტემაში, ე. ი. $k < q$.

I. როდესაც რიცხვები დიდი ფუძის მქონე სისტემიდან გადაგვყავს პატარა ფუძის მქონე სისტემაში $(k > q)$, მაშინ როგორც ახალი სისტემის ფუძე, ისე ბოლო განაყოფი და ყველა ნაშთი $(a_0 \div a_{n-1})$, რომლებიც წარმოადგენენ q- ობითი სისტემის ციფრებს გამოსახულს k- ობით სისტემაში, არიან ერთთანრიგა რიცხვები და მიეკუთვნებიან როგორც k-ობით, ასევე q - ობით სისტემას

მ ა გ ა ლ ი თ ე ბ ი:

1. ათობითი რიცხვი 976 გადავიყვანოთ რვაობით სისტემაში: $976_{10} \to 8^1$; $k = 10$; $q = 8$; ე. ი. $k > q$, ამიტომ $q_k = q = 8$. დაუშვათ, რომ ნაპოვნია ამ რიცხვის რვაობითი ექვივალენტი $976_{10} = 1720_8$.

ამ გამოსახულების მარჯვენა მხარე წარმოვადგინოთ ათობით სისტემაში ჯამის სახით

$$976_{10}=(1 \cdot 8^3 + 7 \cdot 8^2 + 2 \cdot 8^1 + 0)_{10}$$

ჩავატაროთ მიმდევრობითი გაყოფა $q = 8$ -ზე (ძველ სისტემაში). პირველი განაყოფი იქნება:

$$A_{k_1} = (1 \cdot 8^2 + 7 \cdot 8^1 + 2)_{10} \ ,$$

ხოლო პირველი ნაშთია „0“ , რომელიც გადასაყვანი რიცხვის q - ობითი ექვივალენტის უმცროსი თანრიგის ციფრია. მეორე განაყოფის შედეგად მივიღებთ მეორე განაყოფს $A_{k_2} = (1 \cdot 8^1 + 7)_{10}$ და

მეორე ნაშთს 2, მესამე გაყოფის შემდეგ კი მივიღებთ ბოლო განაყოფს 1-ს და ნაშთს 7.

ბოლო განაყოფი და ყველა ნაშთი ბოლოდან დასაწყისისაკენ შეადგენენ გადასაყვანი რიცხვის q - ობითი ექვივალენტის ციფრებს.

მართლაც:

976	8		
17	122	8	
16	42	15	8
0	2	7	1

ე. ი. $976_{10} = 1720_8$

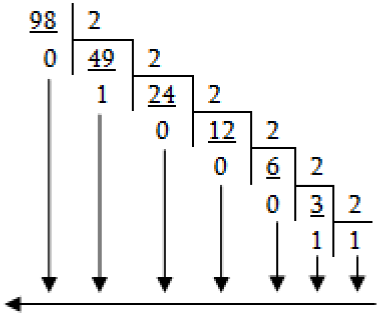
2. ათობითი რიცხვი 98 გადავიყვანოთ ორობით სისტემაში. დავუშვათ, ნაპოვნია მისი ორობითი ექვივალენტი:

$$98_{10} = 1100010_2 \ ; \ k = 10 \ ; \ q_k = q = 2 \ ; \ k > q \ .$$

ოპერაციები ჩავატაროთ პირველი მაგალითის ანალოგიურად:

$$98_{10} = (1 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0)_{10} \ .$$

უკანასკნელი გამოსახულების ორივე მხარის მიმდევრობით $q\ = 2$ -ზე გაყოფის შედეგად მივიღებთ გადასაყვანი რიცხვის ორობითი ექვივალენტის ყველა ციფრს. არითმეტიკულ ოპერაციებს ვაწარმოებთ ძველ სისტემაში.



ე. ი. $98_{10} = 1100010_2$.

II. როდესაც რიცხვები პატარა ფუძის მქონე სისტემიდან გადაგვყავს დიდი ფუძის მქონე სისტემაში, ე. ი. $k\ <\ q$. მაშინ ახალი სისტემის ფუძე ძველ სისტემაში q_k გამოისახება ორი, სამი ან ოთხთანრიგა რიცხვით (თუ ახალი სისტემის ფუძედ ავიღებთ 16-ზე მეტ რიცხვს, მაშინ q_k შეიცავს ოთხზე მეტ თანრიგს). ბოლო განაყოფი და ნაშთები $a_0 \div a_n$ კი შეიძლება იყვნენ როგორც ერთთანრიგა, ასევე ორი, სამი და ოთხთანრიგა რიცხვები. ბოლო განაყოფი და ნაშთები, რომლებიც შეიცავენ ერთზე მეტ თანრიგს, წარმოადგენენ k-ობითი სისტემის რიცხვებს, და ისინი უნდა შეიცვალონ შესაბამისი q -ობითი ექვივალენტებით.

მ ა გ ა ლ ი თ ე ბ ი:

3. რვაობითი რიცხვი 1720_8 გადავიყვანოთ ათობით სისტემაში. დაუშვათ, რომ ნაპოვნია მისი ათობითი ექვივალენტი:

$$1720_8 = 976_{10};\ \ k = 8\ ;\ \ q = 10\ ;\ \ k < q\ .$$

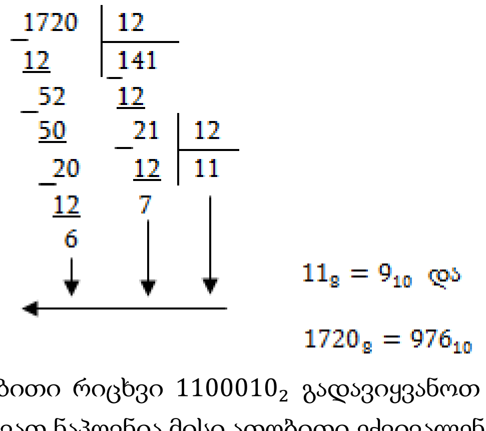
ახალი სისტემის ფუძე გამოვსახოთ ძველ სისტემაში, გვექნება $q_k = 12$.

გადასაყვანი რვაობითი რიცხვი შესაბამისი ათობითი ექვივალენტით გადავწერთ ისე, რომ ათობითი ექვივალენტის როგორც კოეფიციენტები, ასევე სხვადასხვა ხარისხის ფუძე გამოვსახოთ რვაობით სისტემაში ჯამის სახით:

$$1720_8 = (11 \cdot 12^2 + 7 \cdot 12^1 + 6 \cdot 12^0)_8$$

მიღებული გამოსახულების ორივე მხარე მიმდევრობით გავყოთ $q_k = 12$ -ზე არითმეტიკული ოპერაციები უნდა ვაწარმოოთ რვაობით (ძველ) სისტემაში.

ვინაიდან ათობითი ექვივალენტი სამთანრიგა რიცხვია, ორი მიმდევრობითი გაყოფის შედეგად მიიღეთ ბოლო განაყოფს - 11 და ორ ნაშთს $a_0 = 6$, $a_1 = 7$, რომლებიც წარმოადგენენ რვაობით რიცხვებს და ისინი უნდა შეიცვალონ სათანადო ათობითი ექვივალენტებით: $11_8 = 9_{10}$, ხოლო ერთთანრიგა რიცხვები 6 და 7 როგორც რვაობით, ისე ათობით სისტემაში ერთნაირად გამოისახება. საბოლოოდ გვექნება: $1720_8 = 976_{10}$.



4. ორობითი რიცხვი 1100010_2 გადავიყვანოთ ათობით სისტემაში. დავუშვათ ნაპოვნია მისი ათობითი ექვივალენტი:

$$1100010_2 = 98_{10}; \; k = 2; \; q = 10; \; k < q \, .$$

პირველ ყოვლისა ახალი სისტემის ფუძე $q = 10$ გამოვსახოთ (ძველ) ორობით სისტემაში $q_k = 1010$.

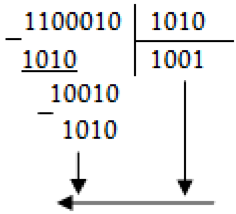
ოპერაციები ვაწარმოოთ მე-3 მაგალითის ანალოგიურად:

$$1100010_2 = 98_{10};\quad k = 2;\quad q = 10;\ k < q\ .$$

ვინაიდან მოცემული ორობითი რიცხვის ათობითი ექვივალენტი ორთანრიგა რიცხვია, ორივე მხარის $q_k = 1010$-ზე გაყოფის შედეგად მივიღებთ განაყოფს 1001 და ნაშთს 1000, რომლებიც ორობითი რიცხვებია, ამიტომ ისინი უნდა შეიცვალონ შესაბამისი ათობითი ექვივალენტებით

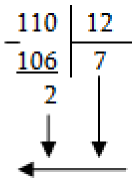
$$1001_2 = 9_{10} \text{ და } 1000_2 = 8_{10}\ .$$

მართლაც,



ე. ი. $1100010_2 = 98_{10}$.

5. $110_8 \to 10;\ k = 8;\ q = 10;\ k < q;\ q_k = 12;$



ე. ი. $110_8 = 72_{10}$.

არითმეტიკულ ოპერაციებს ვაწარმოებთ რვაობით (ძველ)სისტემაში.

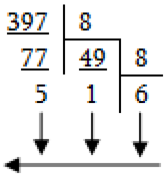
საბოლოოდ შეგვიძლია ჩამოვაყალიბოთ მთელი რიცხვების ერთი სისტემიდან მეორეში გადაყვანის მიმდევრობითი გაყოფის წესი: იმისათვის, რომ k- ობითი სისტემის A_k მთელი რიცხვი გადავიყვანოთ q -ობით სისტემაში, საჭიროა ახალი სისტემის q ფუძე გამოვსახოთ ძველ სისტემაში q_k და მოვახდინოთ მიმდევრობითი გა-

ყოფა q_k -ზე, გადასაყვანი რიცხვისა და მიღებული განაყოფების ნაშთების გამოყოფით, ვიდრე არ მივიღებთ განაყოფს, რომელიც ნაკლებია ახალი სისტემის q_k ფუძეზე. ბოლო განაყოფი და ყველა ნაშთი ამოწერილი ბოლოდან დასაწყისისაკენ მოგვცემენ რიცხვს ახალ (q -ობით) სისტემაში.

არითმეტიკული მოქმედებები უნდა ვაწარმოოთ ძველ (k-ობით) სისტემაში როდესაც $k < q$, თუ ბოლო განაყოფი ან რომელიმე ნაშთი შეიცავს ერთზე მეტ თანრიგს, ისინი k-ობითი რიცხვებია და უნდა შევცვალოთ შესაბამისი q -ობით ექვივალენტებით.

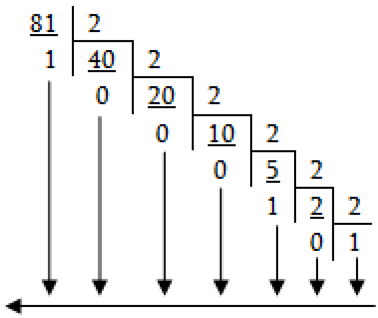
მ ა გ ა ლ ი თ ე ბ ი:

6. $397_{10} \to 8$; $k = 10$; $q = q_k = 8$;



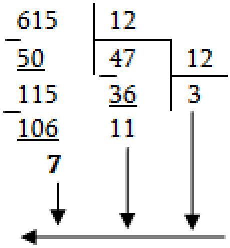
$397_{10} = 615_8$.

7. $81_{10} \to 2$; $k = 10$; $q = q_k = 2$;



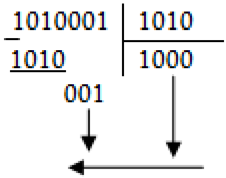
ე. ი. $81_{10} = 1010001_2$.

8. $615_8 \to 10$; კ $= 8$; ქ $= 10$; $10_{10} = 12_8$;



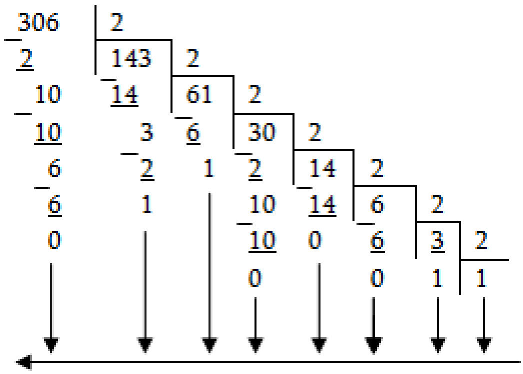
$11_8 = 9_{10}$ და $615_8 = 397_{10}$.

9. $1010001_2 \to 10$; $k = 2$; $q = 10$; $q_k 1010$;



$1000_2 = 8_{10}$ და $1010001_2 = 81_{10}$.

10. $306_8 \to 2$; $k = 8$; $q = q_k = 2$;



ე.ი. $306_8 = 11000110_2$.

11. $103_8 \rightarrow 10;\quad k = 8;\quad q = 10;\quad q_k = 12;$

$$\begin{array}{r|l} -103 & 12 \\ \cline{2-2} \underline{74} & 6 \\ 7 & \end{array}$$

ე. ი. $103_8 = 67_{10}$.

12. $117_8 \rightarrow 10;\quad k = 8;\quad q = 10;\quad q_k = 12;$

$$\begin{array}{r|l} -103 & 12 \\ \cline{2-2} \underline{74} & 6 \\ 7 & \end{array}$$

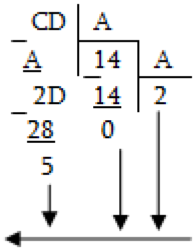
$11_8 = 9_{10}$ და $117_8 = 79_{10}$.

13. $205_{10} \rightarrow 16;\quad k = 10;\quad q = q_k = 16;$

$$\begin{array}{l|l} -103 & 12 \\ \cline{2-2} \underline{74} & 6 \\ 7 & \end{array}$$

$12_{10} = C_{16}$, $13_{10} = D_{16}$ და $205_{10} = CD_{16}$.

14. $CD_{16} \to 10;\quad k = 16;\quad q = 10;\quad q_k = A;$



ე. ი. $CD_{16} = 205_{10}$.

მთელი რიცხვების გადაყვანა თვლის ერთი სისტემიდან მეორეში კოეფიციენტებისა და ფუძის საშუალებებით (ცხრილური ხერხი)

წ ე ს ი : იმისათვის, რომ k-ობითი სისტემის A_k მთელი რიცხვი გადავიყვანოთ q -ობით სისტემაში, საკმარისია ამ რიცხვის როგორც a_iკოეფიციენტები, ისე სხვადასხვა ხარისხის მქონე k - ფუძე გამოვსახოთ q - ობით სისტემაში ჯამის სახით და არითმეტიკული ოპერაციებიც (გამრავლება და შეკრება) ვაწარმოოთ ამავე q - ობით სისტემაში, ე. ი. სისტემაში, რომელშიც გადაგვყავს რიცხვი. ამიტომ $A_k = a_n a_{n-1} a_{n-2} \dots a_2 a_1 a_0$ ჩავწეროთ ჯამის სახით q -ობით სისტემაში, გვექნება: $(A_k)_q = (a_n)_q \cdot k_q^n + (a_{n-1})_q \cdot k_q^{n-1} +$

$$+(a_{n-2})_q \cdot k_q^{n-2} + \dots + (a_2)_q \cdot k_q^2 + (a_1)_q \cdot k_q + (a_0)_q \quad (1.6)$$

გადაყვანის ამ ხერხით სარგებლობა უფრო ადვილია, როდესაც $q = 10$, ანუ, როდესაც ნებისმიერი სისტემიდან მთელი რიცხვები გადაგვყავს ათობით სისტემაში. ამ შემთხვევაში კოეფიციენტები და სხვადასხვა ხარისხის მქონე ფუძე ათობით სისტემაში პირდაპირ ჩაიწერება და არითმეტიკული ოპერაციებიც ათობით სისტემაში ჩვენთვის უფრო ჩვეული და ადვილია, ვიდრე სხვა რომელიმე სისტემაში (გარდა ორობითისა).

სხვადასხვა სისტემაში გამოსახული რიცხვები გადავიყვანოთ ათობით სისტემაში:

1. $100001111_2 = 1 \cdot 2^8 + 0 \cdot 2^7 + 0 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 +$
$+1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 256 + 8 + 4 + 2 + 1 = 271_{10}$;
2. $121021_3 = 1 \cdot 3^5 + 2 \cdot 3^4 + 1 \cdot 3^3 + 0 \cdot 3^2 + 2 \cdot 3^1 + 1 \cdot 3^0 =$
$= 243 + 162 + 27 + 6 + 1 = 439_{10}$;
3. $450_6 = 4 \cdot 6^2 + 5 \cdot 6^1 + 0 \cdot 6^0 = 4 \cdot 36 + 5 \cdot 6 = 144 + 30 =$
$= 174_{10}$;
4. $256_8 = 2 \cdot 8^2 + 5 \cdot 8^1 + 6 \cdot 8^0 = 128 + 40 + 6 = 174_{10}$;
5. $1A_{16} = 1 \cdot 16^1 + 10 \cdot 16^0 = 16 + 10 = 26_{10}$;
6. $D7_{16} = 13 \cdot 16^1 + 7 \cdot 16^0 = 208 + 7 = 215_{10}$;
7. $A0B_{16} = 10 \cdot 16^2 + 0 \cdot 16^1 + 11 \cdot 16^0 = 2560 + 11 = 2571_{10}$.

ზოგად შემთხვევაში, როდესაც k - ობითი სისტემიდან რიცხვები გადაგვყავს q -ობით სისტემაში, და $q \neq 10$, გვხვდება ორგვარი სიძნელე: პირველი მდგომარეობს a_i კოეფიციენტების და სხვადასხვა ხარისხის მქონე ძველი k ფუძის ახალი სისტემის ფუძით გამოსახვაში. ხოლო მეორე - ამ ახალ სისტემაში ოპერაციების ჩატარებაში; ამ უკანასკნელის დაძლევა შედარებით ადვილია, ვინაიდან საჭირო ოპერაციების (გამრავლებისა და შეკრების) ჩატარება ნებისმიერ სისტემაში შესაბამისი შეკრებისა და გამრავლების ცხრილების საშუალებით დიდ სიძნელეს არ წარმოადგენს. რაც შეეხება პირველ სიძნელეს, მას ადვილად ავიცილებთ თავიდან, თუ გამოვიყენებთ სპეციალურ ცხრილს, რომელშიც მოცემულია სხვადასხვა ხარისხის მქონე ძველი ფუძეების გადაყვანა ახალ სისტემაში; ამიტომ ამ ხერხს ზოგჯერ ცხრილურ ხერხს უწოდებენ.

ათობითი რიცხვები 87 და 986 გადავიყვანოთ რვაობით და ორობით სისტემებში:

8. $87 = 8 \cdot 10 + 7 = (10 \cdot 12 + 7)_8 = 120 + 7 = 127_8$;
9. $87 = (1000 \cdot 1010 + 111)_2 = 1010000 + 111 = 1010111_2$;

$$10.\ 986 = 9 \cdot 10^2 + 8 \cdot 10 + 6 = (11 \cdot 144 + 10 \cdot 12 + 6)_8 =$$
$$= 1604 + 120 + 6 = 1732_8;$$
$$11.\ 986 = (1001 \cdot 1100100 + 1000 \cdot 1010 + 110)_2 =$$
$$= 1111011010_2.$$

მიუხედავად იმისა, რომ ფუძის კვადრატების მნიშვნელობა ავიღეთ სპეც. ცხრილიდან, ამ ხერხით ათობითი რიცხვების გადაყვანა სხვა რომელიმე სისტემაში შედარებით უფრო ძნელი და შრომატევადია, ვიდრე მიმდევრობითი გაყოფის ხერხი.

12. ათობითი რიცხვი 271_{10} გადავიყვანოთ ორობით სისტემაში, გვექნება:

$$271_{10} = (2 \cdot 10^2 + 7 \cdot 10^1 + 1 \cdot 10^0)_{10} =$$
$$= (10 \cdot 1100100 + 111 \cdot 1010 + 1)_2 =$$
$$= (11001000 + 1000110 + 1)_2 = 100001111_2\,.$$

არითმეტიკული ოპერაციები ჩავატარეთ ორობით სისტემაში.

13. ათობითი რიცხვი 759_{10} გადავიყვანოთ რვაობით სისტემაში:

$$759_{10} = (7 \cdot 10^2 + 5 \cdot 10^1 + 9 \cdot 10^0)_{10} = (7 \cdot 144 + 5 \cdot 12 + 11)_8 =$$
$$= 1274 + 62 + 11 = 1367_8\,.$$

14. $AB1E_{16}$ გადავიყვანოთ ათობით სისტემაში:

$$AB1E_{16} = (10 \cdot 16^3 + 11 \cdot 16^2 + 1 \cdot 16^1 + 14)_{10} =$$
$$= (40960 + 2816 + 16 + 14)_{10} = 43806_{10}\,.$$

კიდევ უფრო გაგვიადვილდება რიცხვების გადაყვანა ათობით სისტემაში, თუ (2.6) ტოლობას ჩავწერთ ჰორნერის სქემით:

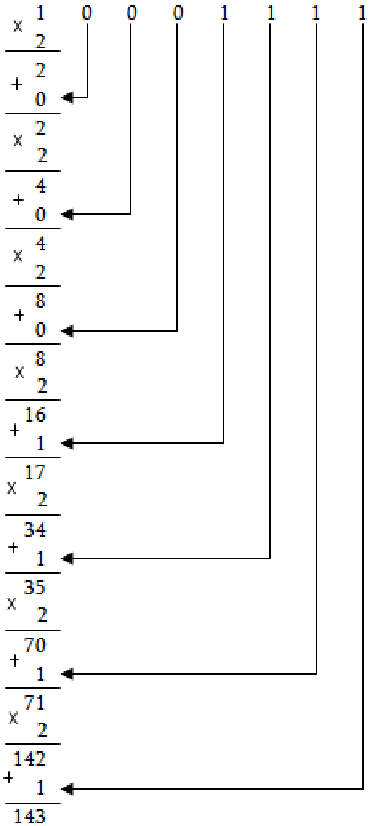
$$(A_k)_q = \left(\left(\left(a_{(n)q} \cdot k_q + a_{(n-1)q}\right)k_q + a_{(n-2)q}\right)k_q + \cdots + a_{(2)q}\right)k_q +$$
$$+ a_{(1)q}k_q + a_{0q}$$

ძველი სისტემის როგორც კოეფიციენტები, ასევე ფუძე უნდა გამოვსახოთ ახალ სისტემაში, არითმეტიკულ ოპერაციებს ვაწარმოებთ ახალ q - ობით სისტემაში

ოპერაციები წარმოებს შემდეგი მიმდევრობით: უფროსი თანრიგის ციფრის ნამრავლს ფუძეზე ვუმატებთ შემდეგი თანრიგის

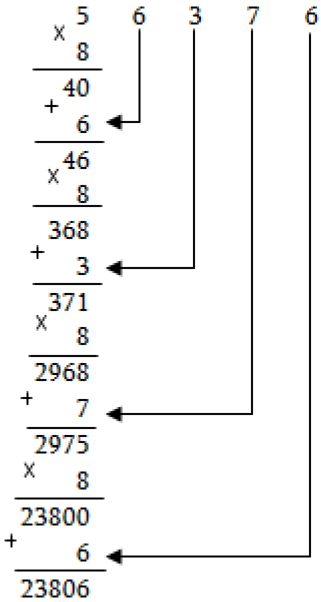
ციფრს; მიღებულ ჯამს ვამრავლებთ ფუძეზე, მიღებულ ნამრავლს ვუმატებთ შემდეგი თანრიგის ციფრს და ა.შ. ბოლოს ვუმატებთ უმცროსი თანრიგის ციფრს.

15. $A_2 = 10001111_2 \rightarrow 10.$



ე. ი. $10001111_2 = 143_{10}$

16. $A_8 = 56376_8 \rightarrow 10.$



ე. ი. $56376_8 = 23806_{10}$

17. $B6EC_{16} \to 10$

$$
\begin{array}{r}
B \quad 6 \quad E \quad C \\
\times\ 11 \\
16 \\
\hline
+\ 66 \\
11 \\
\hline
+\ 176 \\
6 \\
\hline
\times\ 182 \\
16 \\
\hline
+\ 1092 \\
182 \\
\hline
+\ 2912 \\
14 \\
\hline
\times\ 2926 \\
16 \\
\hline
+\ 17556 \\
2926 \\
\hline
+\ 46816 \\
12 \\
\hline
46828
\end{array}
$$

ე. ი. $B6EC_{16} = 46828_{10}$

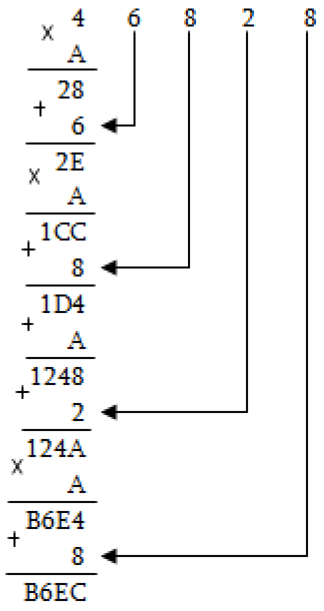
18. $23806_{10} \to 8$

$$
\begin{array}{r}
2 \quad 3 \quad 8 \quad 0 \quad 6 \\
\times\ 2 \\
12 \\
\hline
+\ 24 \\
3 \\
\hline
\times\ 27 \\
12 \\
\hline
+\ 56 \\
27 \\
\hline
+\ 346 \\
10 \\
\hline
\times\ 356 \\
12 \\
\hline
+\ 734 \\
356 \\
\hline
+\ 4514 \\
0 \\
\hline
\times\ 4514 \\
12 \\
\hline
+\ 11230 \\
4514 \\
\hline
+\ 56370 \\
6 \\
\hline
56376
\end{array}
$$

ე. ი. $23806_{10} = 56376_8$

შედარებით რთულია ჰორნერის სქემით რიცხვების გადაყვანა სხვა ფუძის მქონე სისტემაში, ვინაიდან არითმეტიკული ოპერაციები უნდა ჩავატაროთ იმ სისტემაში, რომელშიც გადაგვყავს რიცხვი.

19. $46828_{10} \rightarrow 16$

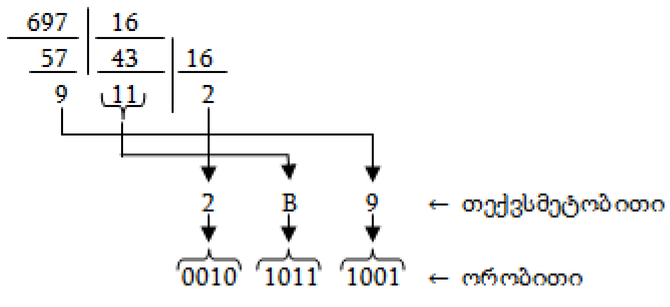


ე. ი. $46828_{10} = B6EC_{16}$

როგორც ვხედავთ, ზედა ორ მაგალითში მიმდევრობითი გაყოფის ხერხის გამოყენება გაცილებით უფრო ადვილია.

იმისათვის, რომ ათობითი რიცხვი გადავიყვანოთ ორობით სისტემაში, ჯერ ათობითი რიცხვი მიმდევრობითი გაყოფით უნდა გადავიყვანოთ თექვსმეტობით სისტემაში და შემდეგ ტეტრადების საშუალებით თექვსმეტობითი რიცხვი - ორობით სისტემაში.

20. $697_{10} \rightarrow 2$



ორობითი რიცხვების ათობით სისტემაში გადასაყვანად ორობითი რიცხვი ტეტრადებით გადაგვყავს თექვსმეტობით სისტემაში, შემდეგ ჰორნერის სქემით თექვსმეტობითი რიცხვი - ათობით სისტემაში.

$A_2 = 101 \quad 1111 \quad 1011 \quad 1000 \quad 1100_2 \quad \rightarrow 10$

$0101 \quad 1111 \quad 1011 \quad 1000 \quad 1100 \quad \leftarrow$ *ორობითი*

$5 \quad F \quad B \quad 8 \quad C \quad \leftarrow$ *თექვსმეტობითი*

$$5 \times 16 = 80$$
$$80 + 15 = 95 \quad (F)$$
$$95 \times 16 = 1520$$
$$1520 + 11 = 1531 \quad (B)$$
$$1531 \times 16 = 24496$$
$$24496 + 8 = 24504 \quad (8)$$
$$24504 \times 16 = 392064$$
$$392064 + 12 = 392076 \quad (C)$$

$392076 \leftarrow$ *ათობითი*

1.4.2. წილადების გადაყვანა თვლის ერთი სისტემიდან მეორეში.

<u>*მიმდევრობითი გამრავლების ხერხი.*</u>

ვთქვათ, k -ობითი სისტემის წილადი რიცხვი A_k გვინდა გადავიყვანოთ q - ობით სისტემაში. დაუშვათ, რომ ნაპოვნია მისი q -ობითი ექვივალენტი

$$A_k = (0, a_{-1}a_{-2}a_{-3} \dots a_{-n})_q \ . \quad (1.7)$$

(1.7) ტოლობა გადავწეროთ ჯამის სახით

$$A_k = (a_{-1})_k \cdot q_k^{-1} + (a_{-2})_k \cdot q_k^{-2} + (a_{-3})_k \cdot q_k^{-3} + \cdots + (a_{-n})_k \cdot q_k^{-n}, \ (1.8)$$

სადაც მარჯვენა მხარის q - ობით კოეფიციენტები და სისტემის q ფუძე ჩაწერილია k -ობითი სისტემაში.

(1.8) ტოლობის ორივე მხარე გავამრავლოთ ძველ სისტემაში გამოსახულ ახალი სისტემის q ფუძეზე. არითმეტიკული მოქმედებები უნდა ვაწარმოოთ k-ობით (ძველ) სისტემაში. მივიღებთ პირველ ნამრავლს A_{k_1}-ს:

$$A_{k_1} = A_k \cdot q_k = (a_{-1})_k + (a_{-2})_k \cdot q_k^{-1} + (a_{-3})_k \cdot q_k^{-2} + \cdots + (a_{-n})_k \cdot q_k^{n-1},$$

სადაც $(a_{-1})_k$ ნამრავლის მთელი ნაწილი წარმოადგენს მოცემული A_k წილადის q - ობითი ექვივალენტის უფროსი (პირველი) თანრიგის ციფრს (კოეფიციენტს) გამოსახულს k -ობით სისტემაში.

პირველი ნამრავლის წილადი ნაწილი $A'_{k_1} = A_{k_1} - (a_{-1})_k$ გავამრავლოთ q_k-ზე, მივიღებთ მეორე ნამრავლს:

$$A_{k_2} = A'_{k_1} \cdot q_k = (a_{-2})_k + (a_{-3})_k \cdot q_k^{-1} + (a_{-n})_k \cdot q_k^{-n+2},$$

სადაც ნამრავლის მთელი ნაწილი $(a_{-2})_k$ გადასაყვანი A_k წილადის q - ობით ექვივალენტის მეორე თანრიგის ციფრია თუ გავაგრძელებთ ნამრავლებში მიღებული წილადი ნაწილების მიმდევრობით გამრავლებას $q_k -$ ზე, ნამრავლების მთელი ნაწილები მოგვცემენ A_k წილადის q - ობით ექვივალენტის მომდევნო კოეფიციენტებს გამოსა-

ხულს k -ობით სისტემაში, რომლებიც უნდა შეიცვალოს შესაბამისი q - ობით ექვივალენტებით.

როგორც მთელი რიცხვების დროს, ასევე წილადების თვლის ერთი სისტემიდან მეორეში გადაყვანისას შეიძლება ადგილი ჰქონდეს ორ შემთხვევას:

1. წილადები დიდი ფუძის მქონე სისტემიდან გადაგვყავს პატარა ფუძის მქონე სისტემაში, ე. ი. $k > q$;

2. წილადები პატარა ფუძის მქონე სისტემიდან გადაგვყავს დიდი ფუძის მქონე სისტემაში, ე. ი. $k < q$.

I. როდესაც $k > q$, მაშინ ახალი სისტემის ფუძე გამოსახული ძველ სისტემაში q_k , ისე ნამრავლების მთელი ნაწილები $(a_{-1})_k \div (a_{-n})_k$ ერთთანრიგა რიცხვებია, რომლებიც წარმოადგენენ q - ობით სისტემის ციფრებს და მიეკუთვნებიან როგორც k -ობით, ასევე q - ობით სისტემას.

მ ა გ ა ლ ი თ ე ბ ი:

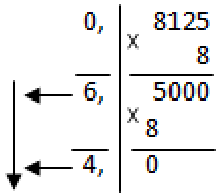
1. ათობითი წილადი 0,8125 გადავიყვანოთ რვაობით და ორობით სისტემაში.

ა) $0{,}8125_{10} \rightarrow 8;\ k = 10;\ q = 8;\ k > q$, ამიტომ $q_k = 8$;

დავუშვათ, რომ ნაპოვნია ამ რიცხვის რვაობითი ექვივალენტი - $0{,}8125_{10} = 0{,}64_8$.

უკანასკნელი გამოსახულების და ნამრავლების წილადი ნაწილების მიმდევრობით $q = 8$ -ზე გამრავლების შედეგად ნამრავლებში მთელი ნაწილების სახით მივიღებთ q - ობით წილადის ყველა კოეფიციენტს ზემოდან ქვემოთ (მარცხნიდან მარჯვნივ, წილადში უფროსი თანრიგიდან მოყოლებული).

პირველი ნამრავლის მთელი ნაწილი იქნება $6(6 \cdot 8^0 + 4 \cdot 8^{-1})$, მეორე გამრავლების შედეგად მივიღებთ 4-ს.



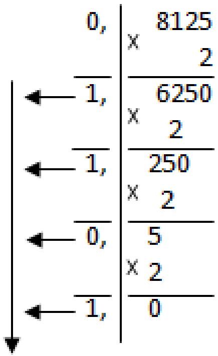
ბ) $0{,}8125_{10} \rightarrow 2;\quad k = 10;\ \ q = 2;\quad k > q;\quad q_k = q = 2.$

დავუშვათ, რომ ნაპოვნია გადასაყვანი წილადის ორობითი ექვივალენტი $0{,}8125_{10} = 0{,}1101_2$. ამ გამოსახულების მარჯვენა მხარე ათობით სისტემაში ჩავწეროთ ჯამის სახით:

$$0{,}8125_{10} = (1 \cdot 2^{1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4})_{10}\ .$$

შემდგომი ოპერაციები ჩავატაროთ ა) მაგალითის ანალოგიურად, რის შედეგად მივიღებთ q-ობითი წილადის ყველა კოეფიციენტს ზემოდან ქვემოთ $a_{-1} = 1,\ a_{-2} = 1,\ a_{-3} = 0$ და $a_{-4} = 1$.

მართლაც,

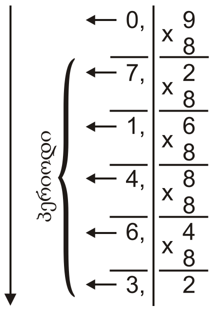


ე. ი. $0{,}8125_{10} = 0{,}1101_2$.

წილადების თვლის ერთი სისტემიდან მეორეში გადაყვანის დროს, მიმდევრობითი გამრავლების ჩატარების შედეგად, ნამრავლების წილადი ნაწილი ძალზე ხშირად არ ხდება ნულის ტოლი, ზოგჯერ ვღებულობთ პერიოდულ წილადებს, რომელთა დამრგვალებას ვახდენთ მოთხოვნილი სიზუსტის შესაბამისად, ე. ი. ადგილი

აქვს გადაყვანის გარკვეულ ცდომილებას, ამიტომ წილადის გადაყვანა ახალ სისტემაში მთავრდება მაშინ, როდესაც ნამრავლის წილადი ნაწილი ხდება ნულის ტოლი ან როდესაც მივაღწევთ სასურველ სიზუსტეს.

2. $0{,}9_{10} \rightarrow 8; \quad k = 10; \; q = 8; \; k > q; \; q_k = 8$.



განხილულ მაგალითში, ახალ სისტემაში გადაყვანის შედეგად მივიღეთ პერიოდული წილადი $0{,}9_{10} = 0{,}7(1463)_8$.

II. როდესაც წილადები პატარა ფუძის მქონე სისტემიდან გადაგვყავს დიდი ფუძის მქონე სისტემაში, ე. ი. $k < q$, მაშინ q_k ახალი სისტემის ფუძე ძველ სისტემაში q_k გამოისახება ორი, სამი, ოთხთანრიგა რიცხვით. ნამრავლების მთელი ნაწილები $(a_{-1})_k \div (a_{-n})_k$ კი შეიძლება იყოს როგორც ერთთანრიგა, ასევე ორი, სამი და ოთხთანრიგა რიცხვები. ნამრავლების მთელი ნაწილები, რომლებიც შეიცავენ ერთზე მეტ თანრიგს, წარმოადგენენ k -ობითი სისტემის რიცხვებს. ამიტომ ისინი უნდა შეიცვალონ შესაბამისი q-ობითი ექვივალენტებით.

მ ა გ ა ლ ი თ ე ბ ი:

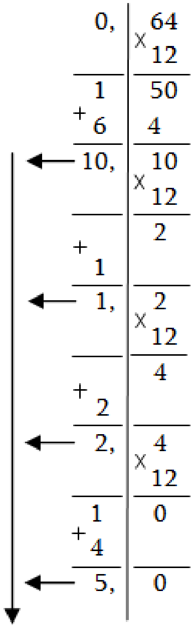
3. რვაობითი წილადი $0{,}64_8$ გადავიყვანოთ ათობით სისტემაში. დაუშვათ, ნაპოვნია მისი ათობითი ექვივალენტი $0{,}64_8 = 0{,}8125_{10}$. $k = 8$; $q = 10$; $k < q$; $q_k = 12$.

გადასაყვანი რვაობითი წილადი შესაბამისი ათობითი ექვივალენტით გადავწეროთ ისე, რომ ათობითი ექვივალენტის, როგორც კოეფიციენტები, ასევე სხვადასხვა ხარისხის მქონე ფუძე გამოვსახოთ რვაობით სისტემაში ჯამის სახით

$$0{,}64_8 = (10 \cdot 12^{-1} + 1 \cdot 12^{-2} + 2 \cdot 12^{-3} + 5 \cdot 12^{-4})_8 .$$

მიღებული გამოსახულების ორივე მხარე მიმდევრობით გავამრავლოთ $q_k = 12$ -ზე . პირველ ნამრავლში გამოვყოთ მთელი ნაწილი (10), მეორე ნამრავლში (1), მესამეში (2) და მეოთხე ნამრავლში (5).

არითმეტიკული ოპერაციები უნდა ვაწარმოოთ რვაობით (ძველ) სისტემაში:



ნამრავლების მთელი ნაწილები, რომლებიც შეიცავენ ერთზე მეტ თანრიგს, წარმოადგენენ რვაობით რიცხვებს და ისინი უნდა შეიცვალონ სათანადო ათობითი ექვივალენტებით.

პირველი ნამრავლის მთელი ნაწილი $a_{-1} = 10$ არის რვაობითი რიცხვი და იგი უნდა შევცვალოთ ათობითი ექვივალენტით $10_8 = 8_{10}$, ხოლო დანარჩენი ნამრავლების მთელი ნაწილები (1, 2 და 5) მიეკუთვნებიან როგორც რვაობით, ასევე ათობით სისტემას, ამიტომ

$0{,}64_8 = 0{,}8125_{10}$.

4. ორობითი წილადი $0{,}1101_2$ გადავიყვანოთ ათობით სისტემაში. დავუშვათ ნაპოვნია მისი ათობითი ექვივალენტი $0{,}1101_2 = 0{,}8125_{10}$;

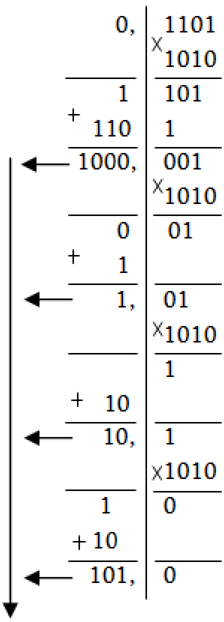
$k = 2$; $q = 10$; $k < q$; $q_k = 1010$.

ოპერაციები ვაწარმოოთ მე-3 მაგალითის ანალოგიურად, გვექნება

$$0{,}1101_2 = (1000 \cdot 1010^{-1} + 1 \cdot 1010^{-10} + 10 \cdot 1010^{-11} + 101 \cdot \cdot 1010^{-100})_2$$

უკანასკნელი გამოსახულების მიმდევრობით $q_k = 1010$ - ზე გამრავლების შედეგად, ნამრავლებში მივიღებთ მთელ ნაწილებს: (1000), (1), (10) და (101), რომლებიც წარმოადგენენ ორობით რიცხვებს და ისინი უნდა შეიცვალოს სათანადო ათობითი ექვივალენტებით: $1000_2 = 8_{10}$, $10_2 = 2_{10}$ და $101_2 = 5_{10}$, ე. ი. $0{,}1101_2 = 0{,}8125_{10}$.

ყველა სისტემაში 0,1-ზე გამრავლებიას მეორე თანამამრავლში მძიმე ერთი თანრიგით უნდა გადავწიოთ მარცხნივ და ა.შ.

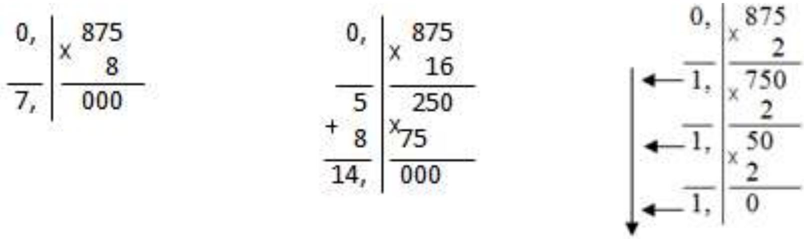


წილადების თვლის ერთი სისტემიდან მეორეში გადაყვანის მიმდევრობითი გამრავლების წ ე ს ი საბოლოოდ შეგვიძლია ჩამოვაყალიბოთ შემდეგნაირად: იმისათვის, რომ k-ობითი სისტემის A_k წესიერი წილადი გადავიყვანოთ q-ობით სისტემაში, საჭიროა ახალი სისტემის q ფუძე გამოვსახოთ ძველ სისტემაში - q_k და ვაწარმოოთ მიმდევრობითი გამრავლება - q_k -ზე გადასაყვანი წილადისა და მიღებული ნამრავლების წილადი ნაწილებისა, ნამრავლებში მთელი ნაწილების გამოყოფით. ეს მთელი ნაწილები (პირველი ნამრავლიდან მოყოლებული, რომელიც უფროსი თანრიგის ციფრია) წარმოადგენს q-ობითი წილადის თანრიგებს.

როდესაც, $k < q$ ნამრავლის მთელი ნაწილები, რომლებიც შეიცავენ ერთზე მეტ თანრიგს, k-ობითი რიცხვებია და ისინი უნდა შეიცვალონ სათანადო k-ობითი ექვივალენტებით.

არითმეტიკული მოქმედებები უნდა ვაწარმოოთ ძველ (k-ობით) სისტემაში. გადაყვანის პროცესი მთავრდება მაშინ, როდესაც ნამრავლის წილადი ნაწილი გახდება ნულის ტოლი ან როდესაც მივაღწევთ სასურველ სიზუსტეს.

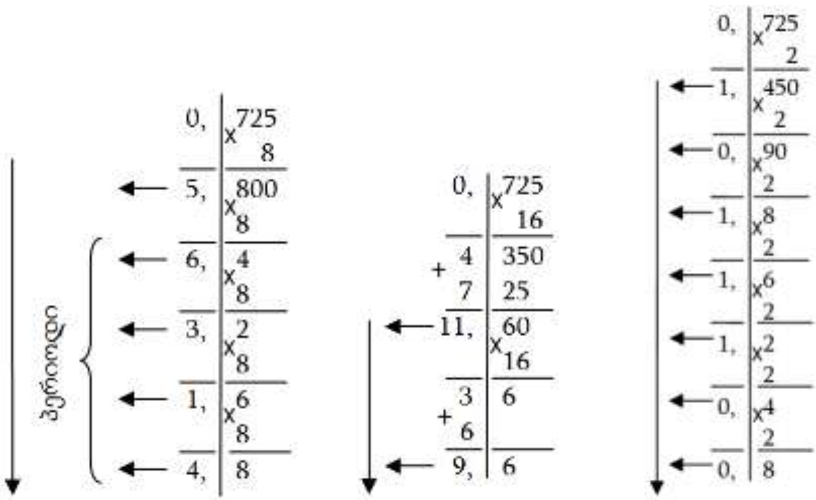
5. $0{,}875_{10}$ გადავიყვანოთ რვაობით, თექვსმეტობით და ორობით სისტემაში:



$0{,}875_{10} = 0{,}7_8$; $14_{10} = E_{16}$; $0{,}875_{10} = 0{,}111_2$;

ე. ი. $0{,}875_{10} = 0,E_{16} = 0{,}7_8 = 0{,}111_2$.

6. $0{,}725_{10}$ გადავიყვანოთ რვაობით, თექვსმეტობით და ორობით სისტემაში:



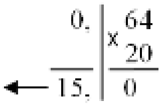
$$11_{10} = B_{16}$$

$0{,}725_{10} = 0{,}5(6314)_8$; $0{,}725_{10} = 0,B(9)_{16}$; $0{,}725_{10} = 0{,}101(1100)_2$;

ე. ი. $0{,}725_{10} = 0,B(9)_{16} = 0{,}5(6314)_8 = 0{,}101(1100)_2$.

7. $0{,}64_{\,8} \to 16$

$k = 8,\ \ q = 16,\ \ q_k = 20.$

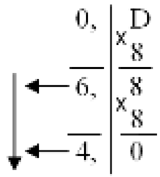


$15_8 = 13_{10} = D_{16}$;

ე. ი. $0{,}64_8 = 0,D_{16}$.

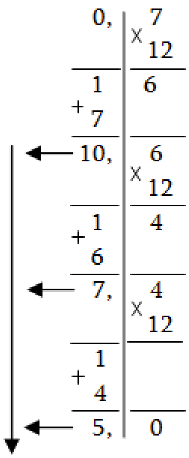
8. $0,D_{16} \to 8$

$k = 16,\ \ q = q_k = 8.$



$0,D_{16} = 0{,}64_8$;

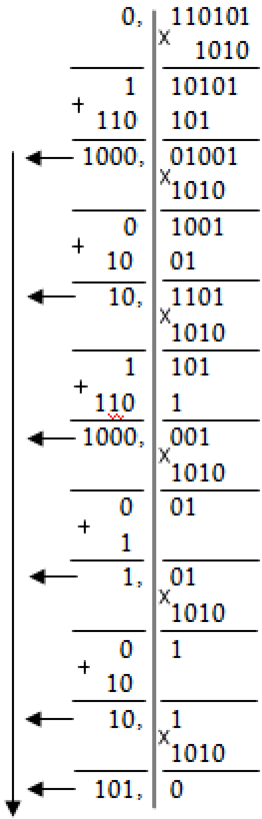
9. $0{,}7_8 \to 10$, $k = 8$, $q = 10$, $q_k = 12$.



არითმეტიკულ ოპერაციებს ვაწარმოებთ რვაობით სისტემა-ში.

$10_8 = 8_{10}$, ე. ი. $0{,}7_8 = 0{,}875_{10}$.

10. $0{,}110101_2 \rightarrow 10$; $k = 2$, $q = 10$ და $q_k = 1010$.

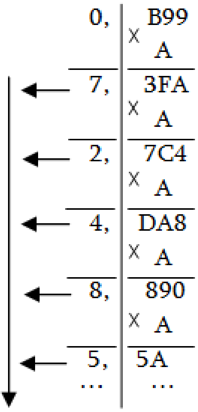


ნამრავლების მთელი ნაწილები, რომლებიც შეიცავენ ერთზე მეტ თანრიგს, ორობითი რიცხვებია და მათ შეესაბამე-ბათ, შემდეგი ათობითი ექვივალენტებია: $1000_2 = 8_{10}$, $10_2 = 2_{10}$, $101_2 = 5_{10}$;

ე. ი. $0{,}110101_2 = 0{,}828125_{10}$.

11. $0,B99_{16} \rightarrow 10$;

$k = 16$, $q = 10$, $q_k = A$.



9A – ათობით სისტემაში 90-ის ტოლია, თექვსმეტობით სის-ტემაში კი უდრის 5A -ს. $FA = 96_{16}$, ე. ი. $0,B99_{16} \approx 0{,}72485 \ldots$.

მიმდევრობითი გამრავლების ხერხით შედარებით ადვილია წილადების გადაყვანა ათობითი სისტემიდან სხვა რომელიმე პატა-

რა ფუძის მქონე სისტემაში, ვინაიდან, გარდა ზემოთ აღნიშნული მოსაზრებებისა, არითმეტიკულ ოპერაციებს ვატარებთ ჩვენთვის ცნობილ ათობით სისტემაში.

როდესაც პატარა ფუძიდან გადავდივართ დიდ ფუძეზე, დიდი ფუძე, ამ შემთხვევაში ახალი სისტემის ფუძე, ძველ სისტემაში გამოისახება ერთზე მეტი თანრიგით (მაგალითები 9, 10). ნამრავლების მთელი ნაწილიც ერთი, ორი, სამი და ოთხთანრიგა რიცხვებია. გარკვეულ სიძნელეს წარმოადგენს არითმეტიკული ოპერაციების წარმოება რვაობით, სამობით და სხვა ათობითისაგან განსხვავებულ სისტემებში. ამ მხრივ გამონაკლისს წარმოადგენს შემთხვევა, როდესაც პატარა ფუძე ათის ტოლია (მაგალითები: 5 და 6) თუ ათობითი სისტემიდან გადავდივართ თექვსმეტობით სისტემაში.

12. ათობითი წილადი $0{,}75_{10}$ გადავიყვანოთ თექვსმეტობით სისტემაში:

$$\begin{array}{r|l} 0, & \times\,75 \\ & \;\;\,16 \\ \hline +\;4 & \times\,50 \\ 7 & \;\;\,5 \\ \hline 12, & 00 \end{array}$$

ვინაიდან $12_{10} = C_{16}$, ამიტომ გვექნება $0{,}75_{10} = 0,C_{16}$.

წილადების გადაყვანა თვლის ერთი სისტემიდან მეორეში კოეფიციენტებისა და ფუძის საშუალებით (ცხრილური ხერხი)

იმისათვის, რომ k - ობითი სისტემის A_k წილადი, $A_k = 0, a_{-1}, a_{-2}, \dots, a_{-n}$ გადავიყვანოთ q - ობით სისტემაში, საჭიროა ამ წილადის როგორც კოეფიციენტები, ისე სხვადასხვა ხარისხის მქონე ფუძე q - ობით სისტემაში გამოვსახოთ ჯამის სახით

$$(A_k)_q = (a_{-1})_q \cdot k_q^{-1} + (a_{-2})_q \cdot k_q^{-2} + \dots + (a_{-n})_q \cdot k_q^{-n} \qquad (1.9)$$

და არითმეტიკული ოპერაციებიც (გამრავლება, შეკრება და გაყოფა) ვაწარმოოთ ამავე q-ობით სისტემაში, ე. ი. სისტემაში, რომელშიც გადაგვყავს წილადი.

წილადების გადაყვანის ამ ხერხით სარგებლობა უფრო ადვილია, როდესაც $q = 10$, ე. ი. როდესაც ნებისმიერი სისტემიდან წილადები გადაგვყავს ათობით სისტემაში, ვინაიდან კოეფიციენტები და სხვადასხვა ხარისხის მქონე ფუძე ათობით სისტემაში პირდაპირ ჩაიწერება და არითმეტიკული ოპერაციებიც ათობით სისტემაში უფრო ადვილია, ვიდრე სხვა რომელიმე სისტემაში.

მ ა გ ა ლ ი თ ე ბ ი:

1. ათობითი წილადი $0{,}89_{10}$ გადავიყვანოთ რვაობით სისტემაში. ამისათვის როგორც კოეფიციენტები, ისე ფუძე უნდა გამოვსახოთ რვაობით სისტემაში. მოქმედებებიც უნდა ვაწარმოოთ რვაობით სისტემაში. მივიღებთ:

$$0{,}89_{10} = (8 \cdot 10^{-1} + 9 \cdot 10^{-2})_{10} = (10 \cdot 12^{-1} + 11 \cdot 12^{-2})_8 =$$
$$= \frac{10}{12} + \frac{11}{144} - \frac{120+11}{144} = \left(\frac{131}{144}\right)_8,$$

მაშასადამე, $0{,}89_{10} = \left(\frac{131}{144}\right)_8 \approx 0{,}70753_8 \approx 0{,}71_8$.

2. რვაობითი წილადი $0{,}64_8$ გადავიყვანოთ ორობით სისტემაში, გვექნება

$$0{,}64_8 = (6 \cdot 8^{-1} + 4 \cdot 8^{-2})_{10} = \frac{6}{8} + \frac{4}{64} = \frac{48+4}{64} = \frac{52}{64} = \left(\frac{13}{16}\right)_{10} =$$
$$= 0{,}8125_{10}.$$

როგორც ვხედავთ, ამ წილადის გადაყვანა ათობით სისტემაში მეორე ხერხით უფრო ადვილია, ვიდრე პირველით.

3. ორობითი წილადი $0{,}0101101_2$ გადავიყვანოთ ათობით სისტემაში

$$0{,}0101101_2 = (0 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} + 1 \cdot 2^{-5} + 0 \cdot 2^{-6})_{10}$$
$$= \frac{1}{4} + \frac{1}{16} + \frac{1}{32} + \frac{1}{128} = \frac{32+8+4+1}{128} = \left(\frac{45}{128}\right)_{10} = 0{,}3515625_{10}\,.$$

კიდევ უფრო გაგვიადვილდება ამოცანა, სახელდობრ, აღარ დაგვჭირდება გაერთმნიშვნელიანების ოპერაცია, თუ გადასაყვან წილადს დასაწყისში განვიხილავთ, როგორც მთელ რიცხვს. მისი ახალ სისტემაში გადაყვანის შემდეგ იგი უნდა გავყოთ იმ ხარისხის ფუძეზე, რამდენი თანრიგიანიცაა წილადი.

ეს წესი გამოვიყენოთ ზემოთ განხილულ მე-2 და მე-3 მაგალითებისათვის, მივიღებთ:

4. მოცემულია $0{,}64_8$ მაშასადამე, ამ წილადს განვიხილავთ როგორც მთელ რიცხვს და გადავიყვანთ ათობით სისტემაში $64_8 = (6 \cdot 8^1 + 4 \cdot 8^2)_{10} = 48 + 4 = 52_{10}$. მიღებული რიცხვი უნდა გავყოთ ფუძეზე, რომლის ხარისხის მაჩვენებელი ორის ტოლია (ვინაიდან წილადი ორთანრიგაა, ანუ მასში გვაქვს ორი წილადი ნიშანი) და ისევე როგორც ზემოთ, მხოლოდ უფრო მოკლე და ადვილი გზით მივიღებთ

$$\frac{52}{8^2} = \frac{52}{64} = \frac{13}{16} = 0{,}8125_{10}\,.$$

5. $0{,}0101101_2$ - წილადი განვიხილოთ, როგორც მთელი რიცხვი $101101_2 = (1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0)_{10} = (32 + 8 + 4 + 1)_{10} = = 45_{10}$ და მიღებული რიცხვები გავყოთ $2^7 = 128$- ზე. ვინაიდან წილადი შვიდთანრიგაა და ისევე, როგორც ზემოთ, მივიღებთ

$$\frac{45}{128} = 0{,}3515625_{10}\,.$$

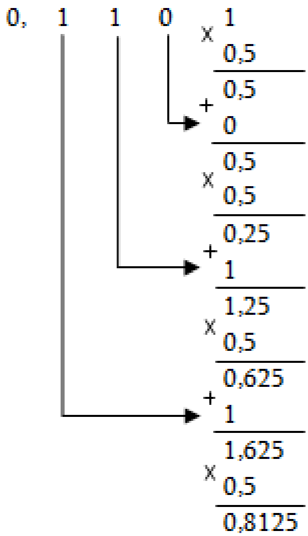
თუ (1.9) ტოლობას გადავწერთ ჰორნესის სქემით, მივიღებთ

$$(A_k)_q = k_q^{-1}\left(a_{(-1)q} + k_q^{-1}\left(a_{(-2)q} + \cdots + k_q^{-1}\left(a_{(-n+1)q} + a_{(-n)q} \cdot k_q^{-1}\right)\right)\right)$$

მთელი რიცხვების შემთხვევისაგან განსხვავებით, წილადების დროს ოპერაციებს ვიწყებთ უმცროსი თანრიგის ციფრის გამრავლებით ფუძეზე, ამ ნამრავლს ვუმატებთ შემდეგი თანრიგის ციფრს, ჯამს ვამრავლებთ ფუძეზე და ა. შ.

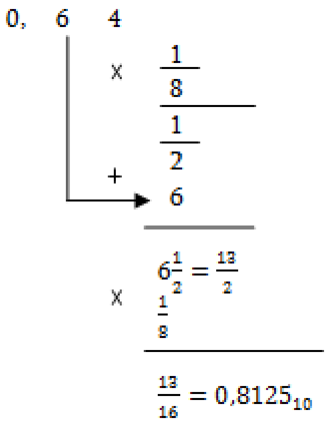
ვინაიდან ფუძე მოცემული გვაქვს უარყოფითი ხარისხით, ორობითი წილადების ათობით სისტემაში გადაყვანისას $k_q^{-1} = 0{,}5$.

6. ორობითი წილადი $0{,}1101_2$ გადავიყვანოთ ათობით სისტე-
მაში



ე. ი. $0{,}1101_2 = 0{,}8125_{10}$

7. $0{,}64_8 \to 10;$ $k_q^{-1} = \frac{1}{8}.$



წილადების ერთი სისტემიდან მეორეში გადაყვანის ორივე ხერხის განხილვის შემდეგ ვრწმუნდებით, რომ წილადების გადაყვანა მიმდევრობითი გამრავლების ხერხით შედარებით ადვილია, როდესაც ათობითი სისტემიდან წილადები გადაგვყავს სხვა რომელიმე სისტემაში, ვინაიდან არითმეტიკული მოქმედების წარმოება გვიხდება ჩვენთვის ცნობილ ათობით სისტემაში. პირიქით, თუ რომელიმე სისტემიდან წილადი გვინდა გადავიყვანოთ ათობით სისტემაში, მაშინ იმავე მიზეზების გამო უფრო ადვილია ამ ოპერაციის ჩატარება მეორე ხერხით, ანუ კოეფიციენტებისა და ფუძის ხერხით. წილადებისათვის ამ ხერხით სარგებლობის წესი სრულიად ისეთივეა, როგორც მთელი რიცხვებისათვის, ამიტომ თუ გადასაყვანი რიცხვი არაწესიერი წილადია, მაშინ როგორც მთელი ისე წილადი ნაწილებისათვის მეორე ხერხის გამოყენება საშუალებას გვაძლევს რიცხვის მთელი და წილადი ნაწილი კოეფიციენტებისა და ფუძის საშუალებით ერთდროულად გადავიყვანოთ თვლის ერთი სისტემიდან მეორეში. პირიქით, თუ გამოვიყენებთ პირველ ხერხს, მაშინ ცალ-ცალკე უნდა გადავიყვანოთ თვლის ერთი სისტემიდან მეორეში <u>მთელი რიცხვები მიმდევრობითი გაყოფის ხერხით, ხოლო წილადი ნაწილი მიმდევრობითი გამრავლების ხერხით.</u>

1.5. ორობით კოდირებული სისტემები. ტრიადებისა და ტეტრადების ცნება. თვლის ორობით - ათობითი სისტემა.

რიცხვების გადაყვანა თვლის ერთი სისტემიდან მეორეში ადვილად ხორციელდება მაშინ, როდესაც ერთი სისტემის ფუძე წარმოადგენს მეორე სისტემის ფუძის მთელ ხარისხს, ე. ი. თუ ადგილი აქვს ტოლობას

$$k = q^n \qquad (1.10)$$

სადაც n ნატურალური რიცხვია, და ორივე სისტემის ბაზა ან არაუარყოფითია, ან სიმეტრიული.

ასეთ შემთხვევაში რიცხვების გადაყვანას k-ობით სისტემიდან q-ობითში ვაწარმოებთ თანრიგობრივ, სახელდობრ: თითოეული k-ობითი ციფრი უნდა შევცვალოთ მისი ექვივალენტური n-თანრიგა q-ობითი რიცხვით. პირიქით, თუ გვინდა რიცხვი q-ობითი სისტემიდან გადავიყვანოთ k-ობითში, q-ობითი რიცხვი უნდა დავყოთ მძიმიდან მარცხნივ და მარჯვნივ n-თანრიგა ჯგუფებად, თუ რიცხვის საწყისი და ბოლო ჯგუფები არ შეიცავენ n თანრიგს, საჭიროა ეს ჯგუფები შევავსოთ n თანრიგამდე ნულების მიწერით: მარცხენა, პირველ ჯგუფს (მთელ ნაწილს) ნულები მიეწერება მარცხნიდან, ხოლო ბოლო ჯგუფს (წილად ნაწილს) - მარჯვნიდან; შემდეგ ეს ჯგუფები უნდა შევცვალოთ მათი ექვივალენტური k-ობითი ციფრებით.

როდესაც $q = 2$, მაშინ ძალზე ადვილია რიცხვების გადაყვანა თვლის რვაობითი და თექვსმეტობითი სისტემებიდან ორობით სისტემაში და, პირიქით, ვინაიდან $8 = 2^3$ და $16 = 2^4$.

ცხრილი 1.3

რვაობითი ციფრები	0	1	2	3	4	5	6	7
ორობითი ტრიადები	000	001	010	011	100	101	110	111

ცხრილი 1.4

ათობითი და თექვსმეტობითი ციფრები	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
ორობითი ტეტრადები	0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1

ცხრილ 1.3-ში მოცემულია რვაობითი ციფრები და მათი ექვივალენტური სამთანრიგა ორობითი რიცხვები, რომლებსაც ტრიადები ეწოდება, ხოლო ცხრ. 1.4-ში მოყვანილია ათობითი და თექვსმეტობითი ციფრები და მათი ექვივალენტური ოთხთანრიგა ორობითი რიცხვები, რომლებსაც ტეტრადებს უწოდებენ. თვით სახელწოდება ტრიადა და ტეტრადა ბერძნული წარმოშობისაა და შესაბამისად ნიშნავს სამს და ოთხს.

წესი: რვაობითი რიცხვების ორობით სისტემაში გადასაყვანად საკმარისია თითოეული რვაობითი ციფრი შევცვალოთ შესაბამისი ორობითი ტრიადით.

ორობითი რიცხვი რომ გადავიყვანოთ რვაობით სისტემაში, ამისათვის საჭიროა ორობითი რიცხვი დავყოთ ტრიადებად მძიმიდან მარცხნივ (რიცხვის მთელი ნაწილი) და მარჯვნივ (რიცხვის წი-

ლადი ნაწილი). თუ რიცხვის დასაწყისში და ბოლოს მივიღებთ არასრულ ტრიადებს, მაშინ ისინი უნდა შევავსოთ ნულებით: მთელ რიცხვებს ნულები უნდა მიუწეროთ მარცხნიდან, წილადების კი მარჯვნიდან; შემდეგ თითოეული ტრიადა უნდა შევცვალოთ შესაბამისი რვაობითი ციფრით.

მ ა გ ა ლ ი თ ე ბ ი:

1. რვაობითი რიცხვი $3705{,}24_8$ გადავიყვანოთ ორობით სისტემაში, ამისათვის თითოეული რვაობითი ციფრი შევცვალოთ შესაბამისი ტრიადით, მივიღებთ:

$$3705{,}24 = (011)(111)(000)(101),(010)(100) = 11111000101{,}0101_2\,.$$

მთელი რიცხვის წინ და წილადის ბოლოს მიწერილი ნულები გადავაგდეთ, ვინაიდან ისინი არ ცვლიან რიცხვის მნიშვნელობას.

2. ორობითი რიცხვი $11110{,}1011_2$ გადავიყვანოთ რვაობით სისტემაში. პირველ ყოვლისა მოცემული რიცხვი უნდა დავყოთ ტრიადებად მძიმიდან მარცხნივ და მარჯვნივ. აღებულ მაგალითში სრული ტრიადების მისაღებად რიცხვის მთელ ნაწილს მარცხნიდან უნდა მივუწეროთ ერთი ნული, ხოლო წილად ნაწილს კი მარჯვნიდან მივუწეროთ ორი ნული და მიღებული ტრიადები შევცვალოთ შესაბამისი რვაობითი ციფრებით, გვექნება:

$$11110{,}1011_2 = 11(110),(101)1 = (011)(110),(101)(100) = 36{,}54_8\,.$$

ანალოგიურად ხდება თექვსმეტობითი რიცხვების გადაყვანა ორობით სისტემაში და ორობითი რიცხვების გადაყვანა თექვსმეტობით სისტემაში, მხოლოდ იმ განსხვავებით, რომ თექვსმეტობითი ციფრები უნდა შევცვლოთ შესაბამისი ორობითი ტეტრადებით და ორობითი რიცხვი დავყოთ მძიმიდან მარცხნივ და მარჯვნივ ტეტრადებად, რომლებსაც შევცლით შესაბამისი თექვსმეტობით ციფრებით.

3. თექვსმეტობითი რიცხვი $DA,E6_{16}$ გადავიყვანოთ ორობით სისტემაში. ამისათვის ცხრილ 2-დან მოვძებნოთ მოცემულ რიცხვში შემავალი ციფრების შესაბამისი ტეტრადები, მივიღებთ:

$$DA,E6_{16} = (1101)(1010),(1110)(0110) = 11011010{,}1110011_2.$$

4. ორობითი რიცხვი $11111{,}0010101_2$ გადავიყვანოთ თექვსმეტობით სისტემაში.

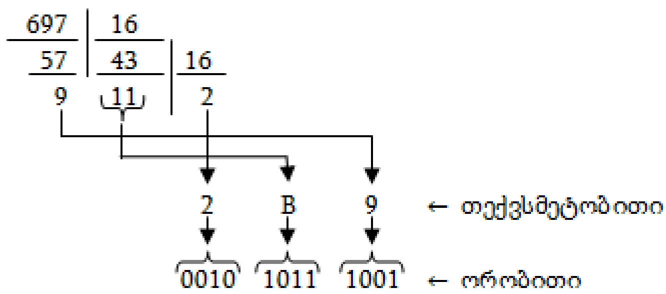
მოცემული რიცხვი მძიმიდან მარცხნივ და მარჯვნივ დავყოთ ტეტრადებად. საწყისი და ბოლო ტეტრადები შევავსოთ და ისინი ცხრილ 2-ის დახმარებით შევცვალოთ თექვსმეტობითი ციფრებით, მივიღებთ: $11111{,}0010101_2 = 1(1111),(0010)101 =$

$$= (0001)(1111),(0010)(1010) = 1F,1A\,.$$

ათობითი რიცხვების გადაყვანა ორობით სისტემაში პრაქტიკულად ხორციელდება ორ ეტაპად: პირველ ეტაპზე რიცხვი გადაგვყავს თექვსმეტობით (რვაობით) სისტემაში, მეორე ეტაპზე კი მიღებული თექვსმეტობით (რვაობითი) რიცხვი ტეტრადების საშუალებით გადაგვყავს ორობით სისტემაში.

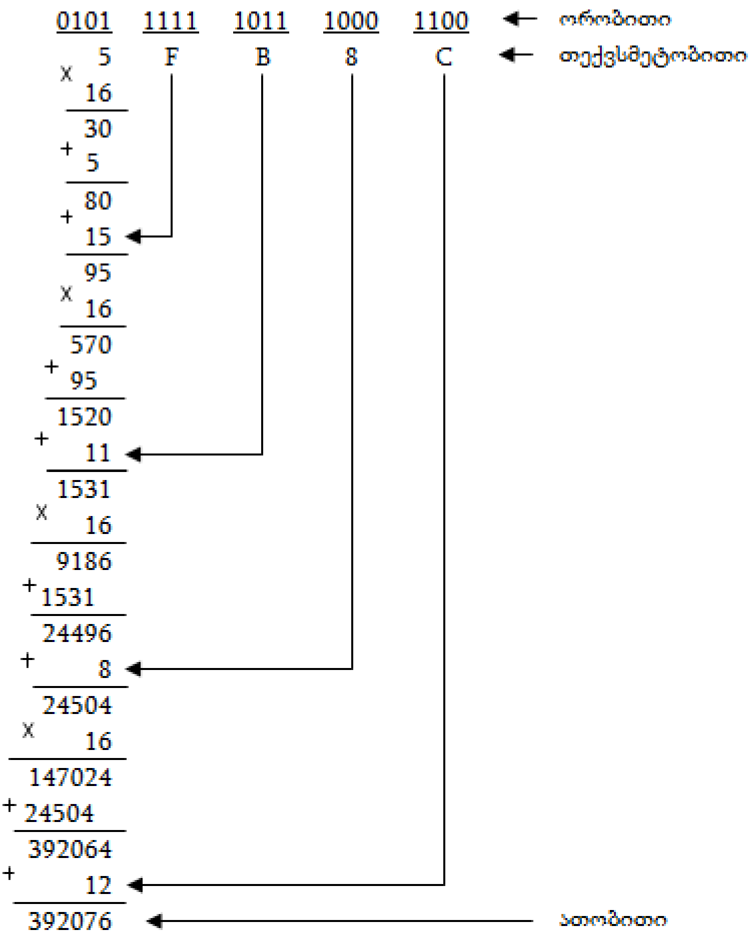
როგორც 1.4.1-ში იყო აღნიშნული, ათობითი რიცხვი რომ გადავიყვანოთ ორობით სისტემაში, საჭიროა ჯერ ათობითი რიცხვი მიმდევრობითი გაყოფით გადავიყვანოთ თექვსმეტობით სისტემაში, ხოლო შემდეგ ტეტრადების საშუალებით თექვსმეტობითი რიცხვი - ორობით სისტემაში.

5. $697_{10} \rightarrow 2$



ორობითი რიცხვების ათობით სისტემაში გადასაყვანად ორობითი რიცხვი ტეტრადებით გადაგვყავს თექვსმეტობით სისტემაში, შემდეგ ჰორნერის სქემით თექვსმეტობითი რიცხვი - ათობით სისტემაში:

$$A_2 = 101\ 1111\ 1011\ 1000\ 1100_2 \to 10$$



ასეთი გადაყვანის უპირატესობა ის არის, რომ ათობითი რიცხვის გადაყვანა თექვსმეტობით სისტემაში შედარებით უფრო ადვილია, ვიდრე უშუალოდ ორობით სისტემაში, ვინაიდან მოითხოვს გაცილებით ნაკლები რაოდენობის გაყოფის ოპერაციებს. თექვსმეტობითი რიცხვის გადაყვანა ორობით სისტემაში კი, როგორც ზემოთ დავინახეთ, ხორციელდება ძალზე მარტივად და არ მოითხოვს გამოთვლებს, ამიტომ პროგრამის შედგენისას ოპერაციის კოდებს და მისამართებს ვწერთ თექვსმეტობით ან რვაობით სისტემაში, რადგან ჩაწერა გამოდის მოკლე.

პროგრამა მანქანაში შეყვანის დროს ავტომატურად გადაიყვანება ორობით სისტემაში.

როდესაც $q = 2$ -ს, თუ (1.10) ტოლობის ადგილი არა აქვს, ე. ი. $k \neq 2^n$, მაშინ k მოთავსებული იქნება შემდეგ შუალედში

$$2^{n-1} < k < 2^n \tag{1.11}$$

ცხადია, ასეთ შემთხვევაში k -ობითი სისტემის ციფრები კოდირებული იქნება n - თანრიგა ორობითი რიცხვებით. მაგრამ ვინაიდან n - თანრიგა ორობითი რიცხვების რაოდენობა ტოლია 2^n –ის, ჩვენ კი გვესაჭიროება მხოლოდ k რაოდენობის ციფრების კოდირება, ამიტომ n - თანრიგა ორობითი რიცხვების გამოუყენებელი რაოდენობა იქნება$2^n - k$. ამ გამოუყენებელ რიცხვებს უწოდებენ ნულებისა და ერთიანების „ აკრძალულ კომბინაციებს.“

თუ $k = 10$ -ს, მაშინ (1.11)-ის თანახმად $n = 4$- ს.

ათობითი სისტემის რიცხვის ციფრების ოთხთანრიგა ორობითი რიცხვებით, ანუ ტეტრადებით კოდირებისას თანრიგების ბუნებრივი წონით (8, 4, 2, 1) ვღებულობთ ორობით-ათობით სისტემას და გამოუყენებელ ოთხთანრიგა ორობითი რიცხვების ანუ „აკრძალული კომბინაციების“ რაოდენობა ტოლია $2^n - k = 2^4 - 10 = 16 - 10 = 6$. ეს კომბინაციებია: 1010, 1011, 1100, 1101, 1110 და 1111.

ზოგიერთ კომპიუტერში გამოიყენება თვლის ორობით-ათობითი სისტემა, არითმეტიკული ოპერაციები ორობით-ათობით რი-

ცხვებზე წარმოებს სპეციალური წესების მიხედვით. თუ მანქანაში ოპერაციები წარმოებს ორობით სისტემაში, მაშინ თვლის ორობით-ათობითი სისტემა შუალედურია, მისი საშუალებით ათობითი რიცხვები შეგვყავს მანქანაში, შემდეგ კი თვით მანქანაში სპეციალური ფორმულების (კოეფიციენტებისა და ფუძის) საშუალებით ხდება ორობით-ათობითი რიცხვების გადაყვანა ორობით სისტემაში.

რიცხვების გადაყვანა ათობით სისტემიდან ორობით-ათობითში და ორობით-ათობითიდან ათობით სისტემაში ხდება იმავე წესებით, როგორც თექვსმეტობითი სისტემიდან ორობითში და ორობითი სისტემიდან თექვსმეტობითში.

6. ათობითი რიცხვი $915{,}48_{10}$ გადავიყვანოთ ორობით-ათობით სისტემაში. ამისათვის ცხრილ 2-ის საშუალებით რიცხვში შემავალი ყველა ათობითი ციფრი შევცვალოთ სათანადო ტეტრადებით, მივიღებთ:

$$915{,}48_{10} = (1001)(0001)(0101),(0100)(1000) = \\ = 100100010101{,}01001_{2-10}\,.$$

7. ორობითი რიცხვი $1001001{,}01101_{2-10}$ გადავიყვანოთ ათობით სისტემაში

$$1001001{,}01101_{2-10} = 100(1001),(0110)1 = \\ = (0100)(1001),(0110)(1000) = 49{,}68_{10}\,.$$

როგორც ვხედავთ, რიცხვების გადაყვანა ათობითი სისტემიდან ორობით-ათობითში და, პირიქით, სრულდება ძალზე მარტივად და არ არის დაკავშირებული რაიმე გამოთვლებთან.

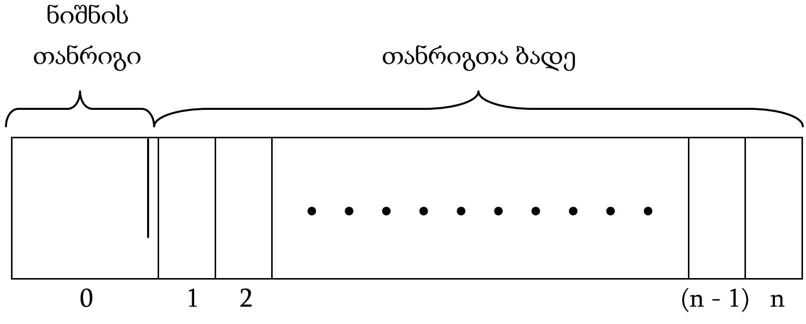
ზემოთ განხილულ სისტემებს აგრეთვე ორობით კოდირებულ სისტემებს უწოდებენ.

1.6. რიცხვების წარმოდგენა ფიქსირებული და მცურავი მძიმით.

კომპიუტერში გამოიყენება რიცხვების წარმოდგენის ორი ფორმა: ფიქსირებული მძიმით „ბუნებრივი ფორმა“ და მცურავი მძიმით „ნახევარლოგარითმული ფორმა“, რომელსაც ზოგჯერ უწოდებენ რიცხვების წარმოდგენას ნორმალური ფორმით.

რიცხვების ფიქსირებული მძიმით წარმოდგენის დროს, უჯრედში, რომელშიც უნდა ჩაიწეროს რიცხვი, მძიმე ფიქსირებულია გარკვეული თანრიგების შემდეგ.

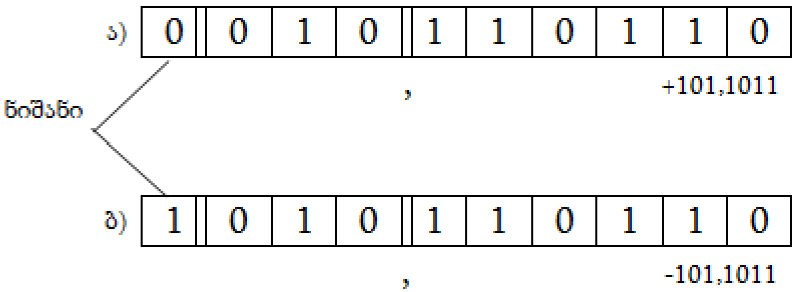
თანრიგების წონის ზრდა ხდება მარჯვნიდან მარცხნივ, ხოლო ნუმერაციისა მარცხნიდან მარჯვნივ. მარცხნიდან პირველი თანრიგი - ნულოვანი (ნახ. 1.1) განკუთვნილია რიცხვის ნიშნისათვის. ვინაიდან კომპიუტერები ოპერაციებს აწარმოებენ ორობით ან ორობით კოდირებულ რიცხვებზე, ნიშნის თანრიგში ჩაწერილი ნული იმის მაჩვენებელია, რომ რიცხვი დადებითია, ხოლო ერთიანი კი უარყოფითი რიცხვის მაჩვენებელია.



ნახ. 1.1.

უჯრედის დანარჩენი თანრიგების ერთობლიობას, სადაც ჩაწერილია რიცხვის ციფრული ნაწილი, თანრიგთა ბადე ეწოდება.

ვთქვათ, ათთანრიგა უჯრედში, სადაც ოთხი თანრიგი გამოყოფილია რიცხვის მთელი ნაწილისათვის, უნდა ჩავწეროთ რიცხვი $\pm 101{,}1011_2$. უჯრედებში მათ ექნებათ შემდეგი სახე (ნახ. 1.2 ა, ბ):



ნახ. 1.2.

ძირითადად მძიმე ფიქსირებულია ან უფროსი თანრიგის წინ, ან უმცროსი თანრიგის შემდეგ, პირველ შემთხვევაში უჯრედში წარმოდგენილია მხოლოდ წესიერი წილადები, მეორეში - მხოლოდ მთელი რიცხვები.

როდესაც გვაქვს $n + 1$- თანრიგა უჯრედი (ნახ. 1.1), რიცხვების წარმოდგენის დიაპაზონი წესიერი წილადების შემთხვევაში ასეთია

$$2^{-n} \leq |X| \leq 1 - 2^{n}.$$

მთელი რიცხვების დროს

$$1 \leq |X| \leq 2^{n} - 1\,.$$

როდესაც მანქანა ოპერაციებს აწარმოებს ფიქსირებული მძიმით წარმოდგენილ ოპერანდებზე როგორც საწყისი, ასევე შუალედური და საბოლოო სიდიდეები არ უნდა გამოდიოდნენ იმ დიაპაზონიდან, რომელიც მიღებული გვაქვს მანქანის უჯრედში, წინააღმდეგ შემთხვევაში შედეგი იქნება მცდარი. ამისათვის ამოცანის პროგრამის შედგენის დროს გამოთვლით ოპერაციებში მონაწილე სიდიდეები აიღებიან სათანადო სამაშტაბო კოეფიციენტებით.

დამასშტაბება შედარებით ადვილია, როდესაც ყველა რიცხვი მოდულით ნაკლებია ერთზე, ვინაიდან გამრავლების დროს თანრიგთა ბადის გადავსება გამორიცხულია, ხოლო გაყოფის დროს ავტომატურად ოპერაციის ჩატარების გარეშე შეიძლება თანრიგთა ბადის გადავსების კონტროლი (გასაყოფი მეტია გამყოფზე), თანრიგთა ბადის გადავსებას შეიძლება ადგილი ჰქონდეს, როდესაც ხდება ერთნაირი ნიშნის მქონე წესიერი წილადების შეკრება. თუ მძიმე ფიქსირებულია უმცროსი თანრიგის შემდეგ თანრიგთა ბადის გადავსებას უფრო ხშირად აქვს ადგილი გამრავლების ოპერაციის დროს. თანრიგთა ბადის გადავსების კონტროლი, როდესაც ოპერანდები წარმოადგენენ მთელ რიცხვებს, შედარებით რთულია. თანრიგა ბადის გადავსება შეიძლება მივიღოთ, როდესაც თანამამრავლთა თანრიგების რაოდენობის ჯამი მეტია ნამრავლისათვის განკუთვნილ თანრიგთა რაოდენობაზე. წინააღმდეგ შემთხვევაში თანრიგთა ბადის გადავსების დადგენა ხდება მხოლოდ ოპერაციის ბოლოს.

რიცხვების ფიქსირებული მძიმით წარმოდგენა საშუალებას გვაძლევს გავამარტივოთ მანქანის სქემები, გავზარდოთ მისი სწრაფქმედება, სამაგიეროდ გარკვეულ სიძნელეს წარმოადგენს სამაშტაბო კოეფიციენტის შერჩევა. პირველი კომპიუტერები წარმოადგენენ მანქანებს ფიქსირებულ მძიმით, სადაც მძიმე, როგორც წესი, ფიქსირებული იყო უფროსი თანრიგის წინ.

თანამედროვე მანქანებში გამოიყენება რიცხვების წარმოდგენა როგორც მცურავი მძიმით, რომელიც არ მოითხოვს მონაცემების დამაშტაბებას, ასევე ფიქსირებული მძიმით, ვინაიდან ოპერაციები ასეთ რიცხვებზე მოითხოვს ნაკლებ დროს.

რიცხვების წარმოდგენა მცურავი მძიმით (ნახევრად ლოგარითმული ფორმით) ზოგადად ჩაიწერება შემდეგნაირად:

$$X = k^{\pm\beta x} \pm M_x, \tag{1.12}$$

სადაც M_x არის რიცხვის მანტისა, რომელიც ყოველთვის $|M_x| < 1$;

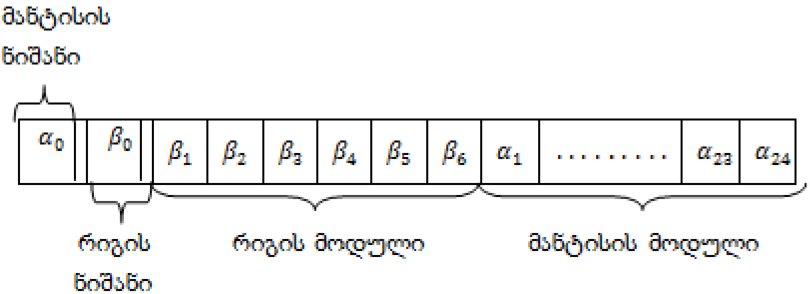
k – თვლის სისტემის ფუძე;

β_x- რიგი, რომელიც შეიძლება იყოს დადებითი ან უარყოფითი და მიუთითებს მძიმის ადგილს რიცხვში.

ერთსა და იმავე რიცხვში რიგის გაზრდისას ან შემცირებისას მძიმე შესაბამისად გადაიწევა მარცხნივ ან მარჯვნივ. ე. ი. დაცურავს რიცხვის გამოსახულებაში.

ჩანაწერს (1.12) ეწოდება ნახევრად ლოგარითმული იმიტომ, რომ ლოგარითმული ფორმით წარმოდგენილია არა მთელი რიცხვი, არამედ მხოლოდ მისი ნაწილი $k^{\pm\beta_x}$.

განვიხილოთ 32- თანრიგა ფორმატი (ნახ. 1.3), რომელშიც ჩაწერილია ორობითი სიტყვა: $\alpha_0\beta_0\beta_1 \dots \beta_6\alpha_1 \dots \alpha_{24}$. $\beta_0\beta_1 \dots \beta_6$ თანრიგები გამოსახავენ რიგს, სადაც β_0 გვიჩვენებს რიგის ნიშანს, ხოლო $\beta_1 \dots \beta_6$ თანრიგები - რიგის მოდულს.



ნახ. 1.3.

ნახევრად ლოგარითმული ფორმით წარმოდგენილი რიცხვი ნორმალიზებულია თუ მისი მანტისა აკმაყოფილებს შემდეგ პირობას

$$\frac{1}{k} \leq |M_x| < 1 \qquad (1.13)$$

(1.13) უტოლობა გვიჩვენებს, რომ ორობითი რიცხვი ნორმალიზებულია, თუ უფროს თანრიგში გვაქვს ერთიანი. როდესაც გვაქვს ორობით კოდირებული თვლის სისტემა ($8 = 2^3$, $16 = 2^4$),

მაშინ რიცხვის ნორმალიზაციისათვის საჭიროა, რომ შესაბამისად უფროსი ტრიადა, ტეტრადა განსხვავდებოდეს ნულისაგან.

დავუშვათ, მანტისას უფროსი t რაოდენობის თანრიგი ნულის ტოლია, მაშინ რიცხვის ნორმალიზაციისათვის უნდა მოვახდინოთ მანტისის t თანრიგით მარცხნივ ძვრა, შესაბამისად t ერთეულით უნდა შევამციროთ რიგი და მანტისის t უმცროს თანრიგებში ჩაიწე-რება ნულები.

მაგალითად, $x = 10^3(-0{,}085)$, $y = 2^2 \cdot 0{,}000101$ რიცხვები ნორმალიზაციის შემდეგ ჩაიწერება შემდეგი სახით:

$$x = 10^2(-0{,}850) \text{ და } y = 2^{-1} \cdot 0{,}101000\,.$$

რიცხვების წარმოდგენის დიაპაზონი დამოკიდებულია თვლის სისტემის ფუძეზე და თანრიგების რაოდენობაზე, რომელიც გამოყოფილია რიგისათვის. გამოთვლების სიზუსტე კი განისაზღვ-რება მანტისის თანრიგების რაოდენობით, ამიტომ ზოგჯერ სიზუს-ტის გაზრდისათვის იღებენ ორმაგი სიგრძის ფორმატს (წარმოდგე-ნის სიზუსტე ორჯერ იზრდება); რიგებისათვის გამოყოფილი თან-რიგების რაოდენობა კი უცვლელი რჩება და მაშასადამე, არ იცვ-ლება რიცხვების წარმოდგენის დიაპაზონი. მაგრამ ორმაგი ფორმა-ტის გამოყენება ზრდის ოპერაციის შესრულების დროს და მეხსიე-რების მოცულობას, სადაც ინახება საწყისი სიდიდეები.

მანტისის თანრიგების ფიქსირებული რაოდენობის დროს რი-ცხვების წარმოდგენა ნორმალიზებული სახით საშუალებას გვაძ-ლევს თანრიგთა ბადეში შევინახოთ მეტი ნიშნადი ციფრები, რაც ზრდის გამოთვლების სიზუსტეს და გაანგარიშების დროს აადვი-ლებს მოქმედებებს რიგებზე და მანტისებზე.

„ნორმალიზებული ნული“ მოდულით უმცირესი ნორმალი-ზებული რიცხვია, რომელიც ჩაიწერება უჯრედში. რიცხვს, რომელ-საც აქვს ნულოვანი მანტისა და დადებითი ნულის მქონე რიგი, ე. ი. უჯრედის ყველა თანრიგში გვაქვს ნულები, ეწოდება „ჭეშმარიტი ნული“.

1.7. რიცხვების კოდირება კომპიუტერში.

1.7.1. პირდაპირი კოდი.

ამოცანის ამოხსნისას ოპერაციების წარმოება გვიხდება როგორც დადებით, ასევე უარყოფით რიცხვებზეც.

გამოკლების ოპერაციას განვიხილავთ როგორც ალგებრულ შეკრებას,სადაც ერთ-ერთი შესაკრები უარყოფითია$x - y = x + (-y)$.

ცნობილია, რომ შეკრების ალგორითმი განსხვავდება გამოკლების ალგორითმისაგან. ორი რიცხვის შეკრების დროს ჯამის თანრიგებში, გადატანის ერთიანის დამატებით, შეიძლება მივიღოთ თვლის სისტემის მაქსიმალური $k - 1$ ციფრი (k -თვლის სისტემის ფუძეა) და გადატანის ერთიანი მეზობელ უფროს თანრიგში.

გამოკლების ოპერაციის დროს გამორიცხული არ არის მაკლების რომელიმე თანრიგში გვქონდეს უფრო დიდი ციფრი, ვიდრე საკლების იმავე თანრიგში და სესხების ოპერაციის ჩასატარებლად მოგვიხდეს რამდენიმე თანრიგით მარცხნივ წაწევა. გარდა ამისა, არითმეტიკული მოწყობილობის ძირითად ნაწილს - ამჯამავს ორი შესაკრებიდან, რომელს მივაწვდით პირველად, არა აქვს მნიშვნელობა; მოწყობილობას კი, რომელშიც უნდა ჩატარდეს გამოკლების ოპერაცია, ჯერ უნდა მივაწოდოთ საკლები, ხოლო შემდეგ - მაკლები; ამის გამო, ამჯამავი კონსტრუქციულად უფრო მარტივია და კომპაქტური, ვიდრე მოწყობილობა გამოკლების ოპერაციის ჩასატარებლად.

ოთხი არითმეტიკული ოპერაციიდან ყველაზე ადვილია შეკრების ოპერაცია. თუ დროს, რომლის განმავლობაში სრულდება ერთი შეკრების ოპერაცია, მივიღებთ ერთეულად, მაშინ დაახლოებით გამოკლების ოპერაციას ესაჭიროება დროის 1,5 ერთეული, გამრავლების ოპერაციას $2 \div 3$ ერთეულამდე, გაყოფის ოპრაციას კი $6 \div 9$ ერთეულამდე.

ზემოთ აღვნიშნეთ, რომ გამრავლების ოპერაცია დაიყვანება მარცხნივ ან მარჯნივ დაძრული სამრავლების შეკრების ოპერაციამდე; გაყოფის ოპერაცია კი წარმოადგენს გასაყოფიდან გამყოფის მიმდევრობითი გამოკლების ოპერაციას.

კომპიუტერში გამოკლების ოპერაციას ცვლიან სპეციალური კოდებით წარმოდგენილი რიცხვების შეკრების ოპერაციით. უარყოფითი რიცხვების წარმოდგენისათვის მანქანებში გამოიყენება სამი კოდი პ ი რ დ ა პ ი რ ი , შ ე ბ რ უ ნ ე ბ უ ლ ი და დ ა მ ა ტ ე ბ ი თ ი.

პირდაპირი კოდი გამოიყენება რიცხვების ჩასაწერად მეხსიერების მოწყობილობის უჯრედებში, გამრავლებისა და შეკრების ოპერაციებისათვის, განსაკუთრებით მაშინ, როდესაც შესაკრებებს აქვთ ერთნაირი ნიშანი - ხდება შესაკრებთა ნიშნის გაანალიზება და ჯამს დაეწერება იგივე ნიშანი, რაც აქვს შესაკრებებს.

როგორც აღვნიშნეთ, უჯრედში, რომელშიც ჩაწერილია რიცხვი, მარცხენა კიდურა თანრიგი დათმობილი აქვს რიცხვის ნიშანს. თუ რიცხვი დადებითია, ნიშნის თანრიგში ჩაიწერება „0“, ხოლო თუ რიცხვი უარყოფითია - $k-1$, ვინაიდან ძირითადად ოპერაციებს ვაწარმოებთ ორობით ან ორობით კოდირებულ რიცხვებზე, უარყოფითი რიცხვის გამოსახვის დროს ნიშნის თანრიგში ჩაიწერება „ 1“.

ნიშნის თანრიგში ჩაწერილი ციფრები შეკრების ოპერაციაში ისევე მონაწილეობენ, როგორც რიცხვის ციფრული თანრიგები.

კოდების ადვილად წასაკითხად, მათი წერილობითი გამოსახვის დროს, ნიშნის კოდს რიცხვითი ნაწილიდან გამოვყოფთ წერტილით, ხოლო მძიმის ადგილს მივუთითებთ შესაბამისი თანრიგის ზემოთ.

დადებითი რიცხვის სამივე კოდი ერთნაირად გამოისახება და უდრის თვით რიცხვს, ე. ი. თუ

$$x > 0, \qquad მაშინ \quad [x]_{პ} = [x]_{შ} = [x]_{დ}.$$

მ ა გ ა ლ ი თ ი . $x = 0{,}1101$, მაშინ

$$[0{,}1101]_პ = [0{,}1101]_შ = [0{,}1101]_დ = 0{,}1101 .$$

უარყოფითი რიცხვების სამივე კოდი ერთმანეთისაგან განსხვავებულია.

უარყოფითი წესიერი ორობითი $x = -0, x_1 x_2 \dots x_n$ წილადის პირდაპირი კოდის მისაღებად, ნიშნის თანრიგში უნდა ჩავწეროთ „1“, ხოლო ციფრული თანრიგები უნდა დავტოვოთ უცვლელი სახით:

$$[x]_პ = 1\text{;}x_1 x_2 \dots x_n.$$

მაგალითად, უარყოფითი წესიერი ორობითი $x = -0{,}1011$ წილადის პირდაპირი კოდია $[-0{,}1011]_პ = 1\text{;}1011$.

თუ პირდაპირი კოდის გამოსახულებაში ნიშნის თანრიგში ჩაწერილ ერთიანს ფორმალურად ჩავთლით რიცხვის მთელ ნაწილად, მაშინ პირდაპირი კოდი ჩაიწერება შემდეგი სახით:

$$[x]_პ = 1 + [x] = 1 - x.$$

პირდაპირ კოდში ნულს ორი გამოსახულება აქვს: „დადებითი ნული“ $[+0{,}00 \dots 0]_პ = 0\text{;}00 \dots 0$ და უარყოფითი ნული $[-0{,}00 \dots 0]_პ = 1\text{;}00 \dots 0$. (ცხადია, $+0 = -0 = 0$. დადებითი ციფრული ნაწილის შემცირებით, მივიღებთ +0-ს, ხოლო თუ ნულს ვუახლოვდებით უარყოფითი ციფრული ნაწილის შემცირებით, მივიღებთ -0 -ს).

$x > 0$, $x = 0$ და $x < 0$ შემთხვევებისათვის პირდაპირი კოდის მიღების წესები შეიძლება ასე გავაერთიანოთ:

$$[x]_პ = \begin{cases} x, & \text{თუ} \quad x \geq 0; \\ 1 - x, & \text{თუ} \quad x \leq 0. \end{cases}$$

წესიერი წილადების შემთხვევაში პირდაპირი კოდით შეიძლება წარმოვადგინოთ რიცხვები ინტერვალიდან

$$[-(1 - 10^{-n}) \div 1 - 10^{-n}].$$

1.7.2. შებრუნებული კოდი.

იმისათვის, რომ შევქმნათ k - ობითი სისტემის უარყოფითი რიცხვის შებრუნებული კოდი, ნიშნის თანრიგში უნდა ჩავწეროთ $k\ -1$ და ციფრული თანრიგები უნდა შევავსოთ $k\ -1$ - მდე.

რიცხვების შებრუნებული კოდების შეკრების დროს, თუ ნიშნის თანრიგიდან მივიღებთ გადატანის ერთიანს იგი უნდა დაემატოს უმცროს ციფრულ თანრიგს; ამ პროცესს ციკლური გადატანა ეწოდება.

ჩამოყალიბებული წესი მართებულია როგორც მთელი ისე წილადი რიცხვებისათვის.

თუ გვაქვს უარყოფითი წესიერი k- ობითი $x = -0,x_1x_2 \ldots x_n$ წილადი, მაშინ

$[x]_შ = k - 1.\bar{x}_1\bar{x}_2 \ldots \bar{x}_n$, სადაც $x_i = (k-1) - \bar{x}_i$, ე. ი. $\bar{x}_i$ არის x_i - ის დამატება $k\ -1$ - მდე და $x_i + \bar{x}_i = k - 1$.

უარყოფითი წილადების შებრუნებული კოდების შექმნისას ოპერანდების თანრიგთა რაოდენობა წინასწარ უნდა გათანაბრდეს ნაკლები თანრიგების მქონე ოპერანდისათვის მარჯვნიდან ნულების მიწერით.

თუ წილადის ბოლო ნიშნადი ციფრის მარჯვნივ გვრჩება ცარიელი თანრიგები, ისინი აუცილებლად უნდა შეივსოს ნულებით, რომლებიც მხედველობაშია მისაღები უარყოფითი წილადის შებრუნებული კოდის შექმნისას.

ავიღოთ ხუთთანრიგა უჯრედები, რომლებშიც მარცხნიდან პირველი თანრიგი განკუთვნილია წილადის ნიშნისათვის, დანარჩენი ოთხი თანრიგი კი წილადებისათვის.

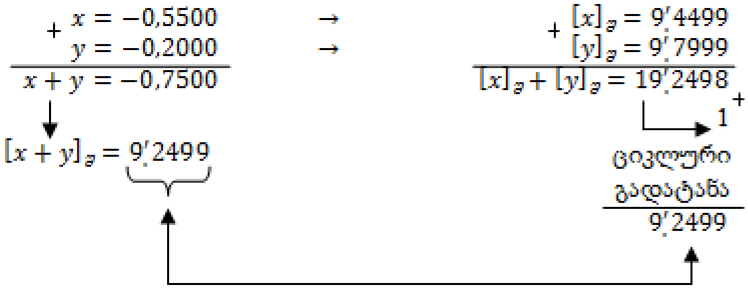
განვიხილოთ ორი ათობითი წილადის შეკრების მაგალითი.

1. $x = -0{,}55$ და $y = -0{,}2$.

იმისათვის, რომ გამოვრიცხოთ თანრიგთა ბადის გადავსება, როგორც შესაკრებები, ასევე მათი ჯამი მოდულით ნაკლები უნდა

იყოს 1-ზე. რიცხვებზე ოპერაციები რომ შევცვალოთ მათ კოდებზე ოპერაციებით, აუცილებელია შეკრების ყველა შემთხვევაში ჯ ა მ ი ს კ ო დ ი უ დ რ ი დ ე ს კ ო დ ე ბ ი ს ჯ ა მ ს .

თანრიგების გათანაბრებისა და უჯრედების ცარიელი თანრიგების ნულებით შევსების შემდეგ გვექნება:



მთელი რიცხვების პირდაპირი, შებრუნებული და დამატებითი კოდების შექმნის დროს მხედველობაშია მისაღები თანრიგთა ბადის სიდიდე (თანრიგების რაოდენობა) მთელი რიცხვები უნდა გათანაბრდეს მარჯვენა საზღვრით - უმცროსი თანრიგებით. ნიშნის თანრიგისა და რიცხვის ნიშნად ნაწილს შორის ცარიელი თანრიგები უნდა შეივსოს ნულებით, ე. ი. ნულები მთელ რიცხვს მიეწერება მარცხნიდან და ისინი აუცილებლად მხედველობაში უნდა მივიღოთ უარყოფითი რიცხვების შ ე ბ რ უ ნ ე ბ უ ლ ი და დ ა მ ა ტ ე ბ ი თ ი კ ო დ ე ბ ი ს შექმნისას.

მანქანაში ეს ცარიელი თანრიგები ავტომატურად აღირიცხება. თუ უჯრედში n თანრიგი გამოყოფილია მთელი რიცხვების ჩასაწერად, რათა არ მოხდეს თანრიგთა ბადის გადავსება, როგორც შესაკრებები, ასევე მათი ჯამი მოდულით არ უნდა აღემატებოდეს n - თანრიგა რიცხვს.

მანქანაში მაკლების მიწოდებისას, მისი ნიშანი ხელოვნურად შეიცვლება მოპირდაპირეზე და უკანასკნელი განსაზღვრავს მაკლების კოდს.

ჩავატაროთ გამოკლების ოპერაცია, $x - y = x + (-y)$.

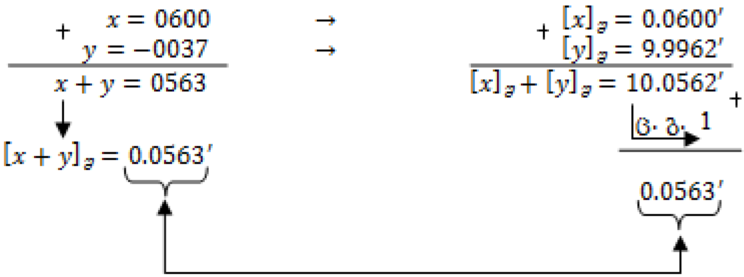
2. $x = 600$, $y = 37$, $n = 4$.

ზემოთ აღვნიშნეთ, რომ დადებითი რიცხვის შებრუნებული კოდი უდრის თვით რიცხვს.

ვინაიდან $n = 4$, რომ არ მივიღოთ თანრიგთა ბადის გადავსება, როგორც შესაკრებები, ასევე მათი ჯამი არ უნდა აღემატებოდეს ოთხთანრიგა რიცხვს.

ცარიელ თანრიგებში ნულების ჩაწერის და მაკლების ნიშნის შეცვლის შემდეგ ოპერანდები მიიღებენ შემდეგ სახეს:

$$x = 600, \quad y = -0037.$$



განხილულ მაგალითებში უარყოფითი რიცხვების შებრუნებული კოდების შექმნისას ნიშნის თანრიგში ჩავწერეთ $(k - 1) = 10 - 1 = 9$ და ციფრული თანრიგებიც შევავსეთ $k - 1$ - მდე (ზოგიერთ შემთხვევაში k - ბითი სისტემის უარყოფითი რიცხვის კოდის შექმნისას, ნიშნის თანრიგში წერენ „1“-ს ამის გამო, კოდების ჯამის ციფრული თანრიგების მოდული თვლის აღებული სისტემის k ფუძის ტოლია, ნიშნის თანრიგის მოდული კი 2-ის, ე. ი. ციფრულ თანრიგებში ოპერაციებს ვაწარმოებთ k-სისტემაში, ნიშნის თანრიგებში კი - ორობით სისტემაში.) კოდების ნიშნის თანრიგებში ჩაწერილი ციფრები შევკრიბეთ ისევე, როგორც მთლიანი რიცხვის თანრიგები.

იმისათვის, რომ გამოკლების ოპერაცია შევცვალოთ ოპერანდების შებრუნებული კოდების შეკრებით, ნიშნის თანრიგიდან მი-

ღებული გადატანის ერთიანი დავუმატეთ კოდების ჯამის უმცროს თანრიგს, ე. ი. მოვახდინეთ ციკლური გადატანა.

ვინაიდან ორობითი სისტემის ციფრებია 0 და 1, ამიტომ $k - 1 = 2 - 1 = 1$, და უარყობითი ორობითი რიცხვის შებრუნებული კოდის შექმნისას თანრიგები უნდა შევავსოთ ერთამდე. ავიღოთ წესიერი უარყოფითი ორობითი წილადი

$$x = -0, x_1 x_2 \dots x_n \,.$$

წესი: უარყოფითი ორობითი რიცხვის შებრუნებული კოდის შესაქმნელად, ნიშნის თანრიგში უნდა ჩავწეროთ „1“ და ციფრულ თანრიგებში ერთიანები უნდა შევცვალოთ ნულებით, ნულები კი ერთიანებით. ამ პროცესს ინვერსია ეწოდება. ფაქტიურად ციფრულ ნაწილს ვავსებთ $1 - 10^{-n}$- მდე.

მაშასადამე, თუ $x = -0, x_1 x_2 \dots x_n$, მაშინ $[x]_{\text{შ}} = 1{,}\bar{x}_1 \bar{x}_2 \dots \bar{x}_n$, სადაც $\bar{x}_i$ არის x_i-ის შევსება ერთამდე, ე. ი. თუ $x_i = 1$, მაშინ $\bar{x}_i = 0$, ხოლო თუ $x_i = 0$, მაშინ $\bar{x}_i = 1$, $i = 1 \div n$. ცხადია, რომ $x_i + \bar{x}_i = 1$.

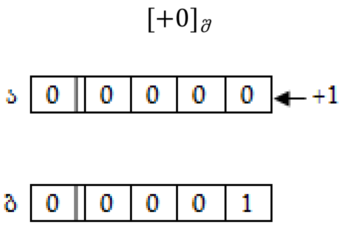
$$[x]_{\text{შ}} - x = 1{,}\bar{x}_1 \bar{x}_2 \dots \bar{x}_n - (0, x_1 x_2 \dots x_n) = 1{,}11 \dots 1 = 10 - 10^{-n}.$$

აქედან

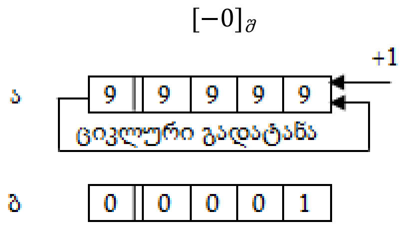
$$[x]_{\text{შ}} = 10 - 10^{-n} + x \qquad (1.14)$$

დადებითი ნულის შებრუნებული კოდი იქნება $[+0]_{\text{შ}} = [0{,}00 \dots 0]_{\text{შ}} = 0{,}00 \dots 0$, უარყოფითი ნულის შებრუნებული კოდი კი $[-0]_{\text{შ}} = [-0{,}00 \dots 0]_{\text{შ}} = 1{,}11 \dots 1$. ცხადია, იმავე შედეგს მივიღებთ (1.14) ტოლობიდანაც: $[-0]_{\text{შ}} = 10 - 10^{-n} - 0 = 1{,}11 \dots 1$. როგორც ვხედავთ, დადებითი და უარყოფითი ნულის შებრუნებული კოდი განსხვავებულად გამოისახება, მაგრამ ადვილად დავრწმუნდებით მათ იდენტურობაში, თუ $[+0]_{\text{შ}}$ და $[-0]_{\text{შ}}$ გამოსახულებების უმცროს თანრიგში დავუმატებთ 1-ს.

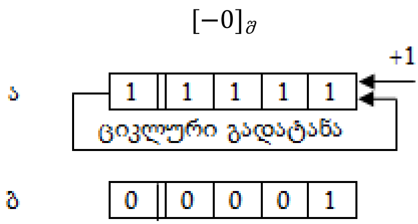
ვთქვათ, ორივე გამოსახულება ჩაწერილია უჯრედებში, რომელთა თანრიგების რაოდენობა 5 -ის ტოლია, სადაც ერთი თანრიგი (მარცხნიდან პირველი) დათმობილი აქვს ნიშანს, მაშინ მივიღებთ:



ნახ. 1.4



ნახ. 1.5



ნახ. 1.6

ნახ. 1.4-ის, 1.5-ის, და 1.6-ის შედარებით ვრწმუნდებით, რომ ოპერაციის შედეგი ერთნაირია და არ არის დამოკიდებული თვლის სისტემის ფუძეზე.

საბოლოოდ $x > 0$, $x = 0$ და $x < 0$, როდესაც მოცემული გვაქვს წესიერი წილადადები $|x| < 1$, შებრუნებული კოდის ფორმულები შეიძლება გამოვსახოთ შემდეგნაირად:

$$[x]_{შ} = \begin{cases} x, & \text{თუ} \quad x \geq 0; \\ 10 - 10^{-n} + x, & \text{თუ} \quad x \leq 0. \end{cases} \tag{1.15}$$

მთელი რიცხვებისათვის გვექნება:

$$[x]_{შ} = \begin{cases} x\,, & \text{თუ} \quad x \geq 0; \\ 10^{n+1} - 1 + x, & \text{თუ} \quad x \leq 0, \end{cases} \tag{1.16}$$

სადაც n არის უჯრედის ციფრული თანრიგების რაოდენობა.

ჩავწეროთ $x = -587_{10}$ და $y = -1011_2$- ის შებრუნებული კოდები უჯრედებში, რომლებშიც n = 4.

$$[x]_{\text{შ}} = 10^5 - 1 - 587 = \begin{array}{r} -\,9.9999' \\ 587 \\ \hline 9.9412' \end{array}$$

, ე. ი. $[-0587]_{\text{შ}} = 9.9412'$.

$$[y]_{\text{შ}} = 10^5 - 1 - 1011 = \begin{array}{r} -\,1.1111' \\ 1011 \\ \hline 1.0100' \end{array}$$

, მაშასადამე $[-1011]_{\text{შ}} = 1.0100'$.

არაწესიერი წილადებისათვის გვექნება:

$$[x]_{\text{შ}} = \begin{cases} x, & \text{თუ } x \geq 0; \\ 10^{n+1} - 10^{-r} + x, & \text{თუ } x \leq 0, \end{cases} \qquad (1.17)$$

სადაც n არის უჯრედში რიცხვის მთელი ნაწილისთვის გამოყოფილი თანრიგების რაოდენობა, ხოლო r - თანრიგების რაოდენობა, რომელიც გამოყოფილია უჯრედში რიცხვის წილადი ნაწილისათვის.

ჩავწეროთ $x = -F5C,34_{16}$ და $y = -101,11_2$- ის შებრუნებული კოდები უჯრედებში, სადაც $n = 4$, და $r = 3$. (1.17) -ის თანახმად გვექნება:

$$[x]_{\text{შ}} = 10^5 - 10^{-3} - F5C,34 = 100000 - 0{,}001 - F5C,34 =$$

$$= \begin{array}{r} F.FFFF'FFF \\ F5C,34 \\ \hline F.F0A3'CBF \end{array},$$

ე. ი. $[-0F5C,340]_{\text{შ}} = F.F0A3'CBF$;

$$[y]_{\text{შ}} = 10^5 - 10^{-3} - 101,11 = \begin{array}{r} -\,1.1111'111 \\ 101{,}11 \\ \hline 1.1010'001 \end{array},$$

ე. ი. $[-0101{,}110]_{\text{შ}} = 1.1010'001$.

განვიხილოთ ორი წესიერი წილადის ალგებრული შეკრება ფიქსირებულ მძიმიან მანქანაში შებრუნებულ კოდში.

იმისათვის, რომ გამოვრიცხოთ თანრიგთა ბადის გადავსება, ვგულისხმობთ, რომ როგორც შესაკრებები, ისე მათი ჯამი მოდულით ნაკლებია ერთზე, ე. ი.

$$[x] < 1,\ \ [y] < 1,\ \ [x+y] < 1, \qquad (1.18)$$

ხოლო $0 \le [A]_{\partial} < 10$, სადაც A შეიძლება იყოს როგორც ნებისმიერი შესაკრები, ისე შესაკრებთა ჯამიც.

(1.18) გამოსახულების პირველ ორ უტოლობას ყოველთვის აქვს ადგილი, როდესაც ოპერაციებს ვაწარმოებთ წესიერ წილადებზე როგორც ფიქსირებულ, ისე მცურავ მძიმიან რეჟიმებში, მესამე უტოლობა კი აუცილებელია იმისათვის, რომ არ მოხდეს თანრიგთა ბადის გადავსება და ოპერაციის შედეგი ჩაიწეროს მანქანის მეხსიერების უჯრედში.

რიცხვებზე ოპერაციები რომ შევცვალოთ ამავე რიცხვების შებრუნებულ კოდებზე ოპერაციებით, აუცილებელია <u>შეკრების ყველა შემთხვევაში კოდების ჯამი უდრიდეს ჯამის კოდს.</u>

$$[x]_{\partial} + [y]_{\partial} = [x+y]_{\partial} \qquad (1.19)$$

ორი რიცხვის შეკრების დროს შეიძლება ადგილი ჰქონდეს შემდეგ ოთხ ძირითად შემთხვევას:

1. ორივე შესაკრები დადებითია;
2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია;
3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია;
4. ორივე შესაკრები უარყოფითია.

საწყის პირობას წარმოადგენენ (1.18) და (1.19) გამოსახულებები. მხედველობაში უნდა მივიღოთ, რომ კოდის ნიშნის თანრიგში ჩაწერილი ციფრები ოპერაციაში მონაწილეობენ, როგორც რიცხვის მთელი ნაწილები, და გამოკლება არის ალგებრული შეკრება, სადაც ერთ-ერთი შესაკრები უარყოფითია $x - y = x + (-y)$.

1. ორივე შესაკრები დადებითია.

$$0 < x < 1,\ \ 0 < y < 1 \ \ \textit{და} \ \ 0 < x+y < 1.$$

ამ სამი უტოლობისა და (1.15) ფორმულების საფუძველზე შეგვიძლია დავწეროთ, რომ $[x]_{\partial} = x$, $[y]_{\partial} = y$ *და* $[x+y]_{\partial} = x + y$. კოდების ჯამი $[x]_{\partial} + [y]_{\partial} = x + y$. უკანასკნელი ორი გამოსახულების

მარჯვენა მხარეები ტოლია, ამიტომ ტოლი იქნება მარცხენა მხარეებიც $[x]_შ + [y]_შ = [x + y]_შ$.

2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია.

$$0 < x < 1, \qquad -1 < y < 0 \;\; და \;\; 0 < x + y < 1.$$

მოყვანილი უტოლობების და (1.15)-ის საფუძველზე გვექნება: $[x]_შ = x$, $[y]_შ = 10 - 10^{-n} + y$. ვინაიდან ჯამი დადებითია ჯამის კოდი $[x + y]_შ = x + y$. კოდების ჯამი კი

$$[x]_შ + [y]_შ = 10 - 10^{-n} + x + y.$$

უკანასკნელი გამოსახულების მარჯვენა მხარე არის შემდეგი სახის რიცხვი : $1{,}11\ldots1 + 0\ldots = 10{,}\ldots$ და ნ ი შ ნ ი ს თ ა ნ რ ი გ ი დ ა ნ მ ი ვ ი ღ ე ბ თ გ ა დ ა ტ ა ნ ი ს ე რ თ ი ა ნ ს.

კოდების ჯამი რომ მივიღოთ ჯამის კოდის ტოლი, აუცილებელია კოდების ჯამის მარჯვენა მხარეს გამოვაკლოთ 10 და დავუმატოთ 10^{-n}, რასაც ვახორციელებთ კოდების ჯამის ნიშნის თანრიგიდან მიღებული გადატანის ერთიანის დამატებით უმცროსი ციფრული თანრიგისათვის, რასაც ციკლური გადატანა ეწოდება.

საბოოლოდ გვექნება $[x]_შ + [y]_შ = [x + y]_შ$.

3. ერთი შესაკრები უარყოფითია, და ჯამიც უარყოფითია.

$$-1 < x < 0, \qquad 0 < y < 1 \;\; და \;\; -1 < x + y < 0.$$

ამ უტოლობების და (2.13) ფორმულების საფუძველზე გვექნება: $[x]_შ = 10 - 10^{-n} + x$, $[y]_შ = y$. ვინაიდან ჯამი უარყოფითია, ჯამის კოდი $[x + y]_შ = 10 - 10^{-n} + x + y$. კოდების ჯამი ტოლი იქნება $[x]_შ + [y]_შ = 10 - 10^{-n} + x + y$. უკანასკნელი ორი გამოსახულების მარჯვენა მხარეები ტოლია, ამიტომ ტოლი იქნება მარცხენა მხარეებიც ე. ი. $[x]_შ + [y]_შ = [x + y]_შ$.

კოდების ჯამის ნიშნის თანრიგიდან არ მივიღებთ გადატანის ერთიანს, ვინაიდან მისი მარჯვენა მხარე წარმოადგენს შემდეგი სა-

ხის რიცხვს $10-10^{-n}+x+y=1{,}11\dots1-0,\dots=1,\dots$ და ცხადია, ადგილი არ ექნება ციკლურ გადატანას.

4. ორივე შესაკრები უარყოფითია.

$$-1<x<0, \qquad -1<y<0 \ \text{ და } \ -1<x+y<0.$$

ამ უტოლობების და (1.15) ფორმულების საფუძველზე გვექნება:

$[x]_შ=10-10^{-n}+x,\ [y]_შ=10-10^{-n}+y$ და

$[x+y]_შ\ =10-10^{-n}+x+y.$ კოდების ჯამი $[x]_შ+[y]_შ=$
$=10-10^{-n}+x+10-10^{-n}+y=10-10^{-n}+[10-10^{-n}+(x+y)].$

კოდების ჯამი რომ უდრიდეს ჯამის კოდს, კოდების ჯამის მარჯვენა მხარეს უნდა გამოვაკლოთ 10 და დავუმატოთ 10^{-n}. კვადრატულ ფრჩხილებში მოთავსებული სიდიდე ნაკლებია 10-ზე, ვინაიდან $(x+y)<0$, ამიტომ კოდების ჯამი ტოლი იქნება
$1{,}11\dots1+1,\dots=11,\dots$ და ნიშნის თანრიგიდან მივიღებთ გადატანის ერთიანს, რომელსაც დავუმატებთ უმცროს ციფრულ თანრიგს, ე. ი. ჩ ა ვ ა ტ ა რ ე ბ თ ც ი კ ლ უ რ გ ა დ ა ტ ა ნ ა ს .

საბოლოოდ გვექნება $[x]_შ+[y]_შ=[x+y]_შ$.

ვინაიდან დადებითი და უარყოფითი ნულის შებრუნებული კოდები განსხვავებულად გამოისახება, ამიტომ დამატებით უნდა განვიხილოთ სამი კერძო შემთხვევა:

1. ორივე შესაკრები ნულია;
2. შესაკრებები განსხვავებულია ნულისაგან, მაგრამ მათი ჯამი ნულის ტოლია;
3. ერთ-ერთი შესაკრები ნულია:

1. ორივე შესაკრები ნულია.

$$x=0 \ \text{ და } \ y=0.$$

ადვილი შესამჩნევია, რომ $[+0]_შ+[+0]_შ=[+0]_შ$,
$[-0]_შ+[-0]_შ=[-0]_შ$. ვაჩვენოთ, რომ $[-0]_შ+[-0]_შ=[-0]_შ$.
მართლაც,

$$
\begin{array}{r}
+\begin{array}{r} [-0]_{\text{შ}} = 1'111 \ldots 11 \\ [-0]_{\text{შ}} = 1'111 \ldots 11 \end{array} \\
\hline
[-0]_{\text{შ}} + [-0]_{\text{შ}} = 11'111 \ldots 10 \\
\xrightarrow[\text{ც.გ.}]{} \; +1 \\
\hline
1'111 \ldots 11
\end{array}
$$

, ე. ი. $[-0]_{\text{შ}} + [-0]_{\text{შ}} = [-0]_{\text{შ}}$.

2. შესაკრებები განსხვავებულია ნულისაგან, მაგრამ მათი ჯამი ნულის ტოლია.

შესაკრებები აბსოლუტური სიდიდით ტოლია, მაგრამ ნიშნით მოპირდაპირე, ამიტომ თუ x-ით ავღნიშნავთ ერთ შესაკრებს, მეორე იქნება $y = -x$. $0 < x < 1$, $-1 < -x < 0$, $y = -x$, მაშინ

$$[x]_{\text{შ}} = x, \text{ ხოლო } [-x]_{\text{შ}} = 10 - 10^{-n} + (-x),$$

$$[x]_{\text{შ}} + [-x]_{\text{შ}} = x + 10 - 10^{-n} - x = [-0]_{\text{შ}},$$

ე. ი. $[x]_{\text{შ}} + [-x]_{\text{შ}} = [-0]_{\text{შ}}$.

3. ერთი შესაკრები ნულის ტოლია, ხოლო მეორე განსხვავებულია ნულისგან.

ვთქვათ, $x \neq 0$, $y = 0$, მაშინ $0 < [x]_{\text{შ}} < 1'11 \ldots 1$, დავრწმუნდეთ, რომ ამ შემთხვევაშიც კოდების ჯამი უდრის ჯამის კოდს. როგორც ზემოთ აღვნიშნეთ $+0 = -0 = 0$, ამიტომ
$x + (+0) = x + (-0) = x + 0 = x$, ე. ი. უნდა მივიღოთ, რომ

$$[x]_{\text{შ}} + [0]_{\text{შ}} = [x]_{\text{შ}}.$$

თუ შესაკრები $y = +0$, ცხადია $[y]_{\text{შ}} = [+0]_{\text{შ}} = 0$ და $[x]_{\text{შ}} + [+0]_{\text{შ}} = [x]_{\text{შ}}$, ხოლო თუ $y = -0$, მაშინ $[y]_{\text{შ}} = [-0]_{\text{შ}} = 10 - 10^{-n}$. ასეთ დროს უნდა განვიხილოთ ორი შემთხვევა: ა) $0 < X < 1$ და ბ) $-1 < X < 0$.

ა) $0 < X < 1$ და $y = -0$.

ვინაიდან $x + (-0) = x$, ჯამის კოდი $[x]_{\text{შ}} = x$. კოდების ჯამი კი $[x]_{\text{შ}} + [-0]_{\text{შ}} = x + 10 - 10^{-n}$. უკანასკნელი გამოსახულების მარჯვენა მხარე წარმოადგენს შემდეგი სახის რიცხვს $0, \ldots + 1{,}11 \ldots =$

$= 10, \ldots,$ ე. ი. ნ ი შ ნ ი ს თ ა ნ რ ი გ ი დ ა ნ ვ ი ღ ე ბ თ გ ა დ ა ტ ა-ნ ი ს ერთიანს. კოდების ჯამის მარჯვენა მხარეს უნდა გამოვაკლოთ 10 და დავუმატოთ 10^{-n}. რასაც ვახორციელებთ ნიშნის თანრიგიდან მიღებული გადატანის ერთიანის უმცროსი ციფრული თანრიგისა-თვის დამატებით, ანუ ციკლური გადატანით და საბოლოოდ მივი-ღებთ $[x]_შ + [-0]_შ = [x]_შ$.

ბ) $-1 < X < 0$ და $y = -0$.

$x + (-0) = x$, ჯამის კოდი $[x]_შ = 10 - 10^{-n} + x$. კოდების ჯამი $[x]_შ + [-0]_შ = 10 - 10^{-n} + x + 10 - 10^{-n}$. უკანასკნელი გამოსა-ხულების მარჯვენა მხარე წარმოადგენს შემდეგი სახის რიცხვს: $1, \ldots + 1{,}11\ldots1 = 11, \ldots,$ ე. ი. ნ ი შ ნ ის თ ა ნ რ ი გ ი დ ა ნ ვ ი-ღ ე ბ თ გ ა დ ა ტ ა ნ ი ს ე რ თ ი ა ნ ს, რომელსაც ვუმატებთ უმ-ცროს ციფრულ თანრიგს, ანუ ვახორციელებთ ციკლურ გადატანას, რაც ტოლფასია კოდების ჯამის მარჯვენა მხარისათვის 10 - ის გა-მოკლების და 10^{-n} - ის დამატების.

საბოლოოდ მივიღებთ, რომ კოდების ჯამი უდრის ჯამის კოდს $[x]_შ + [-0]_შ = [x]_შ$.

ა ლ გ ე ბ რ უ ლ ი შ ე კ რ ე ბ ი ს მ ა გ ა ლ ი თ ე ბ ი
ფიქსირებულ მძიმიან მანქანაში, როდესაც გამოყენებულია შებრუნებული კოდი.

წესიერი ორობითი წილადების ალგებრული შეკრება.

თუ ერთ-ერთი შესაკრები უარყოფითია, ვთქვათ, გვაქვს $[x]_შ$ და $[y]_შ$, მაშინ ამჯამავის გამოსავალზე მივიღებთ ჯამის კოდს $[x - y]_შ$. ვგულისხმობთ, რომ როგორც შესაკრებების, ისე მათი ჯა-მის აბსოლუტური მნიშვნელობა ნაკლებია ერთზე, ე. ი. $|x| < 1$, $|y| < 1$ *და* $|x + y| < 1$, ხოლო $0 \leq [A]_შ < 10$, სადაც A შეიძლება იყოს როგორც ნებისმიერი შესაკრები, ისე შესაკრებების ჯამიც.

თუ ერთ-ერთი ოპერანდი შეიცავს თანრიგების ნაკლებ რაოდენობას, მაშინ იგი გავუთანაბროთ მეორე ოპერანდს ნულების მიწერით და საერთოდ უჯრედში დარჩენილი ცარიელი თანრიგებიც უნდა შეივსოს ნულებით.

ქვემოთ განხილულ მაგალითებში, როგორც შებრუნებულ, ისე დამატებით კოდში, პირველ სვეტში მოცემული იქნება შესაკრებები, მათი ჯამი და ჯამის კოდი, მეორე სვეტში კი მოცემული იქნება შესაკრებთა კოდები და კოდების ჯამი.

ვინაიდან კოდების ჯამი უნდა უდრიდეს ჯამის კოდს, პირველი და მეორე სვეტის ბოლო სტრიქონში მიღებული შედეგი ერთნაირი უნდა იყოს.

მეორე სვეტში წარმოებს ოპერაცია, რომელსაც ასრულებს ამჯამავი მოწყობილობა.

წესიერ წილადებზე ოპერაციების დროს შებრუნებული კოდით შეიძლება წარმოვადგინოთ რიცხვები ინტერვალიდან

$$[-(1-10^{-n}) \div 1-10^{-n}].$$

1. ორივე შესაკრები დადებითია

$$0 < x < 1, \quad 0 < y < 1 \quad \text{და} \quad 0 < x + y < 1.$$

ვთქვათ, $x = 0{,}11001$, $y = 0{,}00101$

$$+\begin{array}{r} x = 0{,}11001 \\ y = 0{,}00101 \\ \hline x + y = 0{,}11110 \end{array} \qquad \begin{array}{c} \rightarrow \\ \rightarrow \\ \end{array} \qquad +\begin{array}{r} [x]_შ = 0'11001 \\ [y]_შ = 0'00101 \\ \hline [x]_შ + [y]_შ = \underbrace{0'11110} \end{array}$$

$$\downarrow$$

$$[x+y]_შ = \underbrace{0.0563'}$$

2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია.

$$0 < x < 1, \quad -1 < y < 0 \quad \text{და} \quad 0 < x + y < 1.$$

ვთქვათ, $x = 0{,}11001, \quad y = -\,0{,}00101$

$$
\begin{array}{rl}
+ & x = 0{,}11001 \\
  & y = -0{,}00101 \\ \hline
  & x + y = 0{,}10100
\end{array}
\quad
\begin{array}{c} \to \\ \to \end{array}
\quad
\begin{array}{rl}
+ & [x]_{\text{ვ}} = 0'11001 \\
  & [y]_{\text{ვ}} = 1'11010 \\ \hline
  & [x]_{\text{ვ}} + [y]_{\text{ვ}} = 10'10011 \\
+ & \text{ც. გ.}\ 1 \\ \hline
  & 0'10100
\end{array}
$$

$$[x + y]_{\text{ვ}} = 0'10100$$

3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია.

$-1 < x < 0, \quad 0 < y < 1$ და $-1 < x + y < 0$.

ვთქვათ, $x = -0{,}11001, \quad y = 0{,}00101$

$$
\begin{array}{rl}
+ & x = -0{,}11001 \\
  & y = \ \ 0{,}00101 \\ \hline
  & x + y = -0{,}10100
\end{array}
\quad
\begin{array}{c} \to \\ \to \end{array}
\quad
\begin{array}{rl}
+ & [x]_{\text{ვ}} = 1'00110 \\
  & [y]_{\text{ვ}} = 0'00101 \\ \hline
  & [x]_{\text{ვ}} + [y]_{\text{ვ}} = 1'01011
\end{array}
$$

$$[x + y]_{\text{ვ}} = 1'01011$$

4. ორივე შესაკრები უარყოფითია.

$-1 < x < 0, \quad -1 < y < 0$ და $-1 < x + y < 0$

ვთქვათ, $x = 0{,}11001, \quad y = 0{,}00101$

$$
\begin{array}{rl}
+ & x = -0{,}11001 \\
  & y = -0{,}00101 \\ \hline
  & x + y = -0{,}11110
\end{array}
\quad
\begin{array}{c} \to \\ \to \end{array}
\quad
\begin{array}{rl}
+ & [x]_{\text{ვ}} = 1'00110 \\
  & [y]_{\text{ვ}} = 1'00000 \\ \hline
  & [x]_{\text{ვ}} + [y]_{\text{ვ}} = 11'00000 \\
+ & \text{ც. გ.}\ 1 \\ \hline
  & 1'00001
\end{array}
$$

$$[x + y]_{\text{ვ}} = 1'00001$$

შებრუნებული კოდებისათვის დამახასიათებელი კერძო შემთხვევებიდან განვიხილოთ მე-2 და მე-3 შემთხვევები.

2) შესაკრებები განსხვავებულია ნულისაგან, მაგრამ მათი ჯამი ნულის ტოლია, ე. ი. შესაკრებები აბსოლუტური სიდიდით ტოლია, ნიშნით კი მოპირდაპირე:

$$0 < x < 1, \quad და \quad -1 < y = -x < 0.$$

ვთქვათ, $x = 0{,}11001$, $y = 0{,}00101$

$$\begin{array}{r} + \begin{array}{r} x = 0{,}11001 \\ y = -0{,}11001 \end{array} \\ \hline x + y = 0{,}00000 \end{array} \quad \begin{array}{c} \to \\ \to \end{array} \quad \begin{array}{r} + \begin{array}{r} [x]_შ = 0'11001 \\ [y]_შ = 1'00110 \end{array} \\ \hline [x]_შ + [y]_შ = \underbrace{1'11111} \end{array}$$

$$\downarrow$$

$$[-0] = \underbrace{1'11111}$$

3) ერთი შესაკრები ნულის ტოლია, ხოლო მეორე განსხვავებულია ნულისაგან.

ა) $0 < x < 1$ და $y = -0.$

$$x + y = x + (-0) = x.$$

$$\begin{array}{r} + \begin{array}{r} x = 0{,}0101 \\ y = -0{,}0000 \end{array} \\ \hline x + y = 0'0101 \end{array} \quad \begin{array}{c} \to \\ \to \end{array} \quad \begin{array}{r} + \begin{array}{r} [x]_შ = 0'0101 \\ [y]_შ = 1'1111 \end{array} \\ \hline [x]_შ + [y]_შ = 10'0100 \\ + \; \text{ც. გ.} \; 1 \\ \hline \underbrace{0'0101} \end{array}$$

$$\downarrow$$

$$[x + y]_შ = \underbrace{0'0101}$$

ბ) $-1 < x < 0$ და $y = -0$.

ვთქვათ, $x = -0{,}0111$, $y = -0{,}0000$

$$
\begin{array}{r}
x = -0{,}0111 \\
+\; y = -0{,}0000 \\
\hline
x + y = -0{,}0111 \\
\downarrow \\
[x+y]_{\text{გ}} = \underbrace{1'1000}
\end{array}
\qquad
\begin{array}{c}
\rightarrow \\
\rightarrow
\end{array}
\qquad
\begin{array}{r}
[x]_{\text{გ}} = 1'1000 \\
+\; [y]_{\text{გ}} = 1'1111 \\
\hline
[x]_{\text{გ}} + [y]_{\text{გ}} = 11'0111 \\
+\; \text{ც. გ.}\; 1 \\
\hline
\underbrace{1'1000}
\end{array}
$$

მთელი ათობითი რიცხვების ალგებრული შეკრება.

უჯრედში რიცხვების მთელი ნაწილისათვის გამოყოფილია ოთხი თანრიგი ($n = 4$) და ერთი თანრიგი ნიშნისათვის.

მთელი რიცხვები უნდა გათანაბრდეს უმცროსი თანრიგით.

$$|x| < 10^4, \quad |y| < 10^4, \quad |x+y| < 10^4,$$
$$x = 8907, \quad y = 856.$$

1. ორივე შესაკრები დადებითია.

$0 < x < 10^4$, $0 < y < 10^4$ და $0 < x + y < 10^4$

$$
\begin{array}{r}
x = 8907 \\
+\; y = 0856 \\
\hline
x + y = 9763 \\
\downarrow \\
[x+y]_{\text{გ}} = \underbrace{0.9763'}
\end{array}
\qquad
\begin{array}{c}
\rightarrow \\
\rightarrow
\end{array}
\qquad
\begin{array}{r}
[x]_{\text{გ}} = 0.8907' \\
+\; [y]_{\text{გ}} = 0.0856' \\
\hline
[x]_{\text{გ}} + [y]_{\text{გ}} = \underbrace{0.9763'}
\end{array}
$$

2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია.

$0 < x < 10^4, \quad -(10^4) < y < 0$ და $0 < x + y < 10^4.$

$$\begin{array}{r} + \begin{array}{r} x = 8907 \\ y = -0856 \end{array} \\ \hline x + y = 8051 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \\ \\ \end{array} \quad \begin{array}{r} + \begin{array}{r} [x]_გ = 0.8907' \\ [y]_გ = 9.9143' \end{array} \\ \hline [x]_გ + [y]_გ = 10.8050' \\ + \quad ც.\ გ.\ 1 \\ \hline 0.8051' \end{array}$$

$$[x+y]_გ = 0.8051'$$

3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია.

$-(10^4) < x < 0, \quad 0 < y < 10^4$ და $-(10^4) < x + y < 0$

$$\begin{array}{r} + \begin{array}{r} x = -8907 \\ y = 0856 \end{array} \\ \hline x + y = -8051 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \\ \\ \end{array} \quad \begin{array}{r} + \begin{array}{r} [x]_გ = 9.1092' \\ [y]_გ = 0.0856' \end{array} \\ \hline [x]_გ + [y]_გ = 9.1948' \end{array}$$

$$[x+y]_გ = 9.1948'$$

4. ორივე შესაკრებები უარყოფითია.

$-(10^4) < x < 0, \quad -(10^4) < y < 0$ და $-(10^4) < x + y < 0$

$$\begin{array}{r} + \begin{array}{r} x = -8907 \\ y = -0856 \end{array} \\ \hline x + y = -9763 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \\ \\ \end{array} \quad \begin{array}{r} + \begin{array}{r} [x]_გ = 9.1092' \\ [y]_გ = 9.9143' \end{array} \\ \hline [x]_გ + [y]_გ = 19.0235' \\ + \quad ც.\ გ.\ 1 \\ \hline 9.0236' \end{array}$$

$$[x+y]_გ = 9.0236'$$

წესიერი თექვსმეტობითი წილადების ალგებრული შეკრება.

როგორც აღვნიშნეთ, წილადების ბოლო ნიშნადი ციფრის მარჯვნივ ცარიელი თანრიგები უნდა შეივსოს ნულებით.

$$x = 0{,}89FE, \quad y = 0{,}2DC.$$

1. ორივე შესაკრებები დადებითია.

$$0 < x < 1, \quad 0 < y < 1 \quad \text{და} \quad 0 < x + y < 1$$

$$\begin{array}{rcr} + \begin{array}{r} x = 0{,}89FE \\ y = 0{,}2DC0 \end{array} & \begin{array}{c} \rightarrow \\ \rightarrow \end{array} & + \begin{array}{r} [x]_{\text{გ}} = 0'89FE \\ [y]_{\text{გ}} = 0'2DC0 \end{array} \\ \hline x + y = 0{,}B7BE & & [x]_{\text{გ}} + [y]_{\text{გ}} = \underbrace{0'B7BE} \\ \downarrow & & \\ [x + y]_{\text{გ}} = \underbrace{0'E7BE} & & \end{array}$$

2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია.

$$0 < x < 1, \quad -1 < y < 0 \quad \text{და} \quad 0 < x + y < 1$$

$$\begin{array}{rcr} + \begin{array}{r} x = 0{,}89FE \\ y = -0{,}2DC0 \end{array} & \begin{array}{c} \rightarrow \\ \rightarrow \end{array} & + \begin{array}{r} [x]_{\text{გ}} = 0'89FE \\ [y]_{\text{გ}} = F'D23F \end{array} \\ \hline x + y = 0{,}5C3E & & [x]_{\text{გ}} + [y]_{\text{გ}} = 10'5C3D \\ \downarrow & & + \quad \text{ც. გ.} \quad 1 \\ [x + y]_{\text{გ}} = \underbrace{0'5C3E} & & \underbrace{0'5C3E} \end{array}$$

3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია.

$$-1 < x < 0, \quad -1 < y < 0 \quad \text{და} \quad -1 < x + y < 0$$

$$
\begin{array}{rl}
+\begin{array}{r} x = -0{,}89FE \\ y = \ \ 0{,}2DC0 \end{array} & \begin{array}{c} \to \\ \to \end{array} \quad +\begin{array}{r} [x]_{გ} = F'7601 \\ [y]_{გ} = 0'2DC0 \end{array} \\
\hline
x + y = -0{,}5C3E & \qquad [x]_{გ} + [y]_{გ} = \underbrace{F'A3C1} \\
\downarrow & \\
[x+y]_{გ} = \underbrace{F'A3C1} &
\end{array}
$$

4. ორივე შესაკრები უარყოფითია.

$$-1 < x < 0, \quad -1 < y < 0 \quad და \quad -1 < x + y < 0$$

$$
\begin{array}{rl}
+\begin{array}{r} x = -0{,}89FE \\ y = -0{,}2DC0 \end{array} & \begin{array}{c} \to \\ \to \end{array} \quad +\begin{array}{r} [x]_{გ} = F'7601 \\ [y]_{გ} = F'D23F \end{array} \\
\hline
x + y = -0{,}B7BE & \qquad [x]_{გ} + [y]_{გ} = 1F'4840 \\
\downarrow & \qquad\qquad + \quad ც.გ. \ 1 \\
[x+y]_{გ} = \underbrace{F'4841} & \qquad\qquad \underbrace{F'4841}
\end{array}
$$

როდესაც შებრუნებული კოდის ნიშნის თანრიგში ჩაწერილია ნულისაგან განსხვავებული რიცხვი, ეს იმის მაჩვენებელია, რომ იგი წარმოადგენს უარყოფითი რიცხვის კოდს და მიუთითებს თვლის სისტემაზე, რომელშიც აღებულია ეს რიცხვი.

წესიერი უარყოფითი წილადის შებრუნებული კოდიდან რიცხვის აღდგენისათვის ნიშნის თანრიგში ჩაწერილი $k-1$ ციფრი უნდა შევცვალოთ (–„0“) მინუს ნული მთელით, ციფრულ თანრიგებში კი ვატარებთ ზუსტად ისეთივე ოპერაციებს, როგორც უარყოფითი რიცხვის შებრუნებული კოდის შექმნის დროს, ე. ი. ციფრულ თანრიგებს ვავსებთ $k-1$ -მდე ორობითი რიცხვების შემთხვევაში 0-ებს

ვცვლით ერთიანებით, ერთიანებს კი ნულებით, ანუ ვატარებთ ინ-ვერსიას.

წესიერი წილადების შემთხვევაში იმავე შედეგს მივიღებთ (1.15) ტოლობიდან:

$$x = -10 + 10^{-n} + [x]_{შ} = -[10 - 10^{-n}] + [x]_{შ} = -\{10 - 10^{-n} - [x]_{შ}\} \quad (1.20)$$

შებრუნებული კოდიდან პირდაპირი კოდის მისაღებად ნიშ-ნის თანრიგს ვტოვებთ უცვლელად, ციფრულ თანრიგებს კი ვცვლით ისევე, როგორც რიცხვის აღდგენის დროს.

მთელი რიცხვებისათვის (1.16) ტოლობიდან გვექნება:

$$x = -[10^{n+1} - 1] + [x]_{შ} = -\{10^{n+1} - 1 - x_{შ}\} \quad (1.21)$$

არაწესიერი წილადებისათვის კი რიცხვის მნიშვნელობას მივი-ღებთ (1.17) ტოლობიდან:

$$x = -[10^{n+1} - 10^{-r}] + [x]_{შ} = -\{10^{n+1} - 10^{-r} - [x]_{შ}\} \quad (1.22)$$

(1.21) და (1.22) ტოლობებში n არის უჯრედში რიცხვის მთელი ნაწილისათვის გამოყოფილი თანრიგების რაოდენობა, ხოლო (1.22) ტოლობაში r არის თანრიგების რაოდენობა გამოყოფილი რიცხვის წილადი ნაწილისათვის.

წესიერი წილადების აღდგენისათვის ზემოთ ჩამოყალიბებუ-ლი წესი მართებულია, როგორც მთელი რიცხვებისათვის, ისე არაწე-სიერი წილადებისათვის, იმ განსხვავებით, რომ ნიშნის თანრიგში ჩაწერილი ციფრი უნდა შევცვალოთ არა „ -0“ მთელით, არამედ მხოლოდ „ $-$ “ - ით.

მოცემული შებრუნებული კოდებიდან აღვადგინოთ რიცხვები და შევქმნათ პირდაპირი კოდები:

1. $[x]_{შ} = 1'\!.00101$

   $x = -0{,}11010_2$

   $[x]_{პ} = 1'\!.11010$

ნიშნის თანრიგის ციფრი გვიჩვენებს, რომ ეს არის ორობითი რიცხვის კოდი და (2.18) ტოლობიდან გვექნება:

$$x = -(1.11111 - 1\underset{.}{'}00101) = -0\underset{.}{'}11010.$$

2. $[x]_{გ} = 7\underset{.}{'}05021$

$x = -0{,}72756_8$

$[x]_{ვ} = 7\underset{.}{'}72756$

$k - 1 = 7, \quad k = 8$

$$x = -\begin{array}{r} (7\underset{.}{'}77777 \\ -7\underset{.}{'}05021) \\ \hline -0{,}72756_8 \end{array}$$

3. $[x]_{გ} = 9\underset{.}{'}01101$

$x = -0{,}98898_{10}$

$[x]_{ვ} = 9\underset{.}{'}98898$

$k - 1 = 9, \quad k = 10$

$$x = -\begin{array}{r} (9\underset{.}{'}99999 \\ -9\underset{.}{'}01101) \\ \hline -0{,}98898_{10} \end{array}$$

4. $[x]_{გ} = F\underset{.}{'}B12AD$

$x = -0{,}4ED52_{16}$

$[x]_{ვ} = F\underset{.}{'}4ED52$

$k - 1 = F, \quad k = 16$

$$x = -\begin{array}{r} (F\underset{.}{'}FFFFF \\ -F\underset{.}{'}B12AD) \\ \hline -0{,}4ED52_{10} \end{array}$$

5. $[x]_{გ} = 1.1011'$

$x = -0100 = -100_2$

$[x]_{ვ} = 1.0100'$

(1.21) ტოლობიდან გვექნება

$$x = -\begin{array}{r} (1.1111' \\ -1.101') \\ \hline -0100 = -100_2 \end{array}$$

6. $[x]_{გ} = 7.053'$

$x = -724_8$

$[x]_{ვ} = 7.724'$

$$k = 8, \quad x = -\{10^4 - 1 - [x]_{გ}\} =$$

$$= -\begin{array}{r} (7.777' \\ -7.053') \\ \hline -724_8 \end{array}$$

7. $[x]_შ = 9.863'47 \qquad k = 10, \qquad$ (1.22) ტოლობიდან გვექნება

$x = -136{,}52_{10} \qquad x = -\{10^{n+1} - 10^{-r} - [x]_შ\} =$

$[x]_შ = 9.136'52 \qquad = -(9.999'99$

$$\begin{array}{r} -9.863'47) \\ \hline -136{,}52_{10} \end{array}$$

8. $[x]_შ = F.A3'C41 \qquad k = 16, \qquad x = -(F.FF'FFF$

$x = -5C,3BE_{16}$

$[x]_შ = F.5C'3BE$

$$\begin{array}{r} -F.A3'C41 \\ \hline -5C'3BE_{16} \end{array}$$

1.7.3. დამატებითი კოდი.

როგორც აღვნიშნეთ, დადებითი რიცხვის დამატებითი კოდი უდრის თვით რიცხვს, ე. ი. ნიშნის თანრიგში ჩავწერთ „0“- ს, ციფრული თანრიგები კი უცვლელი დარჩება.

იმისათვის, რომ შევქმნათ k - ობითი სისტემის უარყოფითი რიცხვის დამატებითი კოდი, ნიშნის თანრიგში უნდა ჩავწეროთ „$k-1$“ ციფრული თანრიგები შევავსოთ $(k\ -1)$ -მდე და უმცროს ციფრულ თანრიგს დავუმატოთ „1“ ფაქტიურად უმცროსი თანრიგი უნდა შევავსოთ k -მდე. მაშასადამე, უარყოფითი რიცხვის დამატებითი კოდის შექმნისათვის უნდა შევქმნათ ამ რიცხვის შებრუნებული კოდი და უმცროს ციფრულ თანრიგს დავუმატოთ „1“.

k - ობითი სისტემის რიცხვის დამატებითი კოდის შესაქმნელად უფრო ადვილია შემდეგი ხერხით სარგებლობა: მარცხნიდან მარჯვნივ ნიშნის თანრიგის ჩათვლით ყველა ციფრი შევავსოთ $(k\ -1)$ -მდე, მარჯვენა კოდური ნულისაგან განსხვავებული ციფრი კი შევავსოთ k -მდე და ამ ციფრის მარჯვნივ ჩაწერილი ნულები უცვლელი სახით დავტოვოთ.

ჩამოყალიბებული წესები მართებულია როგორც მთელი, ისე წილადი რიცხვებისათვის.

რიცხვების დამატებით კოდების შეკრების დროს <u>ჯამის კოდი რომ უდრიდეს კოდების ჯამს, ნიშნის თანრიგიდან გადატანის ერთიანის მიღების შემთხვევაში, ეს ერთიანი უნდა გადაიგდოს.</u>

უარყოფითი ათობითი და თექვსმეტობითი წილადების შეკრების მაგალითები.

$$x = -0{,}5403_{10}, \qquad y = -0{,}25$$

$$\begin{array}{rcr} \; x = -0{,}5403 & \rightarrow & [x]_{დ} = 9'4597 \\ +\; y = -0{,}25 & \rightarrow & +\;[y]_{დ} = 9'75 \\ \hline x + y = -0{,}7903 & & [x]_{დ} + [y]_{დ} = 19'\underbrace{2097} \\ \downarrow & & \swarrow \\ [x+y]_{დ} = 9'\underbrace{2097} & & \text{გადაიგდება} \end{array}$$

$$\begin{array}{rcr} \; x = -0,AFDC & \rightarrow & [x]_{დ} = F'5024 \\ +\; y = -0,2BE & \rightarrow & +\;[y]_{დ} = F'D42 \\ \hline x + y = -0,DBBC & & [x]_{დ} + [y]_{დ} = 1F'\underbrace{2444} \\ \downarrow & & \swarrow \\ [x+y]_{დ} = F'\underbrace{2444} & & \text{გადაიგდება} \end{array}$$

$$[-0{,}25]_{დ} = 9'75, \qquad [-0{,}2500]_{დ} = \begin{array}{r} 9'7499 \\ +\qquad 1 \\ \hline 9'7500 \end{array}$$

განხილული მაგალითებიდან ჩანს, რომ წილადების დამატებითი კოდების შეკრების დროს <u>წილადების თანრიგთა გათანაბრება არ არის აუცილებელი.</u>

უარყოფითი მთელი რიცხვების დამატებითი კოდების შექმნის დროს მხედველობაში უნდა მივიღოთ უჯრედის თანრიგთა ბადის სიდიდე (თანრიგების რაოდენობა); ნიშნის თანრიგსა და რიცხვის უფროს თანრიგს შორის ცარიელი თანრიგები უნდა შეივსოს ნულებით, რომლებიც აუცილებლად მხედველობაში მიიღება დამატებითი კოდის შექმნისას.

თუ მეხსიერების უჯრედში ჩაწერილი ოპერანდებიდან I ოპერანდს უნდა გამოაკლდეს II ოპერანდი, რომელიც დადებითია, მაშინ ავტომატურად ხდება მისი ნიშნის შეცვლა და ამჯამავს გადაეცემა უარყოფითი რიცხვის კოდი. თუ II ოპერანდი უარყოფითია და უჯრედში იგი ჩაწერილია პირდაპირ კოდში, მას შეეცვლება ნიშანი და ამჯამავს გადაეცემა დადებითი რიცხვის კოდი, ხოლო თუ II ოპერანდი უჯრედში ჩაწერილია დამატებით (შებრუნებულ) კოდში, უარყოფითი რიცხვის აღდგენა ავტომატურად ხდება, მას ეცვლება ნიშანი და ისე გადაეცემა ამჯამავს.

ავიღოთ ხუთთანრიგა უჯრედი, სადაც ერთი თანრიგი დათმობილი აქვს ნიშანს.

I - ოპერანდი - $x = 500$ II - ოპერანდი - $y = 13$

ჩავატაროთ გამოკლების ოპერაცია $x - y = x + (-y)$.

$$
\begin{array}{r}
+\begin{array}{r} x = 0500 \\ y = -0013 \end{array} \\ \hline x + y = 0487
\end{array}
\quad \rightarrow \quad
\begin{array}{r}
+\begin{array}{r} [x]_{\text{დ}} = 0.0500' \\ [y]_{\text{დ}} = 9.9987' \end{array} \\ \hline [x]_{\text{დ}} + [y]_{\text{დ}} = 10'.\underbrace{0487}
\end{array}
$$

$$[x + y]_{\text{დ}} = \underbrace{0.0487'}$$

გადაიგდება

1. უარყოფითი ორობითი რიცხვის დამატებითი კოდის შესაქმნელად ნიშნის თანრიგში უნდა ჩავწეროთ „1“, ციფრული თანრიგები უნდა შევავსოთ ერთამდე, ე. ი უნდა ჩავატაროთ ინვერსია და უმცროს ციფრულ თანრიგს დავუმატოთ „1“, ფაქტიურად უმცროსი ციფრული თანრიგის შევსება ხდება 2-მდე. მაშასადამე, უარყოფითი რიცხვის დამატებითი კოდის შესაქმნელად უნდა შევქმნათ ამ რიცხვის შებრუნებული კოდი და უმცროს ციფრულ თანრიგს დაუმატოთ „1“.

2. უარყოფითი ორობითი წილადის დამატებითი კოდის შესაქმნელად მარცხნიდან მარჯვნივ, ნიშნის თანრიგის ჩათვლით, ნულები შეცვალოთ ერთიანებით ერთიანები კი ნულებით, უცვლელი სახით უნდა დავტოვოთ მარჯვენა კიდურა ერთიანი და მის შემდეგ დაწერილი ნულები.

ავიღოთ უარყოფითი წესიერი ორობითი წილადი

$$x = -0,x_1x_2 \dots x_n, \text{ მაშინ } [x]_{დ} = 1{,}\bar{x}_1x_2 \dots x_n + 0,\underbrace{00 \dots 01}_{n\text{-თანრიგა}},$$

სადაც, თუ $x_i = 1$, მაშინ $\bar{x}_i = 0$, ხოლო, თუ $x_i = 0$, მაშინ $\bar{x}_i = 1$, $i = 1 \div n$, ცხადია $x_i + \bar{x}_i = 1$.

$$[x]_{დ} - x = 1{,}\bar{x}_1\bar{x}_2 \dots \bar{x}_n - (0,x_1x_2 \dots x_n) + 0,00 \dots 01 =$$
$$= 1{,}11 \dots 1 + 0,00 \dots 01 = 10,$$

საიდანაც უარყოფითი წესიერი წილადის დამატებითი კოდი

$$x_{დ} = 10 + x.$$

დადებითი ნულის დამატებითი კოდი.

$[+0]_{დ} = [0,00 \dots 0]_{დ} = 0,00 \dots 0 = 0$, ასევე უარყოფითი ნულის დამატებითი კოდი $[-0]_{დ} = [-0,00 \dots 0]_{დ} = 0,00 \dots 0 =$

$$= 1'11 \dots 1 + 0,00 \dots 01 = \underset{\text{გადაიგდება}}{\underset{\downarrow}{1}}0 = 0$$

იგივე შედეგი მიიღება (2.19) ტოლობიდან:

$$[-0]_{დ} = 10 - 0 = 10 = 0$$

$$\downarrow$$

გადაიგდება

ვინაიდან ნიშნისათვის გამოყოფილი გვაქვს ერთი თანრიგი, ამ თანრიგში დარჩება „0“, ხოლო გადატანის 1-იანი გადაიგდება.

მაშასადამე, როგორც დადებითი, ისე უარყოფითი ნულის დამატებითი კოდი ნულია $[+0]_{დ} = [-0]_{დ} = 0$.

თუ უარყოფითი წესიერი ორობითი წილადის შებრუნებული კოდის შექმნის დროს ციფრული ნაწილის შევსება ხდება $1 - 10^{-n}$ - მდე, დამატებითი კოდის შემთხვევაში ციფრული ნაწილის შევსება ხდება 1 - მდე, ამიტომ წესიერი წილადების დროს დამატებითი კოდით შეიძლება წარმოვადგინოთ რიცხვები ინტერვალიდან

$$[-1 \div 1 - 10^{-n}].$$

საბოლოოდ $x > 0$, $x = 0$ და $x < 0$, შემთხვევებისათვის დამატებითი კოდის ფორმულები შეიძლება შემდეგნაირად გავაერთიანოთ:

$$[x]_{დ} = \begin{cases} x, & თუ \;\; x \geq 0; \\ 10 + x, & თუ \;\; x \leq 0. \end{cases} \tag{1.23}$$

მთელი რიცხვებისა და არაწესიერი წილადებისათვის დამატებითი კოდის ფორმულებს ექნება შემდეგი სახე:

$$[x]_{დ} = \begin{cases} x, & თუ \;\; x \geq 0; \\ 10^{n+1} + x, & \end{cases} \tag{1.24}$$

სადაც n არის უჯრედში რიცხვის მთელი ნაწილისათვის გამოყოფილი თანრიგების რაოდენობა.

ავიღოთ ხუთთანრიგა უჯრედი, სადაც $n = 4$ და ასეთ უჯრედში ჩავწეროთ $x = -987_{10}$ და $y = -EAD_{16}$.

(1.24) ტოლობიდან გვექნება:

$$[x]_{დ} = 10^5 - 0986 = \begin{array}{r} 100000 \\ -\;\;\;986 \\ \hline 9.9014' \end{array}$$

$$[y]_{დ} = 10^5 - 0EAD = \begin{array}{r} 100000 \\ - \quad \text{EAD} \\ \hline F.F153' \end{array}$$

$$[-0986]_{დ} = 9.9014' \qquad [-0EAD]_{დ} = F.F153'$$

თუ იმავე უჯრედებს დაუმატებთ ოთხ თანრიგს წილადი ნაწილისთვის, მაშინ არაწესიერი წილადების $x = -765{,}2410_{10}$ და $y = -101{,}11_2$ დამატებით კოდებს ექნება შემდეგი სახე:

$$[x]_{დ} = 10^5 - 765{,}2410 = \begin{array}{r} 100000{,}0000 \\ - \quad 765{,}241 \\ \hline 9.9234'7590 \end{array}$$

$$[-0765{,}2410]_{დ} = 9.9234'7590\,,$$

$$[y]_{დ} = 10^5 - 101{,}11 = \begin{array}{r} 100000{,}0000 \\ - \quad 101{,}11 \\ \hline 1.1010'0100 \end{array}$$

$$[-0101{,}1100]_{დ} = 1.1010'0100.$$

ორი წესიერი წილადის ალგებრული შეკრება დამატებით კოდში, ფიქსირებულ მძიმიან მანქანაში.

ორი წესიერი წილადის შეკრების შედეგად, რომ არ მივიღოთ თანრიგთა ბადის გადავსება, როგორც შესაკრებები, ასევე მათი ჯამი მოდულით ნაკლები უნდა იყოს ერთზე, ე. ი. $|x| < 1$, $|y| < 1$, და $|x + y| < 1$, და $0 \leq [A]_{დ} < 10$, სადაც A შეიძლება იყოს როგორც ნებისმიერი შესაკრები, ისე შესაკრებების ჯამიც.

რიცხვებზე ოპერაციები რომ შევცვალოთ ამავე რიცხვების დამატებით კოდებზე ოპერაციებით, აუცილებელია შეკრების ყველა შემთხვევაში კ ო დ ე ბ ი ს ჯ ა მ ი უ დ რ ი დ ე ს ჯ ა მ ი ს კ ო დ ს

$$[x]_{დ} + [y]_{დ} = [x + y]_{დ}.$$

ორი რიცხვის შეკრების დროს შეიძლება ადგილი ჰქონდეს შემდეგ ოთხ შემთხვევას:

1. ორივე შესაკრები დადებითია;
2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია;
3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია;
4. ორივე შესაკრები უარყოფითია.

1. ორივე შესაკრები დადებითია.

$$0 < x < 1, \ 0 < y < 1 \ \text{და} \ 0 < x + y < 1.$$

ამ უტოლობების და (1.23) ფორმულის საფუძველზე შეგვიძლია დავწეროთ $[x]_დ = x$, $[y]_დ = y$. ვინაიდან ჯამი დადებითია, ჯამის კოდი $[x+y]_დ = x+y$. კოდების ჯამი $[x]_დ + [y]_დ = x + y$. როგორც ვხედავთ, ორი უკანასკნელი გამოსახულების მარჯვენა მხარე ტოლია; ტოლი იქნება მარცხენა მხარეებიც, ამიტომ შეგვიძლია დავწეროთ $[x]_დ + [y]_დ = [x+y]_დ$.

2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია.

$$0 < x < 1, \ -1 < y < 0 \ \text{და} \ 0 < x + y < 1.$$

ამ უტოლობების და (1.23) ფორმულის საფუძველზე შეგვიძლია დავწეროთ $[x]_დ = x$, $[y]_დ = 10 + y$, ვინაიდან ჯამი დადებითია $[x+y]_დ = x+y$. კოდების ჯამი $[x+y]_დ = x + 10 + y = 10 + x + y$. რადგან $0 < x + y < 1$, უკანასკნელი გამოსახულების მარჯვენა მხარე წარმოადგენს შემდეგი სახის რიცხვს $10, \dots$; იმისათვის, რომ კოდების ჯამის მარჯვენა მხარეს უნდა გამოვაკლოთ 10, რასაც ვახორციელებთ კოდების ჯამის ნიშნის თანრიგიდან მიღებული გადატანის ერთიანის გადაგდებით, და საბოლოოდ მივიღებთ:

$$[x]_დ + [y]_დ = [x+y]_დ.$$

3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია.

$$-1 < x < 0, \ 0 < y < 1 \ \text{და} \ -1 < x + y < 0.$$

ამ უტოლობების და (1.23) ფორმულის საფუძველზე შეგვიძლია დავწეროთ $[x]_{ღ} = 10 + x$, $[y]_{ღ} = y$. **ვინაიდან ჯამი უარყოფითია** $[x + y]_{ღ} = 10 + x + y$, **კოდების ჯამი ტოლი იქნება** $[x]_{ღ} + [y]_{ღ} = 10 + x + y$; უკანასკნელი ორი გამოსახულების მარჯვენა მხარეები ტოლია და ნაკლებია 10-ზე, რადგან $x + y$ უარყოფითი სიდიდეა, რომელიც მოდულით ნაკლებია ერთზე, ამიტომ $10 + x + y$ წარმოადგენს შემდეგი სახის რიცხვს 1,... და კოდების ჯამის ნიშნის თანრიგიდან გადატანის ერთიანს არ მივიღებთ. საბოლოოდ გვექნება:

$$[x]_{ღ} + [y]_{ღ} = [x + y]_{ღ}.$$

4. ორივე შესაკრები უარყოფითია.

$$-1 < x < 0, \quad -1 < y < 0 \text{ და } -1 < x + y < 0.$$

ამ სამი უტოლობისა და (1.23) ფორმულის საფუძველზე შეგვიძლია დავწეროთ $[x]_{ღ} = 10 + x$, $[y]_{ღ} = 10 + y$ *და* $[x + y]_{ღ} =$
$= 10 + x + y$. კოდების ჯამი $[x]_{ღ} + [y]_{ღ} = 10 + x + 10 + y =$
$= 10 + (10 + x + y) = 100 + x + y$.

რადგან $-1 < x + y < 0$, ამიტომ უკანასკნელი გამოსახულების მარჯვენა მხარე $11 < 100 + x + y < 100$ წარმოადგენს შემდეგი სახის რიცხვს 11 ... (აქ 11 და 100 ორობითი რიცხვებია და შესაბამისად გამოსახავენ 3 და 4 ერთეულს).

იმისათვის, რომ კოდების ჯამი უდრიდეს ჯამის კოდს, საჭიროა კოდების მარჯვენა მხარეს გამოვაკლოთ 10, რასაც ვახორციელებთ ნიშნის თანრიგიდან მიღებული გადატანის ერთიანის გადაგდებით, და საბოლოოდ გვექნება:

$$[x]_{ღ} + [y]_{ღ} = [x + y]_{ღ}.$$

ალგებრული შეკრების მაგალითები ფიქსირებულმძიმიან მანქანაში, როდესაც გამოყენებულია დამატებითი კოდი.
წესიერი ორობითი წილადების შეკრება.

1. ორივე შესაკრები დადებითია.

$0 < x < 1, \quad 0 < y < 1$ და $0 < (x + y) < 1.$

ვთქვათ, $x = 0{,}1001$ და $y = 0{,}0110.$

$$
\begin{array}{rcl}
+\begin{array}{r} x = 0{,}1001 \\ y = 0{,}0110 \end{array} & \rightarrow & +\begin{array}{r} [x]_{\text{დ}} = 0'1001 \\ [y]_{\text{დ}} = 0'0110 \end{array} \\
\hline
x + y = 0{,}1111 & & [x]_{\text{დ}} + [y]_{\text{დ}} = \underbrace{0'1111} \\
\downarrow & & \\
[x+y]_{\text{დ}} = \underbrace{0'1111} & &
\end{array}
$$

2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია.

$0 < x < 1, \quad 0 < y < 1$ და $0 < x + y < 1.$

$$
\begin{array}{rcl}
+\begin{array}{r} x = 0{,}1001 \\ y = -0{,}0110 \end{array} & \rightarrow & +\begin{array}{r} [x]_{\text{დ}} = 0'1001 \\ [y]_{\text{დ}} = 1'1010 \end{array} \\
\hline
x + y = 0{,}0011 & & [x]_{\text{დ}} + [y]_{\text{დ}} = 1\underbrace{0'0011} \\
\downarrow & & \swarrow \\
[x+y]_{\text{დ}} = \underbrace{0'0011} & & \text{გადაიგდება}
\end{array}
$$

3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია.

$-1 < x < 0, \quad 0 < y < 1$ და $-1 < x + y < 0.$

$$
\begin{array}{rl}
+\begin{array}{r} x = -0{,}1001 \\ y = 0{,}0110 \end{array} & \begin{array}{c} \rightarrow \\ \rightarrow \end{array} \quad +\begin{array}{r} [x]_{\text{დ}} = 1'0111 \\ [y]_{\text{დ}} = 0'0110 \end{array} \\
\hline
x + y = -0{,}0011 & [x]_{\text{დ}} + [y]_{\text{დ}} = 1'1101
\end{array}
$$

$$
[x + y]_{\text{დ}} = 1'1101
$$

4. ორივე შესაკრები უარყოფითია.

$$
-1 < x < 0, \quad -1 < y < 0 \quad \text{და} \quad 0 < x + y < 1.
$$

$$
-1 < x < 0, \quad -1 < y < 0 \quad \text{და} \quad 0 < x + y < 1.
$$

$$
\begin{array}{rl}
+\begin{array}{r} x = -0{,}1001 \\ y = -0{,}0110 \end{array} & \begin{array}{c} \rightarrow \\ \rightarrow \end{array} \quad +\begin{array}{r} [x]_{\text{დ}} = 1'0111 \\ [y]_{\text{დ}} = 1'1010 \end{array} \\
\hline
x + y = -0{,}1111 & [x]_{\text{დ}} + [y]_{\text{დ}} = 11'0001
\end{array}
$$

გადაიგდება

$$
[x + y]_{\text{დ}} = 1'0001
$$

მთელი ათობითი რიცხვების ალგებრული შეკრების მაგალითები.

ავიღოთ ხუთთანრიგა უჯრედები, სადაც $n = 4$. მთელი რიცხვების გათანაბრება ხდება უმცროსი თანრიგებით და რიცხვის მარცხნივ ცარიელ თანრიგებში ნიშნის თანრიგამდე უნდა ჩაიწეროს ნულები.

1. ორივე შესაკრები დადებითია.

$$
0 < x < 10^4, \quad 0 < y < 10^4 \quad \text{და} \quad 0 < x + y < 10^4.
$$

$$\begin{array}{rr} + & x = 7859 \\ & y = 0064 \\ \hline & x+y = 7923 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \end{array} \quad \begin{array}{rr} + & [x]_{\text{ფ}} = 0.7859' \\ & [y]_{\text{ფ}} = 0.0064' \\ \hline & [x]_{\text{ფ}} + [y]_{\text{ფ}} = \underbrace{0.7923'} \end{array}$$

$$[x+y]_{\text{ფ}} = \underbrace{0.7923'}$$

2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია.

$0 < x < 10^4, \quad -(10^4) < y < 0$ და $0 < x + y < 10^4$.

$$\begin{array}{rr} + & x = 7859 \\ & y = -0064 \\ \hline & x+y = 7795 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \end{array} \quad \begin{array}{rr} + & [x]_{\text{ფ}} = 0.7859' \\ & [y]_{\text{ფ}} = 9.9936' \\ \hline & [x]_{\text{ფ}} + [y]_{\text{ფ}} = 1\underbrace{0.7795'} \end{array}$$

გადაიგდება

$$[x+y]_{\text{ფ}} = \underbrace{0.7795'}$$

3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია.

$-(10^4) < x < 0, \quad 0 < y < 10^4$ და $-(10^4) < x + y < 0$.

$$\begin{array}{rr} + & x = -7859 \\ & y = 0064 \\ \hline & x+y = -7795 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \end{array} \quad \begin{array}{rr} + & [x]_{\text{ფ}} = 9.2141' \\ & [y]_{\text{ფ}} = 0.0064' \\ \hline & [x]_{\text{ფ}} + [y]_{\text{ფ}} = \underbrace{9.2205'} \end{array}$$

$$[x+y]_{\text{ფ}} = \underbrace{9.2205'}$$

4. ორივე შესაკრები უარყოფითია.

$-(10^4) < x < 0, \quad -(10^4) < y < 0$ და $-(10^4) < x + y < 0$.

$$
\begin{array}{rcr}
+\begin{array}{r} x = -7859 \\ y = -0064 \end{array} & \begin{array}{c} \to \\ \to \end{array} & +\begin{array}{r} [x]_{\text{დ}} = 9.2141' \\ [y]_{\text{დ}} = 9.9936' \end{array} \\
\hline
x + y = -7923 & & [x]_{\text{დ}} + [y]_{\text{დ}} = 1\underbrace{9.2077'} \\
\downarrow & & \\
[x+y]_{\text{დ}} = \underbrace{9.2077'} & &
\end{array}
$$

გადაიგდება

წესიერი თექვსმეტობითი წილადების ალგებრული შეკრების მაგალითები.

როგორც აღვნიშნეთ, წილადების დამატებითი კოდების შეკრების დროს წილადების თანრიგების გათანაბრება არ არის საჭირო.

1. ორივე შესაკრები დადებითია.

$$0 < x < 1, \; 0 < y < 1 \; \text{და} \; 0 < x + y < 1.$$

$$
\begin{array}{rcr}
+\begin{array}{r} x = 0,A19C \\ y = 0,3F \end{array} & \begin{array}{c} \to \\ \to \end{array} & +\begin{array}{r} [x]_{\text{დ}} = 0',A19C \\ [y]_{\text{დ}} = 0',3F \end{array} \\
\hline
x + y = 0,E09C & & [x]_{\text{დ}} + [y]_{\text{დ}} = \underbrace{0',E09C} \\
\downarrow & & \\
[x+y]_{\text{დ}} = \underbrace{0',E09C} & &
\end{array}
$$

2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია.

$$0 < x < 1, \; -1 < y < 0 \; \text{და} \; 0 < x + y < 1.$$

$$
\begin{array}{rcl}
x = 0,A19C & \rightarrow & [x]_{დ} = 0'A19C \\
+\; y = -0,3F & \rightarrow & +\; [y]_{დ} = F'C1 \\
\hline
x + y = 0,629C & & [x]_{დ} + [y]_{დ} = 1\underbrace{0'629C} \\
\downarrow & & \\
[x+y]_{დ} = \underbrace{0'629C} & &
\end{array}
$$

გადაიგდება

3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია.

$-1 < x < 0, \quad 0 < y < 1$ და $-1 < x + y < 0.$

$$
\begin{array}{rcl}
x = -0,A19C & \rightarrow & [x]_{დ} = F'5E64 \\
+\; y = 0,3F & \rightarrow & +\; [y]_{დ} = 0'3F \\
\hline
x + y = -0,629C & & [x]_{დ} + [y]_{დ} = \underbrace{F'9D64} \\
\downarrow & & \\
[x+y]_{დ} = \underbrace{F'9D64} & &
\end{array}
$$

4. ორივე შესაკრები უარყოფითია.

$-1 < x < 0, \quad -1 < y < 0$ და $-1 < x + y < 0.$

$$
\begin{array}{rcl}
x = -0,A19C & \rightarrow & [x]_{დ} = F'5E64 \\
+\; y = -0,3F & \rightarrow & +\; [y]_{დ} = F'C1 \\
\hline
x + y = -0,E09C & & [x]_{დ} + [y]_{დ} = 1\underbrace{F'1F64} \\
\downarrow & & \\
[x+y]_{დ} = \underbrace{F'1F64} & &
\end{array}
$$

გადაიგდება

როდესაც დამატებითი კოდის ნიშნის თანრიგში ჩაწერილია ნულისაგან განსხვავებული k – 1 რიცხვი, ეს იმის მაჩვენებელია, რომ იგი წარმოადგენს უარყოფითი რიცხვის კოდს და მიუთითებს თვლის სისტემაზე, რომელსაც მიეკუთვნება ეს რიცხვი.

1. წესიერი უარყოფითი წილადის დამატებითი კოდიდან რიცხვის აღდგენისათვის ნიშნის თანრიგში ჩაწერილი k – 1 ციფრი უნდა შევცვალოთ („ – 0“) მინუს ნული მთელით, ციფრულ თანრიგებში კი ვატარებთ ზუსტად ისეთივე ოპერაციებს, როგორც უარყოფითი რიცხვის დამატებითი კოდის შექმნის დროს; სახელდობრ, ციფრულ თანრიგებში მარცხნიდან მარჯვნივ ყველა ციფრს ვავსებთ (k – 1) - მდე და უმცროსს ვუმატებთ მინუს ერთს. ორობითი რიცხვების შემთხვევაში ყველა თანრიგში ვატარებთ ინვერსიას და უმცროს თანრიგს ვუმატებთ უარყოფით ერთიანს.

2. უფრო ადვილია რიცხვის აღდგენა, როდესაც მარცხნიდან მარჯვნივ ნიშნის თანრიგის ჩათვლით ყველა თანრიგის ციფრებს შევავსებთ (k – 1) - მდე, გარდა მარჯვენა კიდურა ნულისაგან განსხვავებული ციფრისა, რომელსაც ვავსებთ k - მდე, მის შემდეგ მდგომ ნულებს ვტოვებთ უცვლელად. ორობითი რიცხვების შემთხვევაში მარჯვენა კიდურა ერთიანი და მის შემდეგ მდგომი ნულები რჩება უცვლელად.

წესიერი წილადების დროს რიცხვის აღდგენა შეგვიძლია (1.23) ფორმულიდან $x = [x]_{დ} - 10 = -\{10 - [x]_{დ}\}$.

წესიერი წილადების აღდგენისათვის ჩამოყალიბებული წესი მართებულია როგორც მთელი რიცხვებისათვის, ასევე არაწესიერი წილადებისათვის, იმ განსხვავებით, რომ ნიშნის თანრიგში ჩაწერილი ციფრი უნდა შევცვალოთ არა „-0“ მთელით, არამედ მხოლოდ „ – “ - ით.

მთელი რიცხვებისა და არაწესიერი წილადების აღდგენა შეგვიძლია (1.24) ფორმულიდან $x = -10^{n+1} + [x]_{დ} = -\{10^{n+1} - [x]_{დ}\}$.

დამატებითი კოდიდან პირდაპირი კოდის მისაღებად ნიშნის თანრიგს ვტოვებთ უცვლელად, ციფრულ თანრიგებს კი ვცვლით ისევე, როგორც რიცხვის აღდგენის დროს.

მოცემული დამატებითი კოდებიდან აღვადგინოთ რიცხვები და შევქმნათ მათი პირდაპირი კოდი.

1. $[x]_{\text{დ}} = 1'10011 \quad k = 2, \quad x = -\{10 - [x]_{\text{დ}}\} = -(10{,}00000$

$$\begin{array}{r} -1'10011) \\ \hline -0{,}01101 \end{array}$$

$x = -0{,}01100$

$$\begin{array}{r} + \quad -1 \\ \hline -0{,}01101 \end{array}$$

$[x]_{\text{პ}} = 1'01101$

2. $[x]_{\text{დ}} = 1.01100' \quad k = 2, \quad x = -\{10^{n+1} - [x]_{\text{დ}}\} = -(1000000$

$$\begin{array}{r} -1.01100') \\ \hline -10100 \end{array}$$

$x = 10100$

$[x]_{\text{პ}} = 1.10100'$

3. $[x]_{\text{დ}} = 7.20350' \qquad k = 8, \qquad x = -(1000000$

$$\begin{array}{r} -720350) \\ \hline -057430 \end{array}$$

$x = -57430$

$[x]_{\text{პ}} = 7.57430'$

მთელი რიცხვის მარცხნივ მიწერილ ნულს ვუგულვებელყოფთ.

4. $[x]_{\text{დ}} = F.B2DA0' \qquad k = 16, \qquad x = -(1000000$

$$\begin{array}{r} -F.B2DA0) \\ \hline -4D260 \end{array}$$

$x = -4D260$

$[x]_{\text{პ}} = F.4D260'$

5. $[x]_{\text{დ}} = 9.0255' \qquad k = 10, \qquad x = -(1000000$

$$\begin{array}{r} -90255) \\ \hline -9745 \end{array}$$

$x = -9745$

$[x]_{\text{პ}} = 9.9745'$

6. $[x]_{ფ} = 9.965'27$ $\qquad\qquad$ $x = -(10000'00$

$x = -034,73 = -34,73$ $\qquad\qquad$ $-9.965'27)$

$[x]_{კ} = 9.034'73$ $\qquad\qquad$ $\overline{}$ $-0034,73 = -34,73$

7. $[x]_{ფ} = 1.101'011$ $\qquad\qquad$ $x = -(10000'000$

$x = -010,101 = -10,101$ $\qquad\qquad$ $-1.101'011)$

$[x]_{კ} = 1.010'101$ $\qquad\qquad$ $-0010,101 = -10,101$

1.8. თანრიგთა ბადის გადავსება.

როგორც აღვნიშნეთ, ფიქსირებულმძიმიან მანქანებში ოპერაციებს ძირითადად ვაწარმოებთ წესიერ ორობით წილადებზე, ე. ი. ადგილი უნდა ჰქონდეს (1.18) უტოლობებს $|x| < 1$, $|y| < 1$ და $|x + y| < 1$ გამორიცხული არ არის შემთხვევები, როდესაც პირველი ორი უტოლობა დაცულია, მესამე კი არა. ეს იმას ნიშნავს, რომ ორი წესიერი წილადის შეკრების შედეგად შეიძლება მივიღოთ არაწესიერი წილადი. ცხადია ასეთ დროს შესაკრებებს უნდა ჰქონდეს ერთნაირი ნიშანი.

ვთქვათ, $x = 0,1011$ და $y = 0,1001$. მაშინ გვექნება:

$$
\begin{array}{r}
x = 0,1011 \\
+\; y = 0,1001 \\
\hline
x + y = 1,0100
\end{array}
\quad
\begin{array}{c}
\rightarrow \\
\rightarrow \\

\end{array}
\quad
\begin{array}{r}
[x]_{პ} = 0'1011 \\
+\; [y]_{პ} = 0'1001 \\
\hline
[x]_{პ} + [y]_{პ} = 1'0100
\end{array}
$$

$$\downarrow$$

$[x + y]_{პ}$ – არ არსებობს.

ვინაიდან ოპერაციებს ვაწარმოებთ წესიერ წილადებზე, ამიტომ უჯრედში როგორც რიცხვის მთელი ნაწილისათვის, ისე მათი კოდებისათვის თანრიგები არა გვაქვს გამოყოფილი. ჩვენს შემთხვე-

ვაში ჯამი მივიღეთ არაწესიერი წილადი, ამიტომ მისი კოდი ასეთ უჯრედში ვერ ჩაიწერება.

ამჯამავის გამოსავალზე მიღებული შედეგი - კოდების ჯამი $1'0100$ წარმოადგენს უარყოფითი წესიერი წილადის - 0,1011 შებრუნებულ კოდს, რაც ჯამის ნამდვილ მნიშვნელობას არ შეესაბამება. მიღებული მცდარი შედეგი გამოწვეულია იმით, რომ ჯამის მთელი ნაწილი მოხვდა ნიშნის თანრიგში. ორი დადებითი რიცხვის შეკრებამ მოგვცა უარყოფითი რიცხვის კოდი. ამ მოვლენას ეწოდება თანრიგთა ბადის გადავსება, იგი დროულად უნდა იქნეს შემჩნეული, გამომუშავდეს ოპერაციის შეწყვეტის სიგნალი და შეიცვალოს სამაშტაბო კოეფიციენტები.

თუ შესაკრებები უარყოფითია $x -= 0{,}1011$ და $y = -0{,}1001$. მაშინ გვექნება:

$$
\begin{array}{r}
+\begin{array}{r} x = -0{,}1011 \\ y = -0{,}1001 \end{array} \\
\hline
x + y = 1{,}0100 \\
\downarrow \\
[x+y]_{\text{შ}} - \text{არ არსებობს}
\end{array}
\quad
\begin{array}{c} \to \\ \to \end{array}
\quad
\begin{array}{r}
+\begin{array}{r} [x]_{\text{შ}} = 1'0100 \\ [y]_{\text{შ}} = 1'0110 \end{array} \\
\hline
[x]_{\text{შ}} + [y]_{\text{შ}} = 10'1010 \\
+\ 1 \\
\hline
0'1011
\end{array}
$$

ანალოგიურ სურათს მივიღებთ დამატებით კოდის შემთხვევაშიც

$$
\begin{array}{r}
+\begin{array}{r} [x]_{\text{დ}} = 1'0101 \\ [y]_{\text{დ}} = 1'0111 \end{array} \\
\hline
[x]_{\text{დ}} + [y]_{\text{დ}} = 10'1100 \\
\swarrow \\
\text{გადაიგდება}
\end{array}
$$

განხილულ მაგალითებში, შესაკრებთა ჯამი არის უარყოფითი არაწესიერი წილადი, ხოლო მათი როგორც შებრუნებული, ისე დამატებითი კოდების ჯამი, შესაბამისად წარმოადგენენ დადებითი წესიერი 0,1011 და 0,1100 წილადების კოდებს, რაც სინამდვილეს არ

შეესაბამება, ვინაიდან ადგილი აქვს თ ა ნ რ ი გ თ ა ბ ა დ ი ს გ ა-დ ა ვ ს ე ბ ა ს .

მაშასადამე, როდესაც $|x + y| \geq 1$, ადგილი აქვს თანრიგთა ბადის გადავსებას. თუ შესაკრებები დადებითია, მივიღებთ დ ა დ ე-ბ ი თ გ ა დ ა ვ ს ე ბ ა ს, ხოლო თუ შესაკრებები უარყოფითია - უ ა-რ ყ ო ფ ი თ გ ა დ ა ვ ს ე ბ ა ს.

ზემოთ განხილულ მაგალითებში შესაკრებთა კოდების და კოდების ჯამის ნიშნის თანრიგებში ჩაწერილი ციფრების შედარების შედეგად დავასკვნით, რომ თანრიგთა ბადის გადავსების დასადგენად როგორც შებრუნებულ, ისე დამატებით კოდში უნდა ჩავატაროთ ორი შედარების ოპერაცია. პირველ ყოვლისა ვადარებთ კოდების ნიშნის თანრიგის ციფრებს. თუ ისინი ერთნაირია, ე. ი. შესაკრებები ერთნაირი ნიშნისაა, მაშინ მათ ვადარებთ კოდების ჯამის ნიშნის თანრიგში მიღებულ ციფრს. თუ ეს ციფრები ერთმანეთისაგან განსხვავდებიან, ადგილი აქვს თანრიგთა ბადის გადავსებას. მაშასადამე, გადავსების მაჩვენებელია, როდესაც შედეგის და შესაკრებთა ნიშანი მოპირდაპირეა.

დადებითი გადავსების დროს ამჯამავის გამოსავალზე ვიღებთ უარყოფითი რიცხვის კოდს, ხოლო უარყოფითი გადავსებისას კი დადებითი რიცხვის კოდს.

თანრიგთა ბადის გადავსების მაჩვენებელია აგრეთვე კოდების შეკრების დროს გადატანის ერთიანის მიღება, მხოლოდ ციფრული თანრიგიდან ნიშნის თანრიგში, და კოდების ჯამის ნიშნის თანრიგში მივიღებთ 1 -ს - დ ა დ ე ბ ი თ ი გ ა დ ა ვ ს ე ბ ა, ან მხოლოდ ნიშნის თანრიგიდან და კოდების ჯამის ნიშნის თანრიგში მივიღებთ „0“ -ს - უ ა რ ყ ო ფ ი თ ი გ ა დ ა ვ ს ე ბ ა.

თუ გადატანის ერთიანს ვიღებთ როგორც ციფრული, ასევე ნიშნის თანრიგიდან, ან საერთოდ არ ვიღებთ გადატანის ერთიანს, მაშინ თანრიგთა ბადის გადავსებას ადგილი არა აქვს.

ვთქვათ, $x = 0{,}1101$ და $y = -0{,}0001$ და $-1 < x - y < 0.$

$$
\begin{array}{r}
+\begin{array}{r} x = -0{,}1101 \\ y = -0{,}0001 \end{array} \\
\hline
x + y = -0{,}1110
\end{array}
\quad
\begin{array}{c} \rightarrow \\ \rightarrow \end{array}
\quad
\begin{array}{r}
+\begin{array}{r} [x]_{\text{დ}} = 1'_{.}0011 \\ [y]_{\text{დ}} = 1'_{.}1111 \end{array} \\
\hline
[x]_{\text{დ}} + [y]_{\text{დ}} = 11'_{.}0010
\end{array}
$$

$$
\downarrow
$$

$$
[x+y]_{\text{დ}} = 1'_{.}0010
$$

1 1

გადაიგდება

როგორც ვხედავთ, კოდების შეკრების დროს როგორც ციფ-რული, ასევე ნიშნის თანრიგიდან მივიღეთ გადატანის ერთიანები, ე. ი. ადგილი არა აქვს თანრიგთა ბადის გადავსებას, რაც აგრეთვე დასტურდება კოდების ნიშნის თანრიგებში და კოდების ჯამის ნიშ-ნის თანრიგში ერთნაირი ციფრებით.

თანრიგთა ბადის გადავსების ადვილად დასადგენად გამოიყე-ნება მოდიფიცირებული შებრუნებული და მოდიფიცირებული დამატებითი კოდები.

1.9. მოდიფიცირებული კოდები.

მოდიფიცირებული პირდაპირი, შერუნებული და დამატები-თი კოდები ჩვეულებრივი კოდებისაგან განსხვავდებიან მხოლოდ იმით, რომ ნიშნისათვის გამოყოფილი გვაქვს ორი თანრიგი.

თუ რიცხვი დადებითია, ნიშნის თანრიგებში ჩაიწერება ორი ნული, ხოლო თუ რიცხვი უარყოფითია, ნიშნის ორივე თანრიგში გვექნება $k-1$, ორობითი სისტემის დროს - ერთიანები, ციფრულ თანრიგებს კი ვექცევით ისევე, როგორც უარყოფითი რიცხვების ჩვე-ულებრივი კოდების შექმნის დროს.

საბოლოოდ წესიერი წილადისთვის გვექნება:

$$[x]^{\text{მ}}_{\text{შ}} = \begin{cases} x, & \text{თუ } x \geq 0; \\ 100 - 10^{-n} + x, & \text{თუ } x \leq 0. \end{cases}$$

$$[x]^{\text{მ}}_{\text{დ}} = \begin{cases} x, & \text{თუ } x \geq 0; \\ 100 + x, & \text{თუ } x \leq 0. \end{cases}$$

მთელი რიცხვების შემთხვევაში:

$$[x]^{\text{მ}}_{\text{შ}} = \begin{cases} x, & \text{თუ } x \geq 0; \\ 10^{n+2} - 1 + x, & \text{თუ } x \leq 0. \end{cases} \quad (1.25)$$

არაწესიერი წილადებისათვის:

$$[x]^{\text{მ}}_{\text{შ}} = \begin{cases} x, & \text{თუ } x \geq 0; \\ 10^{n+2} - 10^{-r} + x, & \text{თუ } x \leq 0. \end{cases} \quad (1.26)$$

(1.25) და (1.26) ფორმულებში n არის უჯრედში რიცხვის მთელი ნაწილისათვის გამოყოფილი თანრიგების რაოდენობა, ხოლო (1.26) ფორმულაში r - გამოყოფილი წილადი ნაწილისათვის თანრიგის რაოდენობა.

მოდიფიცირებული დამატებითი კოდის დროს, როგორც მთელი რიცხვების, ისე არაწესიერი წილადებისათვის გვაქვს შემდეგი ფორმულა:

$$[x]^{\text{მ}}_{\text{დ}} = \begin{cases} x, & \text{თუ } x \geq 0; \\ 10^{n+2} + x, & \text{თუ } x \leq 0. \end{cases} \quad (1.27)$$

ორი წესიერი წილადის მოდიფიცირებული, შებრუნებული ან დამატებითი კოდების შეკრების დროს ადგილი უნდა ჰქონდეს (1.18) უტოლობებს $|x| < 1$, $|y| < 1$ და $|x + y| < 1$ და ალგებრული შეკრების ყველა შემთხვევაში <u>კოდების ჯამი უნდა უდრიდეს ჯამის კოდს.</u>

გამოკლების ოპერაცია დაიყვანება საკლებისა და მაკლების მოდიფიცირებული კოდების შეკრების ოპერაციამდე, შედეგად ამჯამავის გამოსავალზე მივიღებთ სხვაობის მოდიფიცირებულ კოდს:

$$[x]^{\text{მ}}_{\text{შ}} + [-y]^{\text{მ}}_{\text{შ}} = [x - y]^{\text{მ}}_{\text{შ}}, \qquad [x]^{\text{მ}}_{\text{დ}} + [-y]^{\text{მ}}_{\text{დ}} = [x - y]^{\text{მ}}_{\text{დ}}.$$

ალგებრული შეკრების მაგალითები როგორც მოდიფიცირებულ შებრუნებულ, ასევე მოდიფიცირებულ დამატებით კოდებში.

1. ორივე შესაკრები დადებითია.

$0 < x < 1, \quad 0 < y < 1$ და $0 < x + y < 1,$

x = 0,1101, y = 0,0001.

$$
\begin{array}{rcr}
x = 0{,}1101 & \rightarrow & [x]^{\text{შ}}_{\text{მ}} = 00'1101 \\
+\; y = 0{,}0001 & \rightarrow & +\; [y]^{\text{შ}}_{\text{მ}} = 00'0001 \\
\hline
x + y = 0{,}1110 & & [x]^{\text{შ}}_{\text{მ}} + [y]^{\text{შ}}_{\text{მ}} = \underbrace{00'1110} \\
\downarrow & & \\
[x + y]^{\text{შ}}_{\text{მ}} = \underbrace{00'1110} & &
\end{array}
$$

2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია.

$0 < x < 1, \quad -1 < y < 0$ და $0 < x + y < 1.$

$$
\begin{array}{rcr}
x = 0{,}1101 & \rightarrow & [x]^{\text{შ}}_{\text{მ}} = 00'1101 \\
+\; y = -0{,}0001 & \rightarrow & +\; [y]^{\text{შ}}_{\text{მ}} = 11'1110 \\
\hline
x + y = 0{,}1100 & & [x]^{\text{შ}}_{\text{მ}} + [y]^{\text{შ}}_{\text{მ}} = 100'1011 \\
\downarrow & & +\; \text{ც. გ.} \quad 1 \\
\cline{3-3}
[x + y]^{\text{შ}}_{\text{მ}} = \underbrace{00'1100} & & \underbrace{00'1100}
\end{array}
$$

3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია.

-1 < x < 0, 0 < y < 1 და -1 < x + y <0.

$$\begin{array}{r} x = -0{,}1101 \\ +\ y = 0{,}0001 \\ \hline x + y = -0{,}1110 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \\ \end{array} \quad \begin{array}{r} [x]^{\text{მ}}_{\text{ვ}} = 11'0010 \\ +\ [y]^{\text{მ}}_{\text{ვ}} = 00'0001 \\ \hline [x]^{\text{მ}}_{\text{ვ}} + [y]^{\text{მ}}_{\text{ვ}} = \underbrace{11'0011} \end{array}$$

$$\downarrow$$

$$[x + y]^{\text{მ}}_{\text{ვ}} = \underbrace{11'0011}$$

4. ორივე შესაკრები უარყოფითია.

$$-1 < x < 0, \quad -1 < y < 0 \quad \text{და} \quad -1 < x + y < 0.$$

$$\begin{array}{r} x = -0{,}1101 \\ +\ y = -0{,}0001 \\ \hline x + y = -0{,}1110 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \\ \end{array} \quad \begin{array}{r} [x]^{\text{მ}}_{\text{ვ}} = 11'0010 \\ +\ [y]^{\text{მ}}_{\text{ვ}} = 11'1110 \\ \hline [x]^{\text{მ}}_{\text{ვ}} + [y]^{\text{მ}}_{\text{ვ}} = 111'0000 \\ +\ \text{ც. გ.} \quad 1 \\ \hline \underbrace{11'0001} \end{array}$$

$$\downarrow$$

$$[x + y]^{\text{მ}}_{\text{ვ}} = \underbrace{11'0001}$$

შევქმნათ შემდეგი უარყოფითი მთელი რიცხვების მოდიფი-ცირებული დამატებითი კოდები, $n = 5$:

1) -10100_2, 2) -260_8, 3) -8900_{10}, 4) $-A3FDO_{16}$.

1. $[-10100_2]^{\text{მ}}_{\text{დ}} = 11.01100'$ ცხადია, იმავე შედეგს მივიღებთ (2.25) გამოსახულებიდანაც:

$$[-10100_2]^{\text{მ}}_{\text{დ}} = 10^7 - 10100 = \begin{array}{r} 10000000 \\ -\quad 10100 \\ \hline 11.01100' \end{array}$$

2. $[-00260_8]^{\partial}_{\varphi} = 77.77520'$

$$[-00260_8]^{\partial}_{\varphi} = 10^7 - 10260 = \begin{array}{r} 10000000 \\ - \quad 260 \\ \hline 77.77520' \end{array}$$

3. $[-08900_{10}]^{\partial}_{\varphi} = 99.91100'$

$$[-08900_{10}]^{\partial}_{\varphi} = \begin{array}{r} 10000000 \\ - \quad 8900 \\ \hline 99.91100' \end{array}$$

4. $[-A3FD0_{16}]^{\partial}_{\varphi} = FF.5C030'$

$$[-A3FD0_{16}]^{\partial}_{\varphi} = \begin{array}{r} 10000000 \\ - \quad A3FD0 \\ \hline FF.5C030' \end{array}$$

ახლა შევქმნათ შემდეგი უარყოფითი არაწესიერი წილადების მოდიფიცირებული დამატებითი კოდები, n =3:

5) -011,110$_2$, 6) -075,02$_8$, 7) -83,94$_{10}$, 8) -005,0 A$_{16}$.

5. $[-011{,}110_2]^{\partial}_{\varphi} = 11100'010$, ცხადია, იმავე შედეგს მივიღებთ (2.25) გამოსახულებიდანაც:

$$[-011{,}110_2]^{\partial}_{\varphi} = 10^5 - 11{,}110 = \begin{array}{r} 100000{,}000 \\ - \quad 11{,}110 \\ \hline 11100'010 \end{array}$$

6. $[-075{,}02_8]^{\partial}_{\varphi} = 77.702'76$

$$[-075{,}02_2]^{\partial}_{\varphi} = \begin{array}{r} 100000{,}000 \\ - \quad 75{,}02 \\ \hline 77.702'76 \end{array}$$

7. $[-083{,}94_{10}]^{\partial}_{\varphi} = 99.916'06$

$$[-083{,}94_{10}]^{\partial}_{\varphi} = \begin{array}{r} 100000{,}00 \\ - \quad 83{,}94 \\ \hline 99.916'06 \end{array}$$

8. $[-005{,}0A_{16}]_{დ}^{მ} = FF.FFA'F6$

$$[-005{,}0A_{16}]_{დ}^{მ} = \begin{array}{r} 100000{,}00 \\ -\quad 5{,}0A \\ \hline FF.FFA'F6 \end{array}$$

ჩავატაროთ $x = 0{,}11001$ *და* $y = 0{,}00101$ *რიცხვების ალგებრული შეკრება.*

1. ორივე შესაკრები დადებითია.

$0 < x < 1,\quad 0 < y < 1$ და $0 < x + y < 1.$

$$\begin{array}{r} x = 0{,}11001 \\ +\; y = 0{,}00101 \\ \hline x + y = 0{,}11110 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \\ \\ \end{array} \quad \begin{array}{r} [x]_{დ}^{მ} = 00'11001 \\ +\; [y]_{დ}^{მ} = 00'00101 \\ \hline [x]_{დ}^{მ} + [y]_{დ}^{მ} = 00'\underbrace{11110} \end{array}$$

$$\downarrow$$

$$[x + y]_{დ}^{მ} = 00'\underbrace{11110}$$

2. ერთი შესაკრები უარყოფითია, მაგრამ ჯამი დადებითია.

$0 < x < 1,\quad -1 < y < 0$ და $0 < x + y < 1.$

$$\begin{array}{r} +\; x = 0{,}11001 \\ y = -0{,}00101 \\ \hline x + y = 0{,}10100 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \\ \\ \end{array} \quad \begin{array}{r} [x]_{დ}^{მ} = 00'11001 \\ +\; [y]_{დ}^{მ} = 11'11011 \\ \hline [x]_{დ}^{მ} + [y]_{დ}^{მ} = 100'\underbrace{10100} \end{array}$$

$$\downarrow$$

$$[x + y]_{დ}^{მ} = 00'\underbrace{10100}$$

გადაიგდება

3. ერთი შესაკრები უარყოფითია და ჯამიც უარყოფითია.

$$-1 < x < 0, \quad 0 < y < 1 \quad \text{და} \quad -1 < x + y < 0.$$

$$
\begin{array}{rl}
+ & x = -0{,}11001 \\
 & y = 0{,}00101 \\
\hline
 & x + y = -0{,}10100
\end{array}
\quad
\begin{array}{c} \rightarrow \\ \rightarrow \\ \\ \end{array}
\quad
\begin{array}{rl}
+ & [x]^{\text{მ}}_{\text{დ}} = 11'00111 \\
 & [y]^{\text{მ}}_{\text{დ}} = 00'00101 \\
\hline
 & [x]^{\text{მ}}_{\text{დ}} + [y]^{\text{მ}}_{\text{დ}} = 11'01100
\end{array}
$$

$$[x + y]^{\text{მ}}_{\text{დ}} = 11'01100$$

4. ორივე შესაკრები უარყოფითია.

$$-1 < x < 0, \quad -1 < y < 0 \quad \text{და} \quad -1 < x + y < 0.$$

$$
\begin{array}{rl}
+ & x = -0{,}11001 \\
 & y = -0{,}00101 \\
\hline
 & x + y = -0{,}10100
\end{array}
\quad
\begin{array}{c} \rightarrow \\ \rightarrow \\ \\ \end{array}
\quad
\begin{array}{rl}
+ & [x]^{\text{მ}}_{\text{დ}} = 11'00111 \\
 & [y]^{\text{მ}}_{\text{დ}} = 11'11011 \\
\hline
 & [x]^{\text{მ}}_{\text{დ}} + [y]^{\text{მ}}_{\text{დ}} = 111'00010
\end{array}
$$

გადაიგდება

$$[x + y]^{\text{მ}}_{\text{დ}} = 11'00010$$

როგორც აღვნიშნეთ, წესიერი წილადების შეკრების დროს, თუ შესაკრებთა ჯამი არ აკმაყოფილებს (1.18) გამოსახულების მესამე უტოლობას, ე. ი. $|x + y| \geq 1$, მაშინ მივიღებთ თანრიგთა ბადის გადავსებას.

თუ შესაკრებები დადებითია, როგორც მოდიფიცირებულ შებრუნებულ, ისე დამატებით კოდში მივიღებთ ერთნაირ შედეგს.

1. ვთქვათ, $x = 0{,}1011$ და $y = 0{,}1001$.

$$
\begin{array}{rl}
+\ x = & 0{,}1011 \\
y = & 0{,}1001 \\
\hline
x + y = & 1{,}0100
\end{array}
\quad
\begin{array}{c} \to \\ \to \\ \\ \end{array}
\quad
\begin{array}{rl}
+\ [x]_{\text{შ}}^{\text{მ}} = & 00'1011 \\
[y]_{\text{შ}}^{\text{მ}} = & 00'1001 \\
\hline
[x]_{\text{შ}}^{\text{მ}} + [y]_{\text{შ}}^{\text{მ}} = & 01'0100 \\
 & \ \ 1 \\
 & \text{გადატანა}
\end{array}
$$

ამჯამავის გამოსავალზე მიღებული შედეგი არ წარმოადგენს წესიერი წილადის მოდიფიცირებულ კოდს - ვინაიდან ნიშნის თანრიგში მივიღეთ განსხვავებული ციფრები 01, და გადატანის ერთიანიც მხოლოდ უფროსი ციფრული თანრიგიდან.

როდესაც ორივე შესაკრები უარყოფითია, გვექნება:

$$
\begin{array}{rl}
+\ x = & -0{,}1011 \\
y = & -0{,}1001 \\
\hline
x + y = & -1{,}0100 \\
 & \downarrow \\
\multicolumn{2}{c}{[x+y]_{\text{შ}}^{\text{მ}} - \text{არ არსებობს}}
\end{array}
\quad
\begin{array}{c} \to \\ \to \\ \\ \end{array}
\quad
\begin{array}{rl}
+\ [x]_{\text{შ}}^{\text{მ}} = & 11'0100 \\
[y]_{\text{შ}}^{\text{მ}} = & 11'0110 \\
\hline
[x]_{\text{შ}}^{\text{მ}} + [y]_{\text{შ}}^{\text{მ}} = & 110'1010 \\
 & \ \ 1 \qquad + \\
 & \qquad\qquad 1 \\
\hline
 & 10'1011
\end{array}
$$

იმავე სურათს მივიღებთ დამატებით კოდშიც:

$$
\begin{array}{rl}
+\ x = & -0{,}1011 \\
y = & -0{,}1001 \\
\hline
x + y = & -1{,}0100 \\
 & \downarrow \\
\multicolumn{2}{c}{[x+y]_{\text{დ}}^{\text{მ}} - \text{არ არსებობს}}
\end{array}
\quad
\begin{array}{c} \to \\ \to \\ \\ \end{array}
\quad
\begin{array}{rl}
+\ [x]_{\text{დ}}^{\text{მ}} = & 11'0101 \\
[y]_{\text{დ}}^{\text{მ}} = & 11'0111 \\
\hline
[x]_{\text{დ}}^{\text{მ}} + [y]_{\text{დ}}^{\text{მ}} = & 110'1100 \\
 & \ \ 1 \\
\text{გადაიგდება} &
\end{array}
$$

ამ შემთხვევაშიც ამჯამავის გამოსავალზე ნიშნის თანრიგებში მივიღეთ განსხვავებული ციფრები, მხოლოდ სხვა მიმდევრობით 10, და გადატანის ერთიანი, მხოლოდ ნიშნის უფროსი თანრიგიდან.

მაშასადამე, მოდიფიცირებული შებრუნებული ან დამატებითი კოდების დროს თანრიგთა ბადის გადავსების დასადგენად საკმარისია შევადგინოთ ერთი შედარების ოპერაცია. თუ ნიშნის თანრიგებში მივიღეთ განსხვავებული ციფრები - ციფრების დაუშვებელი კომბინაცია - ადგილი აქვს თანრიგთა ბადის გადავსებას.

გადავსების ნიშანს განსაზღვრავს ციფრი, რომელიც მივიღეთ ნიშნის უფროს (მარცხნიდან პირველ) თანრიგში, ამიტომ მოდიფიცირებული კოდების დროს ნიშნის უფროს თანრიგს, რომელიც გვიჩვენებს გადავსების ნიშანს, ხშირად ნიშნის თანრიგს უწოდებენ. ნიშნის უმცროს თანრიგს, რომელიც იღებს რიცხვის ნიშნის მოპირდაპირე მნიშვნელობას და მიუთითებს თანრიგის ბადის გადავსებაზე, გადავსების თანრიგს უწოდებენ.

მაშასადამე, ნიშნის თანრიგში მიღებული ციფრები 01 დადებითი გადავსების მაჩვენებელია და ამ დროს გადატანის ერთიანი მივიღეთ მხოლოდ უფროსი ციფრული თანრიგიდან, ხოლო 10 - უარყოფითი გადავსების მაჩვენებელია და გადატანის ერთიანი მივიღეთ მხოლოდ ნიშნის უფროსი თანრიგიდან.

k - ობითი სისტემის შემთხვევაში დადებითი გადავსების დროს, ნიშნის თანრიგებში, ისევე როგორც ორობითი სისტემის შემთხვევაში, მივიღებთ 01 - ს, ხოლო უარყოფითი გადავსების დროს - $k-1$, $k-2$ – ს.

განვიხილოთ მაგალითები რვაობით და თექვსმეტობით სისტემაში:

$$x = 0{,}761, \; y = 0{,}206, \; k = 8.$$

$$\begin{array}{rl} + & x = 0{,}761 \\ & y = 0{,}206 \\ \hline & x + y = 1{,}167 \end{array} \quad \begin{array}{c} \rightarrow \\ \rightarrow \\ \end{array} \quad \begin{array}{rl} + & [x]^{გ}_{დ} = 00;761 \\ & [y]^{გ}_{დ} = 00;206 \\ \hline & [x]^{გ}_{დ} + [y]^{გ}_{დ} = 01;167 \end{array}$$

$$\downarrow$$

$[x+y]^{გ}_{დ}$ – არ არსებობს

მივიღეთ დადებითი გადავსება 01 მაჩვენებლით.

$$
\begin{array}{rcl}
\begin{array}{r} x = -0{,}761 \\ +\quad y = -0{,}206 \\ \hline x + y = -1{,}167 \end{array} & \begin{array}{c} \rightarrow \\ \rightarrow \\ \end{array} & \begin{array}{r} [x]^{მ}_{დ} = 77{'_{.}}017 \\ +\quad [y]^{მ}_{დ} = 77{'_{.}}572 \\ \hline [x]^{მ}_{დ} + [y]^{მ}_{დ} = 176{'_{.}}614 \end{array} \\
\downarrow & & \swarrow \\
[x+y]^{მ}_{დ} - \text{არ არსებობს} & & \text{გადაიგდება}
\end{array}
$$

ნიშნის თანრიგებში მიღებული 7 და 6 ციფრები რვაობით სისტემაში შესაბამისად წარმოადგენენ $k-1$ და $k-2$, რაც უარყოფითი გადავსების მაჩვენებელია.

$$x = -0,A5CF, \quad y = -0{,}7EDB, \quad k = 16.$$

$$
\begin{array}{rcl}
\begin{array}{r} x = -0,A5CF \\ +\quad y = -0{,}7EDB \\ \hline x + y = -1{,}24AA \end{array} & \begin{array}{c} \rightarrow \\ \rightarrow \\ \end{array} & \begin{array}{r} [x]^{მ}_{შ} = FF{'_{.}}5A30 \\ +\quad [y]^{მ}_{შ} = FF{'_{.}}8124 \\ \hline [x]^{მ}_{შ} + [y]^{მ}_{შ} = 1FE{'_{.}}DB54 \\ \text{ც. გ.} \qquad + \\ \longrightarrow 1 \\ \hline FE{'_{.}}DB55 \end{array} \\
\downarrow & & \\
[x+y]^{მ}_{შ} - \text{არ არსებობს} & &
\end{array}
$$

ნიშნის თანრიგებში მიღებული F და E ციფრები თექვსმეტობით სისტემაში შესაბამისად $k-1$ და $k-2$ -ია.

უარყოფითი წესიერი წილადების შემთხვევაში, შებრუნებული კოდის დროს, ციფრული ნაწილი ივსება $1-10^{-n}$ - მდე, ამიტომ შებრუნებული კოდით შეიძლება წარმოვადგინოთ რიცხვები $[-(1-10^{-n}) \div 1-10^{-n}]$ ინტერვალიდან, ხოლო დამატებითი კოდის დროს ციფრული ნაწილი ივსება ერთამდე და შესაბამისად დამატებითი კიდით შეიძლება წარმოვადგინოთ რიცხვები ინტერვალიდან $[-1 \div 1-10^{-n}]$.

სწორედ ამ გარემოებითაა განპირობებული, რომ უარყოფითი გადავსების შებრუნებულ კოდში ვიღებთ, როდესაც $(x+y) \leq -1$, ხოლო დამატებით კოდში მხოლოდ მაშინ, როდესაც $(x+y) < -1$-ზე.

თუ $x = -(1-10^{-n})$, მაშინ მის შებრუნებულ კოდს k - ობით სისტემაში ექნება შემდეგი სახე: $[x]_{შ} = k-1{,}00\ldots0$, ორობითი სის-

ტემის დროს კი $[x]_{\text{პ}} = 1{:}00 \ldots 0$. მინუს ერთის დამატებითი კოდი იქ-ნება

$$
\begin{array}{r}
[-1{,}00 \ldots 00]_{\text{დ}} = 0'11 \ldots 11 \\
\overset{+}{1} \\
\hline
1'00 \ldots 00
\end{array}
$$

თუ მოვახდენთ რიცხვების აღდგენას k - ობითი სისტემის დროს, გვექნება

$$
\begin{array}{r}
x = -\{10 - [x]_{\text{დ}}\} = -(10'00 \ldots 0 \\
\underline{-(k-1)'00 \ldots 0)} \\
-1'00 \ldots 0
\end{array}
$$

ორობითი სისტემის შემთხვევაშიც -

$$
\begin{array}{r}
x = -\{10 - [x]_{\text{დ}}\} = -(10'00 \ldots 0 \\
\underline{-1'00 \ldots 0)} \\
-1'00 \ldots 0
\end{array}
$$

დამატებითი კოდის ფორმულებიდანაც მივიღებთ იმავე შე-დეგს.

$$-1 < x < 0,\ -1 < y < 0 \quad \text{და} \quad x + y = -1, \text{ მაშინ}$$

$$[x]_{\text{დ}} + [y]_{\text{დ}} = 10 + x + 10 + y = 100 + (x + y) = 11'00 \ldots 0 = 1'00 \ldots 0$$

გადაიგდება

$$[x]_{\text{დ}}^{\partial} + [y]_{\text{დ}}^{\partial} = 100 + x + 100 + y = 1000 + (x + y) =$$

$$= 111'00 \ldots 0 = 11'00 \ldots 0$$

გადაიგდება

განვიხილოთ მაგალითი, როდესაც შესაკრებთა ჯამი მინუს ერთის ტოლია:

$$
\begin{array}{rcl}
x = -0{,}0101 & \rightarrow & [x]_{\text{დ}} = 1'1011 \\
+\; y = -0{,}1011 & \rightarrow & +\; [y]_{\text{დ}} = 1'0101 \\
\hline
x + y = -1{,}0000 & & [x]_{\text{დ}} + [y]_{\text{დ}} = 11'0000 \\
\downarrow & & \\
[x+y]_{\text{დ}} = 1'0000 & &
\end{array}
$$

გადაიგდება

მოდიფიცირებულ დამატებით კოდში გვექნება:

$$
\begin{array}{rcl}
x = -0{,}0101 & \rightarrow & [x]_{\text{დ}}^{\text{მ}} = 11'1011 \\
+\; y = -0{,}1011 & \rightarrow & +\; [y]_{\text{დ}}^{\text{მ}} = 11'0101 \\
\hline
x + y = -1{,}0000 & & [x]_{\text{დ}} + [y]_{\text{დ}} = 111'0000 \\
\downarrow & & \\
[x+y]_{\text{დ}}^{\text{მ}} = 11'0011 & &
\end{array}
$$

გადაიგდება

იმავე შესაკრებთა შებრუნებული კოდების ჯამი კი მოგვცემს თანრიგთა ბადის უარყოფით გადავსებას:

$$
\begin{array}{rcl}
x = -0{,}0101 & \rightarrow & [x]_{\text{შ}}^{\text{მ}} = 11'1010 \\
+\; y = -0{,}1011 & \rightarrow & +\; [y]_{\text{შ}}^{\text{მ}} = 11'0100 \\
\hline
x + y = -1{,}0000 & & [x]_{\text{შ}}^{\text{მ}} + [y]_{\text{შ}}^{\text{მ}} = 100'1110 \\
\downarrow & & \text{ც. გ.} \quad +\; 1 \\
\cline{3-3}
[x+y]_{\text{შ}}^{\text{მ}} \text{ - არ არსებობს} & & 10'1111
\end{array}
$$

ნაწილი II

კომპიუტერის ლოგიკური საფუძვლები

კომპიუტერის სქემების ანალიზისა და სინთეზის დროს ფართოდ გამოიყენება ლოგიკის ალგებრის მათემატიკური აპარატი, კერძოდ, ბულის ალგებრა.

ლოგიკის ალგებრა - ეს მათემატიკური ლოგიკის ნაწილია, რომლის ყველა ელემენტის მნიშვნელობა (ფუნქციებისა და არგუმენტების) განსაზღვრულია ორელემენტიან სიმრავლეზე: 0 და 1.

ორგანზომილებიანი ლოგიკის საფუძვლები შეიმუშავა ინგლისელმა მათემატიკოსმა ჯორჯ ბულმა მე -19 საუკუნის შუა წლებში ტექნიკური სისტემების ანალიზისათვის ლოგიკის ალგებრის გამოყენების შესაძლებლობაზე პირველმა პ. ს. ერენფესტმა მიუთითა (1910წ.), ხოლო 1838 წელს კ. შენონმა გამოიყენა ბულის ალგებრა რელეური სქემების გათვლისათვის. ბულის ალგებრის ოპერაციებს ჩვეულებრივ ბულის ოპერაციებს უწოდებენ.

2.1. ბულის ლოგიკის ოპერაციები.

1. ბულის ლოგიკის ოპერაცია მოსახერხებელია შემოვიტანოთ ცნებით „სიმრავლე“. ს ი მ რ ა ვ ლ ი ს ქვეშ ჩვენ ვიგულისხმებთ ნებისმიერი ბუნების ელემენტთა ერთობლიობას, რომელიც თვლას ექვემდებარება. თვლა შეიძლება უსასრულოდ დიდხანს გრძელდებოდეს - ან დასაწყისშივე, დასრულდეს ელემენტთა არარსებობის გამო. მიუხედავად ამისა, თ ვ ლ ი ს პ რ ო ც ე დ უ რ ა, რომლის აზ-

რიც იმაში მდგომარეობს, რომ დაამყაროს ურთიერთ ცალსახა შესატყვისობა ელემენტებსა და ნატურალური რიგის რიცხვებს შორის, სიმრავლეებისათვის განმსაზღვრელია. არა თვლად სიმრავლეებთან დაკავშირებული საკითხები ჩვენს მიერ განხილული არ იქნება.

ვთქვათ, მოცემულია საგანთა რაღაც ერთობლიობა, რომელიც გადათვლის შემდეგ შეიძლება ასე ავღნიშნოთ:

$$V = \{1, 2, \ldots .11\}.$$

შემდეგ ვიგულისხმოთ, რომ საგნების ნაწილს, კერძოდ კი: 1, 2, 4 და 6-ს, აქვთ მრგვალი ფორმა, ხოლო ნაწილი - 2, 3, 4, 8 და 9 - თეთრადაა შეღებილი. ამ შემთხვევაში ამბობენ, რომ V სიმრავლეს მრგვალი და თეთრი საგნების ორი ქ ვ ე ს ი მ რ ა ვ ლ ე გააჩნია:

$$A = \{1, 2, 4, 6\} \quad \text{და} \quad B = \{2, 3, 4, 8, 9\}.$$

შეიძლება სხვაგვარად ვთქვათ: საწყის სიმრავლეს დავარქვათ ფ უ ნ დ ა მ ე ნ ტ უ რ ი, ხოლო A და B ქვესიმრავლეებს - უბრალოდ სიმრავლეები.

შედეგად მივიღებთ ელემენტთა ოთხ კლასს:

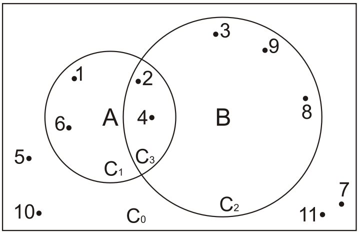
$C_0 = \{6, 7, 10, 11\}$ - ელემენტები, რომლებსაც არ გააჩნიათ ზემოთმოყვანილი თვისებებიდან არც ერთი;

$C_1 = \{1, 6\}$ - ელემენტები, რომლებსაც ახასიათებთ მხოლოდ A თვისება (იყვნენ მრგვალები);

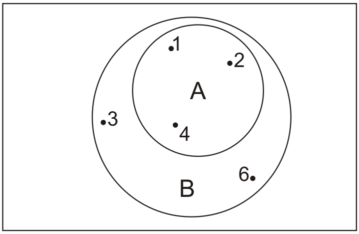
$C_2 = \{3, 8, 9\}$ - ელემენტები, რომლებსაც ახასიათებთ მხოლოდ B თვისება (იყვნენ თეთრები);

$C_3 = \{2, 4\}$ - ელემენტები, რომლებსაც ერთდროულად ახასიათებთ ზემოთ მოყვანილი ორივე თვისება.

ეს კლასები გამოსახულია ეილერ-ვენის გრაფიკულ დიგრამაზე (ნახ. 2.1, 2.2):



ნახ. 2.1.



ნახ. 2.2.

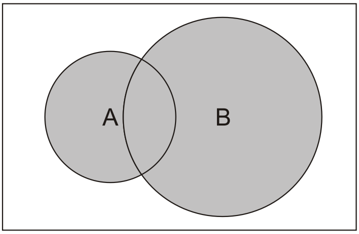
ხშირად დიაგრამებს არ გააჩნიათ განზოგადების მთელი სირ-თულე, მაგალითად ნახ.2.2-ზე გამოსახულს. მასში A სიმრავლე უკვე მთლიანადაა ჩართული B -ში. ასეთი შემთხვევისათვის იყენებენ ჩა-რთვის (მიკუთვნების) სპეციალურ სიმბოლოს ($\subset$) :

$$A \subset B = \{1, 2, 4\} \subset \{1, 2, 3, 4, 6\}.$$

თუ ერთდროულად სრულდება (კმაყოფილება) ორი პირობა: $A \subset B$ და $B \subset A$, მაშინ $A = B$. ამ შემთხვევაში ამბობენ, რომ A და B სიმრავლეები ს რ უ ლ ე ბ ი თ ე ქ ვ ი ვ ა ლ ე ნ ტ უ რ ე ბ ი ა.

იმის შემდეგ, როცა განსაზღვრულია ელემენტთა ოთხი კლასი და მოცემულია საჭირო ცნობები ეილერ-ვენის დიაგრამების შესახებ, შემოვიტანოთ ოპერაციები სიმრავლეებზე. ჯერ განვიხილით გა-ერთიანების ოპერაცია.

2. $A = \{1, 2, 4, 6\}$ და $B = \{2, 3, 4, 8, 9\}$ სიმრავლეთა გ ა ე რ - თ ი ა ნ ე ბ ა ვუწოდოთ სიმრავლეს $A \cup B = \{1, 2, 3, 4, 6, 8, 9\}$, სა-დაც $\cup$- სიმრავლეთა გაერთიანების სიმბოლოა. ამრიგად, გაერთია-ნებით მოცულია ელემენტთა სამი კლასი C_1, C_2 და C_3 რომლებიც ნახ.2.3-ზე გამოსახულ დიგრამაზე დაშტრიხულია.



ნახ.2.3.

ლოგიკურად ორი სიმრავლის გაერთიანების ოპერაცია შეიძლება ასე დავახასიათოთ: x ელემენტი ე კ უ თ ვ ნ ი ს A სიმრავლეს ან B სიმრავლეს. ამასთან კავშირი „ან“ ერთდროულად ნიშნავს „და“ კავშირსაც. x ელემენტის A სიმრავლეზე მიკუთვნების ფაქტი ასე აღინიშნება $x \in A$. ამიტომ ის, რომ x ეკუთვნის A ან/და B-ს, გამოისახება ფორმულით: $x \in A \cup B = (X \in A) \vee (X \in B)$, სადაც $\vee$ ლოგიკური „ან“ კავშირის სიმბოლოა, რომელსაც დ ი ზ ი უ ნ ქ ც ი ა ეწოდება.

ლოგიკის თვალსაზრისით, ერთი x ს ა გ ნ ო ბ რ ი ვ ი ცვლადის ნაცვლად მოხერხებულია ორი ლ ო გ ი კ უ რ ი x_1 და x_2 ცვლადის შემოტანა. x_1 და x_2 განსაზღვრის არე იქნება უკვე არა ნატურალური მწკრივის რიცხვები, არამედ მხოლოდ ორი ლოგიკური მნიშვნელობა: 1 ჭ ე შ მ ა რ ი ტ ი მნიშვნელობისათვის და 0 - ა რ ა - ჭ ე შ მ ა რ ი ტ ი ს ა თ ვ ი ს.

დავუშვათ $x = 7$. ვინაიდან ეს რიცხვი არ მიეკუთვნება არც A სიმრავლეს და არც B სიმრავლეს, ამიტომ ცვლადების ლოგიკური მნიშვნელობა იქნება: $x_1 = 0$, $x_2 = 0$. ცვლადების ეს კომბინაცია პასუხობს C_0 კლასს. ახლა დაუშვათ, რომ არჩეულია რიცხვი 4. ის შედის როგორც A -ში ისე B -ში. გამომდინარე, $x_1 = 1$, $x_2 = 1$, რაც შეესაბამება C_3 კლასს. არსებობს კიდევ ორი ვარიანტი. მაგალითად, რიცხვისათვის $x = 6$ გვაქვს $x_1 = 1$, $x_2 = 0$, ხოლო $x = 8$ - ისათვის - მნიშვნელობები $x_1 = 0$, $x_2 = 1$, რომლებიც პასუხობენ C_1, C_2 კლასებს.

x_1 და x_2 ცვლადები განსაზღვრავენ რაღაც ლ ო გ ი კ უ რ ფ უ ნ ქ ც ი ა ს $y = f(x_1, x_2)$, რომელიც დ ი ზ ი უ ნ ქ ც ი ი ს შემთხვევაში შეიძლება ჩაიწეროს როგორც პ რ ო პ ო ზ ი ც ი უ ლ ი კ ა ვ შ ი რ ი : y = $x_1 \vee x_2$.

ადვილი შესამჩნევია, რომ რიცხვი 7 არ შედის $A \cup B$ გაერთიანებულ სიმრავლეში, ამიტომ $x_1 = 0$, $x_2 = 0$ −ის დროს ლოგიკური y ფუნქციის მნიშვნელობა ნულის ტოლია. როცა ხდება 4, 6 ან 8 რიცხვების ამორჩევა, მაშინ ყველა მათგანი აუცილებლად მოხვდება დიაგრამის დაშტრიხულ ზონაში, გამომდინარე, ამ მნიშვნელობების დროს y ფუნქცია ერთის ტოლია. ყველაფერი ეს მოსახერხებელია ცხრილის სახით გავაფორმოთ, რომელსაც ჭ ე შ მ ა რ ი ტ ე ბ ი ს ც ხ რ ი ლ ი ჰქვია (ცხრილი 2.1).

ცხრილი 2.1.

x_1	x_2	$y = x_1 \vee x_2$
0	0	0
1	0	1
0	1	1
1	1	1

ჭეშმარიტების ცხრილსა და ეილერ-ვენის დიაგრამებს შორის ურთიერთცალსახა შესაბამისობაა. ამიტომ y-ისთვის ერთიანების რიცხვი ყოველთვის დაემთხვევა დიაგრამაზე დაშტრიხული ზონების რიცხვს x_1 და x_2 არგუმენტების ოთხი კომბინაცია პასუხობს C_i ოთხ ზონას. გარდა ამისა, ადვილად დასათვლელია, რომ y ფუნქციისათვის ნულებისა და ერთიანების კომბინაციათა რიცხვი 16 -ის

ტოლია, ამიტომ ორ სიმრავლეზე შესაძლო ოპერაციათა საერთო რიცხვიც ამ რიცხვის (16-ის) ტოლია.

3. A და B სიმრავლეთა კ ვ ე თ ა ეწოდება $A \cap B$ სიმრავლეს, რომელიც შეიცავს A - დან და B - დან იმ ელემენტებს, რომლებიც ერთდროულად შედიან ორივე სიმრავლეში. ჩვენი რიცხვითი მაგალითისათვის გვექნება:

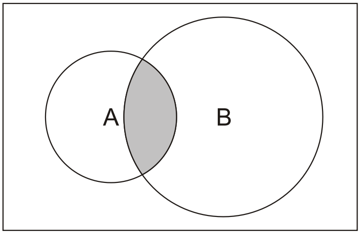
$$A \cap B = \{1, 2\, 4, 6\} \cap \{2, 3, 4, 8, 9\} = \{2,4\} = C_3.$$

ეილერ-ვენის დიაგრამა კვეთისათვის გამოსახულია ნახ. 2.4-ზე.

ის, რომ x ერთდროულად ეკუთვნის ორ სიმრავლეს A და B -ს, შეიძლება წარმოვადგინოთ გამოსახულებით:

$$x \in A \cap B = (x \in A) \wedge (x \in B),$$

სადაც $\wedge$ -ლოგიკური „და“ კავშირის სიმბოლოა, რომელსაც კ ო ნ ი-უ ნ ქ ც ი ა ჰქვია.



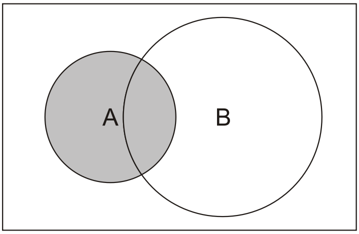
ნახ.2.4.

თუ დიზიუნქციისათვის აგებულ ჭეშმარიტების ცხრილში (ცხრ. 2.1) ყველა ნულს ერთიანებით შევცვლით, ხოლო ყველა ერთიანებს ნულებით, მაშინ მივიღებთ ცხრილ 2.2-ს. ეს ფაქტი განსაზღვრავს კონიუნქციისა და დიზიუნქციის ურთიერთ ო რ ა დ ო ბ ა ს. ნებისმიერი ლოგიკური ოპერაციისათვის შეიძლება ორადის პოვნა.

ცხრილი 2.2.

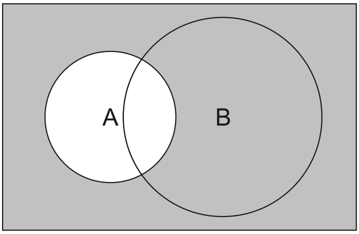
x_1	x_2	$y = x_1 \wedge x_2$
0	0	0
1	0	0
0	1	0
1	1	1

წარმოვიდგინოთ ოპერაცია, რომლის შედეგად დაშტრიხული აღმოჩნდებიან A სიმრავლის შემქმნელი C_1 , და C_3 ზონები (ნახ.2.5).



ნახ. 2.5.

შემდეგ კიდევ ერთი ოპერაცია, რომელიც „მიიტაცებს" ორ სხვა C_0 და C_2 ზონას, რომლებიც $\bar{A}$ (ნახ.2.6). თუ ორივე დიაგრამაზე გავა-



ნახ.2.6.

ერთიანებთ დაშტრიხულ სიმრავლეს 1; A და $\bar{A}$ კვეთა კი გვაძლევს ცარიელ 0 სიმრავლეს, რომელშიც არც ერთი ელემენტი არ შედის:

$$A \cup \bar{A} = 1, \quad A \cap \bar{A} = 0.$$

ანალოგიური ტოლობები სრულდებიან იმ ლოგიკური ფუნქციებისთვისაც, რომლებსაც აქვთ შესაბამისი დასახელებები:

$y = x \vee \bar{x} = 1$ - გამეორება (მესამის გამორიცხვა),

$y = x \wedge \bar{x} = 0$ - წინააღმდეგობა.

გ ა მ ე ო რ ე ბ ა - ეს ყოველთვის ჭეშმარიტი ლოგიკური გამოსახულებაა, რა მნიშვნელობაც არ უნდა მიიღოს ამ დროს x ცვლადმა. წ ი ნ ა ა ღ მ დ ე გ ო ბ ა, პირიქით, ყოველთვის მცდარი გამოსახულებაა.

$\bar{A}$ - სიმრავლე ა ვ ს ე ბ ს A სიმრავლეს ფუნდამეტურ V სიმრავლემდე (ან 1); აქედან დასახელება: დ ა მ ა ტ ე ბ ი თ ი $\bar{A}$ სიმრავლე, ან დ ა მ ა ტ ე ბ ა როგორც ოპერაცია. x ლოგიკური ცვლადისათვის დამატებას, ანუ $\bar{x}$ (არა x), ლოგიკაში ყველაზე ხშირად უწოდებენ x-ის უ ა რ ყ ო ფ ა ს. კვეთისა და დამატების ოპერაციის შემოტანის შემდეგ C_i ყველა ოთხი ზონა ეილერ-ვენის დიაგრამაზე შეიძლება შემდეგი სახით გამოვსახოთ:

$$C_0 = \bar{A} \cap \bar{B}, \ \ C_1 = A \cap \bar{B}, \ \ C_2 = \bar{A} \cap B, \ \ C_3 = A \cap B.$$

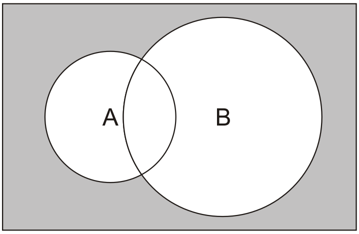
C_i - ს შესაბამისი ზონების გაერთიანების შემდეგ შეიძლება წარმოვიდგინოთ ნებისმიერი მრავლობითი ოპერაცია, მათ რიცხვში თავად გაერთიანებაც კი:

$$A \cup B = (A \cap \bar{B}) \cup (\bar{A} \cap B) \cup (A \cap B).$$

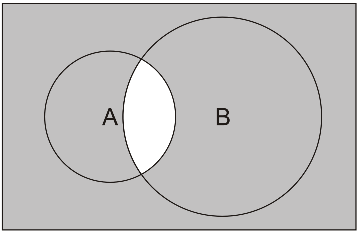
ყველაფერი ეს ლოგიკაზეც ვრცელდება:

$$y = x_1 \vee x_2 = (x_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2) \vee (x_1 \wedge x_2).$$

4. ნახ.2.7-ზე და ნახ.2.8-ზე მოყვანილია ორი ახალი ოპერაციის დიაგრამა, რომლებსაც შესაბამისად პ ი რ ს ი ს ი ს ა რ ს და შ ე ფ ე რ ი შ ტ რ ი ხ ს უწოდებენ. ეს დიაგრამები ავსებენ გაერთიანებისა და კვეთის ოპერაციებს ფუნდამენტურ V სიმრავლემდე.



ნახ.2.7.



ნახ.2.8.

ლოგიკური ფორმულების ენაზე ეს ფაქტი შემდეგი სახით გამოისახება:

პირსის ისარისთვის -

$$(x_1 \vee x_2) \vee (x_1 \downarrow x_2) = 1, \ \ (x_1 \vee x_2) \wedge (x_1 \downarrow x_2) = 0;$$

შეფერი შტრიხისათვის -

$$(x_1 \vee x_2) \vee (x_1 \mid x_2) = 1, \ \ (x_1 \vee x_2) \wedge (x_1 \mid x_2) = 0.$$

ამ ოპერაციების ჭეშმარიტების ცხრილებიდან (ცხრილი 2.3 და ცხრილი 2.4) ჩანს, რომ

$$y = x_1 \downarrow x_2 = \overline{x_1 \vee x_2} = \bar{x}_1 \wedge \bar{x}_2 = (x_1 \vee \bar{x}_2) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_2),$$

$$y = x_1 \mid x_2 = \overline{x_1 \wedge x_2} = \bar{x}_1 \vee \bar{x}_2 = (x_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2) \vee (\bar{x}_1 \wedge \bar{x}_2).$$

ცხრილი 2.3

$\boldsymbol{x_1}$	$\boldsymbol{x_2}$	$\boldsymbol{y = x_1 \downarrow x_2}$
0	0	1
1	0	0
0	1	0
1	1	0

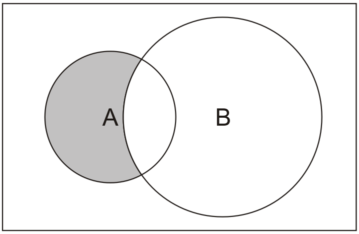
ცხრილი 2.4

x_1	x_2	$y = x_1 \mid x_2$
0	0	1
1	0	1
0	1	1
1	1	0

A და B სიმრავლეებს შორის ს ხ ვ ა ო ბ ა ეწოდება A სიმრავლის იმ ელემენტთა ერთობლიობას, რომლებიც არ შევიდნენ B სიმრავლეში:

$$A - B = \{1,2,4,6\} - \{2,3,4,8,9\} = \{1,6\} = C_1.$$

მისთვის ეილერ-ვენის დიაგრამა მოყვანილია ნახ. 2.9 -ზე .



ნახ. 2.9.

სხვაობის დამატებას ი მ პ ლ ი კ ა ც ი ა წარმოადგენს. ჭეშმარიტების ცხრილები სხვაობისა და იმპლიკაციისათვის წარმოდგენილია ცხრილ 2.5-ში და ცხრილ 2.6-ში.

ცხრილი 2.5

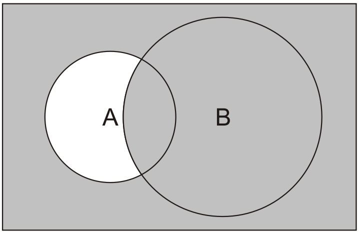
x_1	x_2	$y = x_1 - x_2$
0	0	0
1	0	1
0	1	0
1	1	0

ცხრილი 2.6

x_1	x_2	$y = x_1 \to x_2$
0	0	1
1	0	0
0	1	1
1	1	1

$$y = x_1 \to x_2 = \overline{x_1 - x_2} = \bar{x}_1 \vee x_2 = (x_1 \wedge \bar{x}_2) \vee (\bar{x}_1 \wedge x_2) \vee (\bar{x}_1 \wedge \bar{x}_2),$$
$$(x_1 - x_2) \vee (x_1 \to x_2) = 1, \ \ (x_1 - x_2) \wedge (x_1 \to x_2) = 0.$$

იმპლიკაციისათვის ეილერ-ვენის დიაგრამაზე (ნახ.2.10) ნაჩვენებია A სიმრავლის ნ ა წ ი ლ ო ბ რ ი ვ ი ჩართვა B სიმრავლეში, რომელიც უნდა განვასხვაოთ ს რ უ ლ ი ჩართვისაგან (ნახ. 2.2).



ნახ.2.10.

თუ ვამტკიცებთ, რომ „ A სიმრავლის ელემენტები ჩართული არიან B სიმრავლეში“, მაშინ C_3 ზონა აუცილებლად უნდა იყოს დაშტრიხული, ვინაიდან ის ჭეშმარიტებას შეესაბამება, ხოლო C_1 ზონა ასეთივე აუცილებლობით უნდა იყოს თეთრად დარჩენილი, რადგანაც მას პასუხობს პირდაპირ საწინააღმდეგო მტკიცება. $\bar{A}$ -ში მოთავსებული C_0 და C_1 ზონების შესახებ შევნიშნოთ შემდეგი: ჩვენ არა გვაქვს უფლება ისინი თეთრად დავტოვოთ, ვინაიდან ისინი პირდაპირ არ ეწინააღმდეგებიან პირველ მტკიცებას, მაგრამ, ვინაიდან ლოგიკა ო რ ო ბ ი თ ი ა, ჩვენ ვალდებულები ვართ $\bar{A}$ -ში მოხვედრილი ზონები დავშტრიხოთ. ს ა მ ო ბ ი თ ლოგიკაში ეს ზონები სხვაგვარად უნდა იყვნენ დაშტრიხულნი, ხოლო იმპლიკაციისთვის ჭეშმარიტების ცხრილში (ცხრილი 2.6) პირველ და მესამე სტრიქონებში y-ისათვის უნდა იდგეს არა 1, არამედ ½ , რაც გ ა ნ უ ს ა ზ ღ ვ რ ე ლ მდგომარეობას შეესაბამება. მაგრამ სამობითი და სხვა ა რ ა კ ლ ა ს ი კ უ რ ი ლოგიკები ჩვენს მიერ განხილულნი არ იქნებიან. ეხლა მხოლოდ შევნიშნოთ, რომ კ ლ ა ს ი კ უ რ ო რ ო ბ ი თ ლ ო გ ი კ ა შ ი იმპლიკაცია გადმოიცემა სიტყვებით „თუ A, მაშინ B“.

5. კიდევ დაგვრჩა მოვიტანოთ ორი ურთიერთშემავსებელი ოპერაცია - სიმეტრიული სხვაობა და ექვივალენტობა:

A და B ორი სიმრავლის ს ი მ ე ტ რ ი უ ლ ი ს ხ ვ ა ო ბ ა არის ორი სხვაობის გაერთიანება (ჯამი ორის მოდულით):

$A + B = (A - B) \cup (B - A) = \{1, 3, 6, 8, 9\} = C_1 \cup C_2 = \{1, 6\} \cup \{3, 8, 9\}$.

ე ქ ვ ი ვ ა ლ ე ნ ტ ო ბ ა განისაზღვრება A და B სიმრავლეთა იმ ელემენტებით, რომლებიც მათთვის საერთო არიან. ოღონდ ელემენტები, რომლებიც არ შედიან არც A-ში და არც B-ში, ასევე ექვივალენტურად ითვლებიან:

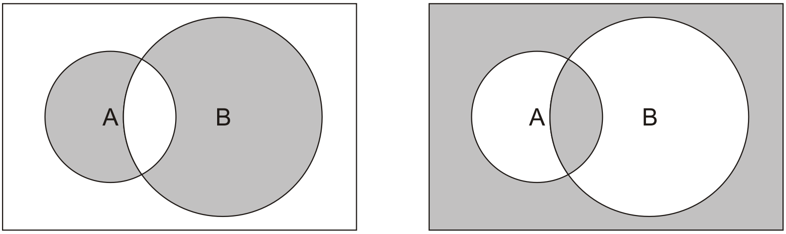
$$A \sim B = (A \cap B) \cup (\bar{A} \cap \bar{B}) = C_0 \cup C_3 = \{2, 4, 5, 7, 10, 11\}.$$

ნახ.2.11-ზე და ნახ.2.12-ზე ნაჩვენებია ეილენ-ვენის დიაგრამათა დაშტრიხვა, ხოლო ცხრილი 7 და ცხრილი 8 წარმოადგენენ შესაბამისი ოპერაციების ჭეშმარიტების ცხრილებს. ოპერაციათა შემავსებლობიდან გამომდინარეობს შემდეგი დამოკიდულებები:

$$(x_1 \oplus x_2) \vee (x_1 \sim x_2) = 1, \qquad (x_1 \oplus x_2) \wedge (x_1 \sim x_2) = 0,$$

$$y = x_1 \sim x_2 = \overline{x_1 \oplus x_2} = (x_1 \wedge x_2) \vee (\bar{x}_1 \wedge \bar{x}_2) =$$

$$= (x_1 \wedge \bar{x}_2) \wedge (\bar{x}_1 \wedge x_2) \wedge (\bar{x}_1 \wedge \bar{x}_2).$$



ნახ. 2.11.   ნახ. 2.12.

ცხრილი 2.7

x_1	x_2	$y = x_1 \oplus x_2$
0	0	0
1	0	1
0	1	1
1	1	0

ცხრილი 2.8

x_1	x_2	$y = x_1 \sim x_2$
0	0	1
1	0	0
0	1	0
1	1	1

შევნიშნოთ, რომ სიმეტრიულ სხვაობას გააჩნია რამდენიმე სახელი: მკაცრი დიზუნქცია, გამომრიცხავი ალტერნატივა, ჯამი ორის მოდულით. ეს ოპერაცია შეიძლება გადმოცემული იყოს შემდეგი სიტყვებით - „ან A, ან B“, ანუ ესაა ლოგიკური კავშირი „ან“, მაგრამ მასში ჩართული „და“ კავშირის გარეშე.

ფართოდ გავრცელებული ლოგიკური ფუნქციების ჭეშმარიტების ცხრილი მოყვანილია ცხრ.2.9-ში, ხოლო ამ ცხრილში ოპერაციათა სრული აღნიშვნა, დასახელება, წაკითხვის ხერხი კი - ცხრ. 2.10-ში.

ცხრილი 2.9

x_1	x_2	უარყოფა $\bar{x}_1$	კონიუნქცია $x_1 \wedge x_2$	დიზიუნქცია $x_1 \vee x_2$	იმპლიკაცია $x_1 \rightarrow x_2$	ექვივალენტობა $x_1 \sim x_2$	ჯამი 2-ის მოდულით $x_1 \oplus x_2$	შეფერის შტრიხი $x_1 \mid x_2$	პირსის ისარი $x_1 \downarrow x_2$
0	0	1	0	0	1	1	0	1	1
0	1	1	0	1	1	0	1	1	0
1	0	0	0	1	0	0	1	1	0
1	1	0	1	1	1	1	0	0	0

საერთოდ შესაბამისი ფუნქციის ცხრილი შეიძლება აგებულ იქნას ნებისმიერი ფორმულისათვის მასში ყველა ნაკრებისათვის ცვლადთა მნიშვნელობების ჩასმით. ფუნქციის (და გამომდინარე, ფორმულის) ცხრილური წარმოდგენა ერთადერთია. მაგრამ ერთი და იგივე ფუნქცია შეიძლება ჩაწერილ იქნას სხვადასხვა გამოსახულების (ფორმულის) სახით. მაგალითად, შეფერი შტრიხ ფუნქცია შეიძლება ჩაწერილ იქნას როგორც $y = f(x_1, x_2) = \bar{x}_1 \vee \bar{x}_2$, ისევე როგორც $y = f(x_1, x_2) = \overline{x_1 \wedge x_2}.$ ფორმულებს, რომლებსაც წარმოადგენენ ერთი და იგივე ფუნქციას, უწოდებენ ე ქ ვ ი ვ ა ლ ე ნ ტ უ რ ან ტ ო ლ ფ ა ს ო ვ ა ნ ფორმულებს.

ფორმულათა ჩასაწერად სავალდებულო არ არის ყველა ფუნქციის გამოყენება. მაგალითად, შეფერი შტრიხი შეგვიძლია გამოვსახოთ ცვლადების უარყოფათა დიზიუნქციით: $\overline{x_1 \wedge x_2} = \bar{x}_1 \vee \bar{x}_2$. არსებობს ლოგიკურ ფუნქციათა ისეთი ნაკრებები, რომელთა დახმარებითაც შეიძლება გამოსახულ იქნან ნებისმიერი სხვა ლოგიკური ფუნქციები. ასეთ ნაკრებებს ფ უ ნ ქ ც ი ო ნ ა ლ უ რ ა დ ს რ უ ლ ე ბ ს ან ბ ა ზ ი ს უ რ ე ბ ს უწოდებენ.

ცხრილი 2.10

ოპერა-ციის აღნიშვნა	ჭეშმარი-ტების ცხრილი	x_1	x_2	ლოგიკური ფუნქციები	
		0 0 1 1	0 1 0 1	ოპერაციის დასახელება	როგორ წავიკითხოთ
$x_1 \wedge x_2$ $x_1 \& x_2$ $x_1 \cdot x_2$ $x_1 x_2$	0 0 0 1			კონიუნქცია, ლოგი-კური გამრავლება, ლოგიკური „და“, თანხვედრის ფუნ-ქცია	x_1 და x_2
$x_1 \vee x_2$	0 1 1 1			დიზიუნქცია, ლო-გიკური ჯამი, ლო-გიკური „ან“, განცა-ლკევების ფუნქცია	x_1 ან x_2
$x_1 \rightarrow x_2$ $x_1 \supset x_2$	1 1 0 1			იმპლიკაცია	თუ x_1, მაშინ x_2; x_1 იწვევს x_2-ს.
$x_1 \sim x_2$ $x_1 \leftrightarrow x_2$ $x_1 \equiv x_2$	1 0 0 1			ექვივალენტობა, ტოლფასოვნების ფუნქცია	x_1 მაშინაა და მხოლოდ მაშინ, როცა x_2 არის; x_1 ექვივალენ-ტურია x_2 -ის.
$x_1 \oplus x_2$	0 1 1 0			ჯამი 2-ის მოდუ-ლით,ექვივალენტო ბის უარყოფა, არა-ტოლფასოვნების ფუნქცია	ან x_1 ან x_2; x_1 არაა ექვივალენტური x_2 -ის.
$x_1 \vert x_2$	1 1 1 0			შეფერი შტრიხი, კონიუნქციის უარყოფა, ანტიკონიუნქცია	x_1 და x_2 არა-თავსებადებია; არაჭეშმარიტია, რომ x_1 და x_2.

$x_1 \downarrow x_2$	1 0 0 0	პირსის ისარი, დიზუნქციის უარყოფა, ანტიდიზიუნქცია	არც x_1, არც x_2.

ისევე როგორც შეფერი შტრიხი, კონიუნქციის, დიზუნქციის და უარყოფის საშუალებით შეიძლება გამოსახულ იქნან ორი ცვლადის სხვა ლოგიკური ფუნქციებიც:

$$x_1 \to x_2 = \bar{x}_1 \vee x_2,$$
$$x_1 \sim x_2 = x_1 \wedge x_2 \vee \bar{x}_1 \wedge \bar{x}_2,$$
$$x_1 \oplus x_2 = x_1 \wedge \bar{x}_2 \vee \bar{x}_1 \wedge x_2,$$
$$x_1 \downarrow x_2 = \overline{x_1 \vee x_2} = \bar{x}_1 \wedge \bar{x}_2.$$

ფორმულებს, რომლებიც შეიცავენ მხოლოდ ამ სამი ოპერაციის ნიშნებს (აგრეთვე ცვლადებსა და ფრჩხილებს), უწოდებენ ბულის ფორმულებს, ხოლო ლოგიკურ ფუნქციათა სიმრავლე, რომელზეც განსაზღვრულია ეს სამი ოპერაცია, ბულის ალგებრად იწოდება. სამართლიანია თეორემა, რომელიც ამტკიცებს, რომ ნებისმიერი ლოგიკური ფუნქცია შეიძლება წარმოდგენილ იქნას ბულის ფორმულის სახით. ამიტომ კონიუნქციის, დიზუნქციის და უარყოფის ნაკრებები ქმნიან ბაზისს.

კონიუნქციისა და უარყოფის, დიზიუნქციისა და უარყოფის ნაკრებები აგრეთვე ბაზისურებია. უარყოფის გამოყენებით კონიუნქცია შეიძლება გამოვსახოთ დიზიუნქციით და პირიქით:

$$x_1 \vee x_2 = \overline{\overline{x_1 \vee x_2}} = \overline{\bar{x}_1 \wedge \bar{x}_2},$$
$$x_1 \wedge x_2 = \overline{\overline{x_1 \wedge x_2}} = \overline{\bar{x}_1 \vee \bar{x}_2}.$$

ბაზისი შეიძლება შედგებოდეს მხოლოდ ერთი ფუნქციისაგან. მაგალითად, შეფერი შტრიხი და პირსის ისარი აგრეთვე ქმნიან ბაზისებს, ვინაიდან ნებისმიერი ფუნქცია - უარყოფა, კონიუნქცია და დიზიუნქცია შეიძლება მათი საშუალებით იქნას გამოსახული:

$$\bar{x} = x | x = x \downarrow x,$$
$$x_1 \wedge x_2 = \overline{\overline{x_1 \wedge x_2}} = \overline{x_1 | x_2} = (x_1 | x_2) \mid (x_1 | x_2),$$

$$x_1 \vee x_2 = \overline{x_1 \downarrow x_2} = (x_1 \downarrow x_2) \downarrow (x_1 \downarrow x_2).$$

ლოგიკური გამოსახულების მნიშვნელობის გამოთვლისას ოპერაციათა პრიორიტეტი შემდეგია (დაქვეითების რიგით): 1) უარყოფა; 2) კონიუნქცია; 3) დიზიუნქცია და დიზუნქციის უარყოფა; 4) შედარების ოპერაციები (ტოლია, არაა ტოლი, მეტია, ნაკლებია, მეტია ან ტოლი, ნაკლებია ან ტოლი).

თუ არსებობს გამოთვლების რიგის შეცვლის აუცილებლობა, საჭიროა მრგვალი ფრჩხილების გამოყენება.

ბულის ფორმულათა ჩაწერისას კონიუნქციის ოპერაციის ნიშნებს არ წერენ, ისევე როგორც გამრავლების ნიშანს ალგებრაში. ანალოგიურად, რამდენიმე კონიუნქციისა და დიზიუნქციის მიმდევრობითი შესრულებისას, არ წერენ ფრჩხილებს; ასევე არ წერენ გარე ფრჩხილებს კონიუნქციაში:

$$\big(x_1 \wedge (x_2 \wedge x_3)\big) \vee \big((x_4 \vee x_5) \wedge x_6\big) = x_1 \wedge x_2 \wedge x_3 \vee (x_4 \vee x_5) \wedge x_6.$$

2.2. ბულის ლოგიკის კანონები.

1. ბულის ლოგიკის ძირითად კანონებს შორის ყველაზე ხშირად მოჰყავთ:

1) ი დ ე მ პ ო ტ ე ნ ტ უ რ ო ბ ი ს   კ ა ნ ო ნ ე ბ ი :

$$a = a \wedge a, \quad a = a \vee a;$$

2) კ ო მ უ ტ ა ტ ი უ რ ო ბ ი ს   კ ა ნ ო ნ ე ბ ი :

$$a \wedge b = b \wedge a, \quad a \vee b = b \vee a;$$

3) ა ს ო ც ი ა ტ ი უ რ ო ბ ი ს   კ ა ნ ო ნ ე ბ ი :

$$a \wedge (b \wedge c) = (a \wedge b) \wedge c, \quad a \vee (b \vee c) = (a \vee b) \vee c;$$

4) დისტრიბუტიულობის კანონები:

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c), \qquad a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c);$$

5) ნულისა და ერთის კანონები:

$$a \wedge \bar{a} = 0, \quad a \wedge 1 = a, \quad a \vee \bar{a} = 1, \quad a \vee 0 = a;$$

6) შთანთქმის კანონები:

$$a \vee (a \wedge b) = a, \quad a \wedge (a \vee b) = a;$$

7) დე მორგანის კანონები.

$$\overline{a \vee b} = \bar{a} \wedge \bar{b}, \quad \overline{a \wedge b} = \bar{a} \vee \bar{b};$$

8) შეწებების კანონები:

$$(a \vee \bar{b}) \wedge (a \vee b) = a, \quad (a \wedge \bar{b}) \vee (a \wedge b) = a;$$

ეს კანონები ერთმანეთზე არიან დამოკიდებულები. ასე, მაგალითად, იდემპოტენტურობის კანონი შეგვიძლია მივიღოთ შთანთქმის კანონიდან დისტრიბუტიულობის კანონის გამოყენებით:

$$a = a \vee (a \wedge b) = (a \vee a) \wedge (a \vee b) = \big(a \wedge (a \vee b)\big) \vee \big(a \wedge (a \vee b)\big) =$$
$$= a \vee a.$$

შთანთქმის კანონი შეიძლება გამოყვანილ იქნას ნულისა და ერთის კანონიდან:

$$a \vee (a \wedge b) = (a \wedge 1) \vee (a \wedge b) = a \wedge (1 \vee b) = a \wedge 1 = a.$$

დიზუნქციის მიმართ იდემპოტენტურობის კანონი უშუალოდ გამოდის ნულისა და ერთის კანონიდან:

$$a \vee a = (a \vee a) \wedge 1 = (a \vee a) \wedge (\bar{a} \vee a) = a \vee (a \wedge \bar{a}) = a \vee 0 = a.$$

ლოგიკური გამოსახულებების დამტკიცებისას ყოველთვის მხედველობაში უნდა გვქონდეს ორადობის პრინციპი. ასე, ზემოთ მოყვანილი ტოლობათა მწკრივი შთანთქმის კანონისათვის შეიძლება შემდეგი სახით იქნას წარმოდგენილი:

$$a \wedge (a \vee b) = (a \vee 0) \wedge (a \vee b) = a \vee (0 \wedge b) = a \vee 0 = a.$$

ამრიგად, კანონთა დამოუკიდებელ სისტე-მად შეგვიძლია ავირჩიოთ შემდეგი კანონები: კომუტატიუ-რობის, ასოციატიურობის, დისტრიბუტიულო- ბის, ნულისა და ერთის.

2. ლოგიკაში ფართოდ გამოიყენება ორი მიდგომა - ა ქ ს ი ო - მ ა ტ უ რ ი და კ ო ნ ს ტ რ უ ქ ც ი უ ლ ი. აქსიომატური დამტკიცებისას გამოიყენება ა ქ ს ი ო მ ა თ ა ხ ი ს ტ ი ს ი ს ტ ე მ ა, რომელიც, მაგალითად, შედგება მხოლოდ ზემოთმოყვანილი ოთხი კანონისაგან. ყველა დანარჩენი იგივეობა საჭიროა წარმოვადგინოთ ამ კანონებით. კონსტრუქციული დამტკიცებისას კი უნდა ვისარგებლოთ კ ო ნ ს ტ რ უ ქ ტ ი ვ თ ა ს ი ს ტ ე მ ი თ, რომლის მაგალითებია ეილერ-ვენის დიაგრამები და ჭეშმარიტების ცხრილები. ვაჩვენოთ განსხვავება ამ ორ მიდგომას შორის.

მარტივი იგივეობის $(a \wedge 0 = 0)$ დასამტკიცებლად აქსიომატური მიდგომის მომხრე მოიყვანს დაახლოებით ასეთ გარდაქმნათა მწკრივს:

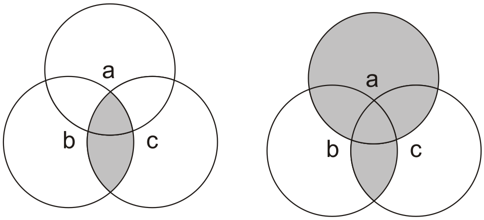
$$a \wedge 0 = a \wedge (a \wedge \bar{a}) = (a \wedge a) \wedge \bar{a} = \big((a \wedge a) \vee 0\big) \wedge \bar{a} =$$
$$= \big((a \wedge a) \vee (a \wedge \bar{a})\big) \wedge \bar{a} = \big(a \wedge (a \vee \bar{a})\big) \wedge \bar{a} = (a \wedge 1) \wedge \bar{a} = a \wedge \bar{a} = 0.$$

და ამას გააკეთებს მხოლოდ იმისათვის, რომ ფორმალურად გამოიყენოს მხოლოდ ზემოთ მოყვანილი აქსიომათა სისტემა. კონსტრუქტივისტისათვის კი საწყისი იგივეობა პრაქტიკულად არავითარ დამტკიცებას არ მოითხოვს.

სურათი საწინააღმდეგოზე იცვლება, მაგალითად, დისტრიბუტიულობის კანონთან მიმართებაში. აქსიომატიკი მოცემულ შემთხვევაში არავითარ მოქმედებას არ განახორციელებს, ხოლო კონსტრუქციული მიდგომის მოყვარული ვალდებულია გვიჩვენოს იგივეობის მარცხენა და მარჯვენა მხარეთა ექვივალენტობა:

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c).$$

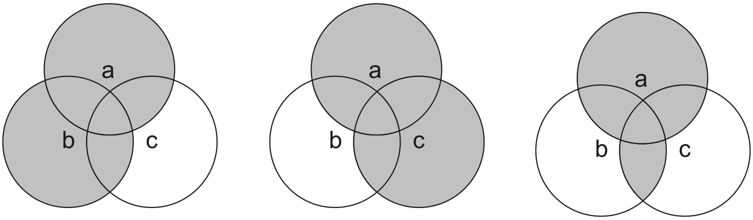
ჩავატაროთ დამტკიცება ეილერ-ვენის დიაგრამის დახმარებით. ამ მიზნით ავაგოთ ნახ.13-ზე გამოსახული ორი დიაგრამა, რომლებიც პასუხობენ იგივეობის მარცხენა მხარის ორ ოპერაციას.



$b \wedge c$ $\quad\quad\quad\quad$ $a \vee (b \wedge c)$

ნახ. 2.13.

ახლა ავაგოთ დისტრიბუტიულობის კანონში მარჯვენა ნაწილში მოყვანილი სამი ოპერაციის შესაბამისი კიდევ სამი დიაგრამა (ნახ.2.14).



$a \vee b$ $\quad\quad\quad\quad$ $a \vee c$ $\quad\quad\quad\quad$ $(a \vee b) \wedge (a \vee c)$

ნახ.2.14.

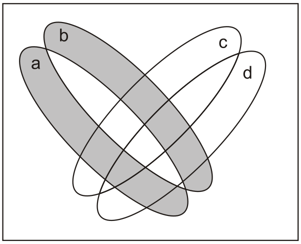
როგორც დიაგრამებიდან ჩანს, დისტრიბუტიულობის კანონის მარცხენა და მარჯვენა ნაწილების ლოგიკური ოპერაციების აგების შედეგები სრულად დაემთხვა ერთმანეთს.

3. დავამტკიცოთ ეილერ-ვენის დიაგრამის დახმარებით უფრო რთული იგივეობის სამართლიანობა:

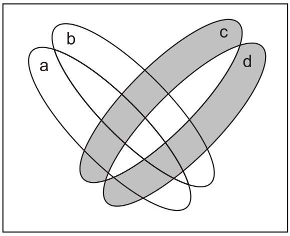
$$a \oplus b \oplus c \oplus d = \big((a \sim b) \to (c \oplus d)\big) \wedge \big((a \oplus b) | (c \oplus d)\big).$$

ნახ.2.15-ზე ნაჩვენებია მოცემული იგივეობის მარცხენა ნაწილში მოცემული ორი ოპერაცია - $a \oplus b$ და $c \oplus d$. საჭიროა აღინიშნოს, რომ ოთხი a, b , c და d ცვლადისათვის წრეების დახმარებით გამოსახული ეილერ-ვენის დიაგრამა არაა სრული, ვინაიდან ის შეიცავს

მხოლოდ 14 ზონას, ხოლო აუცილებელია 16. ამიტომ საწყის ზონებად არჩეულია ელიფსები.



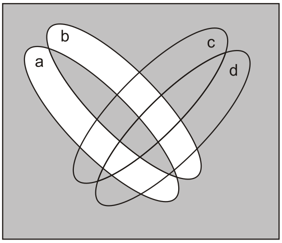
$a \oplus b$



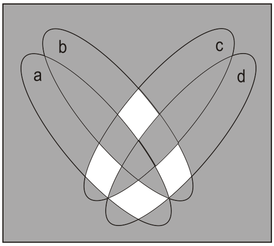
$c \oplus d$

ნახ.2.15.

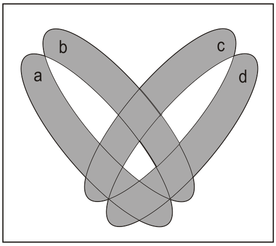
ნახ.2.16-ზე გამოსახულია ოთხი დიაგრამა, რომელიც შეესაბამება იგივეობის მარჯვენა ნაწილის ოპერაციებს; მათგან ბოლო შემა-



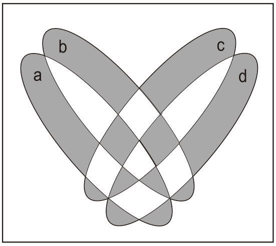
$a \sim b$



$(a \oplus b)|(c \oplus d)$



$(a \sim b) \to (c \oplus d)$



$\big((a \sim b) \to (c \oplus d)\big) \wedge \big((a \oplus b)|(c \oplus d)\big)$

ნახ. 2.16.

ჯამებელია. მაგრამ ზუსტად ასეთი შემაჯამებელი დიაგრამა მიიღება ორი პირველი დიაგრამის $mod(2)$ - ით შეკრების შედეგად: $(a \oplus b) \oplus (c \oplus d)$. ვინაიდან შემაჯამებელი დიაგრამის მარჯვენა და მარცხენა მხარეები ერთნაირია, ამიტომ ჩვენს მიერ მოყვანილი იგივეობა სამართლიანია.

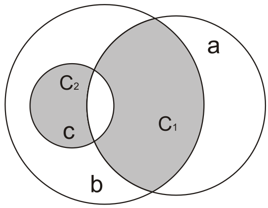
იგივეობის სამართლიანობაში შეიძლება ჭეშმარიტების ცხრილის მეშვეობით დავრწმუნდეთ (ცხრილი 2.11). ის გვიჩვენებს, რომ ნულებისა და ერთიანებისაგან შემდგარი მნიშვნელობათა ნაკრებები მარცხენა ნაწილისათვის (f_L) დაემთხვა მარჯვენა ნაწილის ნაკრებებს (f_R), ე. ი. საწყისი იგივეობა სწორია.

ცხრილი 2.11

a	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
b	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
c	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
d	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$f_1 = a \oplus b$	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
$f_2 = c \oplus d$	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
$f_L = f_1 \oplus f_2$	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0
$f_3 = a \sim b$	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
$f_4 = f_3 \to f_2$	0	1	1	0	1	1	1	1	1	1	1	1	0	1	1	0
$f_5 = f_1 \mid f_2$	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	1
$f_R = f_4 \wedge f_5$	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0

შეიძლება დავსვათ შებრუნებული ამოცანა, ანუ ცნობილი დიაგრამით ვიპოვოთ მისი შესატყვისი კომპაქტური ანალიზური გამოსახულება. ვთქვათ მოცემულია ნახ.2.17 გამოსახული დიაგრამა. ვიპოვოთ მისი შესატყვისი ანალიზური გამოსახულება. ამისათვის დაშტრიხული ზონები წარმოვიდგინოთ კონსტიტუენტების სახით $C_1 = a \wedge b \wedge \bar{c}$, $C_2 = \bar{a} \wedge b \wedge c$. საძიებელი გამოსახულება მიიღება ამ ორი კონსტიტუენტის გაერთიანებით:

$$x = C_1 \vee C_2 = b \wedge \big((a \wedge \bar{c}) \vee (\bar{a} \wedge c)\big) = b \wedge (a \oplus c).$$



ნახ. 2.17.

4. საჭიროა გავარკვიოთ, შესრულდება თუ არა ასოციატიურობის კანონი შეფერი შტრიხის მიმართ:

$$(a|b)|c = a|(b|c) \quad ?$$

აქ შეიძლება არ მივმართოთ ჭეშმარიტების ცხრილებს ან ეილერ-ვენის დიაგრამებს. საკმარისია ოპერაციებს შორის კავშირთა ცოდნა, რომ ვაჩვენოთ ამ კანონის მცდარობა:

$$a|(b|c) = 1 \oplus a(1 \oplus bc) = 1 \oplus a \oplus abc;$$
$$(a|b)|c = 1 \oplus (1 \oplus ab)c = 1 \oplus c \oplus abc.$$

კონსტრუქტივისტების პოზიცია იმაში მდგომარეობს, რომ ნებისმიერმა იგივეობამ მათემატიკურ ლოგიკაში მიიღოს თავისი დამაჯერებელი დასაბუთება, იქნება ის დისტრიბუტიულობის კანონი დიზიუნქციისათვის და კონიუნქციისათვის ან ასოციატიურობის კანონი ამ ოპერაციებისათვის - არაფერი არ უნდა იქნეს აღებული

„აქსიომის“ სახით, ანუ მტკიცებულებების დამტკიცების გარეშე. ანალოგიური კონსტრუქციული ხერხი შეიძლება გამოვიყენოთ მესამის გამორიცხვის წესის დასამტკიცებლად: $\big(a \wedge a(a \to b)\big) \to b = 1$.

დამტკიცება:

$$a(1 \oplus a \oplus ab) \to b = ab \to b = 1 \oplus ab \oplus abb = 1.$$

შემდეგ მაგალითში პოლინომიალური ფორმა უკვე აღარ გამოიყენება, ხოლო დამტკიცება მიმდინარეობს ბულის ოპერაციების საშუალებით. საჭიროა დავამტკიცოთ:

$$\big((a \oplus b) \to (a \vee b)\big) \wedge \big((a \vee b) \to (a \oplus b)\big) = a|b.$$

დატკიცება:

$$\big((a \oplus b) \to (a \vee b)\big) \wedge \big((a \vee b) \to (a \oplus b)\big) =$$

$$= \Big(\Big(\big(\bar{a} \vee \bar{b}\big) \wedge (a \vee b)\Big) \to (a \vee b)\Big) \wedge \Big(\big(\overline{a \vee b}\big) \vee \Big(\big(\bar{a} \vee \bar{b}\big) \wedge (a \vee b)\Big)\Big) =$$

$$= \Big((a \wedge b) \vee \big(\overline{a \vee b}\big) \vee (a \vee b)\Big) \wedge \Big(\big(\overline{a \vee b}\big) \vee \big(\bar{a} \vee \bar{b}\big)\Big) =$$

$$= 1 \wedge \Big(\big(\bar{a} \wedge \bar{b}\big) \vee \big(\bar{a} \vee \bar{b}\big)\Big) = \bar{a} \vee \bar{b} = a|b.$$

ხშირად გვხვდებიან შერეული ხასიათის მტკიცებები, მაგალითად, საჭიროა დავადგინოთ, რომ $\bar{A} \cup B = 1$, თუ $A \subset B$.

როცა გვხვდება მიკუთვნების სიმბოლო, უკეთესია ის გადავიტანოთ ტოლობაში. თუ A სრულადაა ჩართული B-ში, მაშინ ეილერ-ვენის დიაგრამის დახმარებით ადვილი შესამოწმებელია, რომ

$A \cap B = A$, ან $A \cup B = B$.

შემდეგ, ბოლო ტოლობისა და ბულის ლოგიკის აქსიომის გამოყენებით მივიღებთ:

$$\bar{A} \cup B = \bar{A} \cup (A \cup B) = (\bar{A} \cup A) \cup B = 1 \cup B = 1.$$

დავუშვათ, საჭიროა შემდეგი იგივეობის დამტკიცება:

$A \cap B = B$, ან $A \cup B = A$.

მოცემულ შემთხვევაში დამტკიცების ჩატარება შეიძლება, ყველაზე მცირე, ორი ხერხით.

პირველი ხერხი:

$$B = B \cup 0 = B \cup (A \cup \bar{A}) = (A \cup B) \cap (\bar{A} \cup B) = A \cap (\bar{A} \cup B) =$$

$$= (A \cap \bar{A}) \cup (A \cap B) = 0 \cup (A \cap B) = A \cap B.$$

მეორე ხერხი:

$$A \cap B = (A \cup B) \cap B = B.$$

დამტკიცების მეორე ხერხის დროს გამოყენებული იყო შთანთქმის კანონი, რომელიც არ შედის ადრე ჩამოყალიბებული აქსიომების სისტემაში, მაგრამ, მიუხედავად ამისა, მისი სამართლიანობა ადვილი დასადგენია. აქედან დასკვნა: თითოეული მტკიცება დამოკიდებულია იმ საშუალებებზე, რომლებსაც ჩვენ ვფლობთ.

დე მორგანის კანონი შეიძლება შემდეგი სახით დავამტკიცოთ, „გადავამრავლოთ“ იგივეობა მარცხნიდან და მარჯვნიდან $(a \vee b)$ -ზე მივიღებთ:

$$\left(\overline{a \vee b}\right) \wedge (a \vee b) = \left(\bar{a} \wedge \bar{b}\right) \wedge (a \vee b).$$

ვინაიდან $a \wedge \bar{a} = 0$, ამიტომ იგივეობის მარცხენა ნაწილი ნულის ტოლია. მარჯვენა ნაწილში ფრჩხილების გახსნით ვრწმუნდებით, რომ ისიც ნულის ტოლია.

ამ დამტკიცებათა ანალოგიით სრულებით მართებულად გვეჩვენება შთანთქმის კანონის ასეთი დამტკიცება:

$$a = a \wedge (a \vee b).$$

იგივეობის ორივე მხარეს „ვამრავლებთ“ $(a \vee b)$ -ზე:

$$a \wedge (a \vee b) = a \wedge (a \vee b) \wedge (a \vee b).$$

იდემპოტენტურობის კანონის გამოყენებით მივდივართ იგივეობამდე: $a \wedge (a \vee b) = a \wedge (a \vee b)$.

მაგრამ ზემოთ მოყვანილი დამტკიცება მცდარია, ვინაიდან ლოგიკაში ნებისმიერი „გამრავლება“ დაუშვებელია. დასაშვებია მხოლოდ ისეთი „გამრავლება“ და „შეკრება“, რომლებიც პასუხობენ ნულისა და ერთის კანონს, ანუ $a \wedge \bar{a} = 0$, $a \vee \bar{a} = 1$. ნათქვამის საილუსტრაციოდ ავიღოთ აშკარად მცდარი იგივეობა:

$$(a \wedge b) \vee (\bar{a} \wedge c) \vee (a \wedge c) = (a \wedge b) \vee (\bar{a} \wedge c).$$

შემდეგი სახით ვისარგებლოთ შთანთქმის კანონით:

$$a = a \vee (a \wedge c).$$

საწყის გამოსახულებასთან მისი „შეკრებით“ მივიღებთ:

$$(a \wedge b) \vee (\bar{a} \wedge c) \vee (a \wedge c) \vee a = (a \wedge b) \vee (\bar{a} \wedge c) \vee (a \wedge c) \vee a,$$

რამაც უნდა დაგვიმტკიცოს საწყისი გამოსახულების სამართლიანობა. მაგრამ ჭეშმარიტების ცხრილების გამოყენებით (ცხრ. 2.12) ადვილად ვადგენთ, რომ ეს ასე არაა. ორი ბოლო სტრიქონი, რომლებიც შეესაბამებიან საწყისი გამოსახულების მარჯვენა (f_R) და მარცხენა (f_L) ნაწილებს, ერთმანეთისაგან განსხვავდებიან. გამომდინარე, იგივეობა შეცდომითაა ჩაწერილი. ჭეშმარიტი იგივეობა შემდეგია:

$$(a \wedge b) \vee (\bar{a} \wedge c) \vee (b \wedge c) = (a \wedge b) \vee (\bar{a} \wedge c).$$

ცხრილი 2.12.

a	0	1	0	1	0	1	0	1
b	0	0	1	1	0	0	1	1
c	0	0	0	0	1	1	1	1
$f_1 = a \wedge b$	0	0	0	1	0	0	0	1
$f_2 = \bar{a} \wedge c$	0	0	0	0	1	0	1	0
$f_3 = a \wedge c$	0	0	0	0	0	1	0	1
$f_R = f_1 \vee f_2$	0	0	0	1	1	0	1	1
$f_L = f_1 \vee f_2 \vee f_3$	0	0	0	1	1	1	1	1

2.3. ბულის ფუნქციათა წარმოდგენის ფორმები.

ნებისმიერი ბულის ფუნქცია $y = f(a, b)$ შეიძლება წარმოვიდგინოთ როგორც არეთა (ზონების) რაღაც კომბინაცია:

$$C_0 = \bar{a} \wedge \bar{b}, \quad C_1 = a \wedge \bar{b}, \quad C_2 = \bar{a} \wedge b, \quad C_3 = a \wedge b.$$

მაშინ ფუნქციის მნიშვნელობებისა და მოცემული C_i - ისაგან, რომლებსაც ამ შემთხვევაში ჩვენ კ ო ნ ს ტ ი ტ უ ე ნ ტ ე ბ ა დ მოვიხსენიებთ, დამოკიდებულებით მივიღებთ თექვსმეტ ლოგიკურ ოპერაციას:

$$y = \left[\bar{a} \wedge \bar{b} \wedge f(0,0)\right] \vee \left[a \wedge \bar{b} \wedge f(1,0)\right] \vee [\bar{a} \wedge b \wedge f(0,1)] \vee$$
$$\vee [a \wedge b \wedge f(1,1)].$$

ლოგიკური ფუნქციების წარმოდგენის მსგავს ფორმას ეწოდება ს რ უ ლ ყ ო ფ ი ლ ი დ ი ზ ი უ ნ ქ ც ი უ რ ი ნ ო რ მ ა ლ უ რ ი ფ ო რ მ ა (სდნფ).

ბულის ლოგიკაში მოქმედებს ო რ ა დ ო ბ ი ს პ რ ი ნ ც ი პ ი, რომლის მიხედვითაც: $\wedge \Leftrightarrow \vee$ და $1 \Leftrightarrow 0$ სიმბოლოთა ერთდროული ცვლილებისას ყველა ლოგიკური ტოლობა ძალაში რჩება. ამიტომ ჩვენი სდნფ შეიძლება წარმოვიდგინოთ სხვაგვარადაც:

$$y = \left[\bar{a} \vee \bar{b} \vee f(1,1)\right] \wedge \left[a \vee \bar{b} \vee f(0,1)\right] \wedge [\bar{a} \vee b \vee f(1,0)] \wedge$$
$$\wedge [a \vee b \vee f(0,0)].$$

წარმოდგენის ამ ფორმას ს რ უ ლ ყ ო ფ ი ლ ი კ ო ნ ი უ ნ ქ - ც ი უ რ ი ნ ო რ მ ა ლ უ რ ი ფ ო რ მ ა (სკნფ) ეწოდება. აქ უკვე კონსტიტუენტები წარმოდგენილებია არა კ ო ნ ს ტ ი ტ უ ე ნ ტ ე ბ- ი ს სახით, როგორც ეს სდნფ-ში, არამედ დ ი ზ ი უ ნ ქ ტ ე ბ ი ს სახით. ეს დიზიუნქტები კი დაკავშირებულებია კონიუნქციით, სწორედ აქედანაა დასახელებაც - სკნფ.

არსებობს ლოგიკურ ფუნქციათა წარმოდგენის აგრეთვე მესამე ფორმაც - ს რ უ ლ ყ ო ფ ი ლ ი პ ო ლ ი ნ ო მ ი ა ლ უ რ ი ნ ო - რ მ ა ლ უ რ ი ფ ო რ მ ა (სპნფ). მისი მიღება ადვილია სდნფ-დან შეცვლის გზით: $a \vee b = a \oplus b \oplus ab, \quad \bar{a} = 1 \oplus a$.

ვინაიდან კონსტიტუენტები არ გადაიკვეთებიან $(C_iC_j = 0)$, ამიტომ ჩვენ პირდაპირ შეგვიძლია ჩავწეროთ (სპნფ-ში კონიუნქციის სიმბოლო გამოტოვებულია):

$$y = [(1\oplus a)(1\oplus b)f(0,0)]\oplus[a(1\oplus b)f(1,0)]\oplus[(1\oplus a)bf(0,1)]\oplus$$
$$\oplus[abf(1,1)] = f(0,0)\oplus a[f(0,0)\oplus f(1,0)]\oplus b[f(0,0)\oplus f(0,1)]\oplus$$
$$\oplus ab[f(0,0)\oplus f(1,0)\oplus f(0,1)\oplus f(1,1)].$$

ცხრილ 2.13-ში მოყვანილია ე ლ ე მ ე ნ ტ ა რ უ ლ ლ ო გ ი-კ უ რ ფ უ ნ ქ ც ი ა თ ა სრული სია ორი არგუმენტიდან სამ ს რ უ-ლ ყ ო ფ ი ლ ფ ო რ მ ა შ ი - სდნფ, სკნფ და სპნფ - ში წარმოდგენათა სრულყოფილი ფორმები საშუალებას გვაძლევენ გამოვსახოთ ანალიზური ფორმულით ნებისმიერი ფუნქცია, თუ ცნობილია მისი ჭეშმარიტების ცხრილი.

ცხრილი 2.13

$y = f(a,b)$	სდნფ = სკნფ = სპნფ
$y_0 = 0$	$= (a \vee b) \wedge (\bar{a} \vee b) \wedge \left(a \vee \bar{b}\right) \wedge \left(\bar{a} \vee \bar{b}\right) = 0$
$y_1 = a \wedge b$	$= (a \vee b) \wedge (\bar{a} \vee b) \wedge \left(a \vee \bar{b}\right) = ab$
$y_2 = b - a$	$= \bar{a} \wedge b = (a \vee b) \wedge (\bar{a} \vee b) \wedge \left(\bar{a} \vee \bar{b}\right) = b\oplus ab$
$y_3 = b$	$= (\bar{a} \wedge b) \vee (a \wedge b) = (a \vee b) \wedge (\bar{a} \vee b) = b$
$y_4 = a - b$	$= a \wedge \bar{b} = (a \vee b) \wedge \left(a \vee \bar{b}\right) \wedge \left(\bar{a} \vee \bar{b}\right) = a\oplus ab$
$y_5 = a$	$= \left(a \wedge \bar{b}\right) \vee (a \wedge b) = (a \vee b) \wedge \left(a \vee \bar{b}\right) = a$
$y_6 = a\oplus b$	$= \left(a \wedge \bar{b}\right) \vee (\bar{a} \wedge b) = (a \vee b) \wedge \left(\bar{a} \vee \bar{b}\right) = a\oplus b$
$y_7 = a \vee b$	$= \left(a \wedge \bar{b}\right) \vee (\bar{a} \wedge b) \vee (a \wedge b) = a \vee b = a\oplus b\oplus ab$
$y_8 = a \downarrow b$	$= \bar{a} \wedge \bar{b} = (a \vee b) \wedge (\bar{a} \vee b) \wedge \left(\bar{a} \vee \bar{b}\right) = 1\oplus a\oplus b\oplus ab$

$y_9 = a \sim b$	$= (\bar{a} \wedge \bar{b}) \vee (a \wedge b) = (\bar{a} \vee b) \wedge (a \vee \bar{b}) = 1 \oplus a \oplus b$
$y_{10} = \bar{a}$	$= (\bar{a} \wedge \bar{b}) \vee (\bar{a} \wedge b) = (\bar{a} \vee b) \wedge (\bar{a} \vee \bar{b}) = 1 \oplus a$
$y_{11} = a \to b$	$= (\bar{a} \wedge \bar{b}) \vee (\bar{a} \wedge b) \vee (a \wedge b) = \bar{a} \vee b = 1 \oplus a \oplus ab$
$y_{12} = \bar{b}$	$= (\bar{a} \wedge \bar{b}) \vee (a \wedge \bar{b}) = (a \vee \bar{b}) \wedge (\bar{a} \vee \bar{b}) = 1 \oplus b$
$y_{13} = b \to a$	$= (\bar{a} \wedge \bar{b}) \vee (a \wedge \bar{b}) \vee (a \wedge b) = a \vee \bar{b} = 1 \oplus b \oplus ab$
$y_{14} = a \vert b$	$= (\bar{a} \wedge \bar{b}) \vee (a \wedge \bar{b}) \vee (\bar{a} \wedge b) = \bar{a} \vee \bar{b} = 1 \oplus ab$
$y_{15} = 1$	$= (\bar{a} \wedge \bar{b}) \vee (a \wedge \bar{b}) \vee (\bar{a} \wedge b) \vee (a \wedge b) = 1$

ვთქვათ მოცემულია ჭეშმარიტების კონკრეტული ცხრილი (ცხრ. 2.14) სამ არგუმენტზე დამოკიდებული ფუნქციისათვის. მაშინ y-ის ერთეულოვანი მნიშვნელობის გასწვრივ მდებარე კონიუნქტების ამოწერით მივიღებთ სდნფ-ს. თუ კი ამოვიწერთ y-ის ნულოვანი მნიშვნელობების გასწვრივ მოთავსებულ დიზიუნქტებს, მაშინ უკვე მივიღებთ სკნფ-ს. და ბოლოს, სპნფ მიიღება სდნფ-ში ∨-ის ⊕-ზე და x -ის $(1 \oplus x)$-ზე შეცვლით.

ცხრილი 2.14

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

$y_{სდნფ} = (\bar{x}_3 \wedge \bar{x}_2 \wedge x_1) \vee (\bar{x}_3 \wedge x_2 \wedge x_1) \vee (x_3 \wedge \bar{x}_2 \wedge \bar{x}_1) \vee (x_3 \wedge x_2 \wedge x_1)$,

$y_{სკნფ} = (x_3 \vee x_2 \vee x_1) \wedge (x_3 \vee \bar{x}_2 \vee x_1) \wedge (\bar{x}_3 \vee x_2 \vee \bar{x}_1) \wedge (\bar{x}_3 \vee \bar{x}_2 \vee x_1)$,

$y_{სპნფ} = (x_3 \oplus 1)(x_2 \oplus 1)x_1 \oplus x_3 x_2 x_1 \oplus (x_3 \oplus 1)x_2 x_1 \oplus x_3(x_2 \oplus 1)(x_1 \oplus 1)$.

ბოლო გამოსახულება $y_{სპნფ}$-სთვის ადვილად შეიძლება გამარტივდეს, თუ გავხსნით ფრჩხილებს და ურთიერთ შევკვეცავთ ყველა

ერთნაირ შესაკრებს, რომელიც ფორმულაში ლუწჯერ შედის (2-ის ჯერადჯერ შედის): $y_{სპნფ} = x_1 \oplus x_2 \oplus x_2 x_3$.

მსგავსი გამარტივება, რომელსაც ლ ო გ ი კ უ რ ი ფ უ ნ ქ -ც ი ი ს მ ი ნ ი მ ი ზ ა ც ი ა ჰქვია შეიძლება განხორციელდეს სდნფ - სა და სკნფ - სათვისაც.

ბულის ლოგიკაში მოქმედებს შ ე წ ე ბ ე ბ ი ს კ ა ნ ო ნ ი:

$$\left(a \wedge \bar{b}\right) \vee (a \wedge b) = a, \quad \left(a \vee \bar{b}\right) \wedge (a \vee b) = a.$$

ამ კანონების გამოყენება საშუალებას იძლევა ვიპოვოთ მოცე-მული y ფუნქციისათვის უფრო კომპაქტური ანალიზური გამოსახუ-ლებები, ანუ მინიმალური დ ი ზ ი უ ნ ქ ც ი უ რ ი ნ ო რ მ ა ლ უ - რ ი ფ ო რ მ ა $y_{მდნფ}$. მოვიყვანოთ ცხრილი 2.14-ით მოცემული y ფუნქციის წარმოდგენის შესაბამისი ფორმები:

$$y_{მდნფ} = (\bar{x}_3 \wedge x_1) \vee (x_2 \wedge x_1) \vee (x_3 \wedge \bar{x}_2 \wedge \bar{x}_1),$$

და სკნფ-სათვის, ანუ მ ი ნ ი მ ა ლ უ რ ი კნფ:

$$y_{მკნფ} = (x_3 \vee x_1) \wedge (\bar{x}_3 \vee x_2 \vee \bar{x}_1) \wedge (\bar{x}_2 \vee x_1).$$

მ ი ნ ი მ ა ლ უ რ ი ნ ო რ მ ა ლ უ რ ი ფ ო რ მ ე ბ ი ს (მნფ) პოვნის შემდეგ რეკომენდირებულია მათი შემოწმება x_i არგუმენტე-ბის ყველა ნაკრებზე. x_i ან $\bar{x}_i$ ცვლადებს ხშირად უწოდებენ თ ე რ-მ ე ბ ს. სწორედ n თერმის სრული ნაკრები ქმნის კ ო ნ ს ტ ი ტ უ-ე ნ ტ ა ს. მინიმიზაციის პროცესში კი ზოგიერთი თერმი ქრება კონ-სტიტუენტიდან. მაშინ დიზიუნქტის ან კონიუნქტის დარჩენილ ნა-წილს ი მ პ ლ ი კ ა ნ ტ ა ს უწოდებენ.

როგორც მაგალითზე დავრწმუნდით, იმპლიკანტები ჩნდები-ან ერთი თერმით განსხვავებული მ ო ს ა ზ ღ ვ რ ე კონსტიტუენ-ტების შეწებების შედეგად. მაგრამ მრავალ ცვლადზე დამოკიდებუ-ლი ფუნქციებისათვის (მრავალი ცვლადის ფუნქციებისათვის), შე-წებების არაკონტროლირებად პროცესს აუცილებლად მივყავართ ზედმეტ იმპლიკანტებთან. იმპლიკანტების საჭირო რიცხვი შეიძლე-ბა აღმოჩნდეს ბევრად უფრო ნაკლები მოსაზღვრე შეწებებათა შესაძ-ლო რიცხვზე. ასეთ შემთხვევაში ჭეშმარიტ მნფ-ს ღებულობენ მინი-

მიზაციის სპეციალური მეთოდების დახმარებით, რომელთაგან სამს ჩვენ ახლა განვიხილავთ. ამასთან საჭიროა გვახსოვდეს, რომ ქვემოთ განხილული მინიმიზაციის მეთოდები ეხებიან მხოლოდ სდნფ-ს. ეს მეთოდები ორადობის პრინციპის საფუძველზე ადვილად შეიძლება გავრცელდნენ სკნფ-ზეც.

როგორც ზემოთ ავღნიშნეთ, სდნფ და სკნფ -ში ფუნქციათა წარმოდგენა განხორციელებულია სამი ოპერაციით - დიზუნქციით, კონიუნქციით და უარყოფით, ხოლო სპნფ-ში - ორის მოდულით შეკრებით, კონიუნქციით და ერთიანით როგორც ოპერაციით. ბულის ლოგიკაში მოქმედებს ს უ პ ე რ პ ო ზ ი ც ი ი ს პ რ ი ნ ც ი პ ი, რომლიც მიხედვითაც ნებისმიერი რთული ფუნქცია შეიძლება წარმოდგენილი იყოს ორი არგუმენტის ელემენტარული ფუნქციების ერთობლიობით, მაგალითად:

$$F(x_1,x_2,x_3,x_4) = \big((x_1|x_2) \to (x_2 \vee x_3)\big) \downarrow (\bar{x}_3 \oplus x_4) =$$
$$= f_8\{f_{11}[f_{14}(x_1,x_2), f_7(x_2,x_3)], f_6[f_{10}(x_3,x_4), x_4]\}.$$

ცხრილი 2.15 წარმოადგენს ჭეშმარიტების ცხრილს რთული $F(x_1,x_2,x_3,x_4)$ ფუნქციისათვის. არგუმენტთა ყველა ნაკრებზე, გარდა ორისა, ეს ფუნქცია ნულის ტოლია, ამიტომ ის შეიძლება წარმოვადგინოთ ერთი კონსტიტუციის სახით, რომელიც გამოისახება გამოკლების ოპერაციით:

$$F(x_1,x_2,x_3,x_4) = \bar{x}_2 \wedge \bar{x}_3 \wedge x_4 = (x_4 - x_3) - x_2.$$

ცხრილი 2.15

x_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$f_{11} = x_1\|x_2$	1	1	1	0	1	1	1	0	1	1	1	0	1	1	1	0

$f_7 = \boldsymbol{x_2} \vee \boldsymbol{x_3}$	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1
$f_{10} = \bar{\boldsymbol{x}}_3$	1	1	1	1	0	0	0	0	1	1	1	1	0	0	0	0
$f_6 = f_{10} \oplus \boldsymbol{x_4}$	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1
$f_{11} = f_{14} \rightarrow f_7$	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1
$f_8 = f_{11} \downarrow f_6$	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0

დავსვათ შეკითხვა: ლოგიკურ ოპერაციათა კიდევ რომელი სისტემითაა შესაძლებელი ნებისმიერად აღებული ბულის ფუნქციის გამოსახვა? ამ შეკითხვასთან დაკავშირებით განვსაზღვროთ ფ უ ნ ქ ც ი ა თ ა ხუთი კ ლ ა ს ი.

ფ უ ნ ქ ც ი ა, რომელიც ი ნ ა რ ჩ უ ნ ე ბ ს ნ უ ლ ო ვ ა ნ მ ნ ი შ ვ ნ ე ლ ო ბ ა ს თერმების ნულოვან ნაკრებზე: $f(0{,}0) = 0$ განსაზღვრავს 0-კლასს. ამ კლასს მიეკუთვნება ყველა ელემენტალუ-რი ფუნქცია 0-დან 7-მდე (იხ. ცხრილი 2.13).

ანალოგიურად ვსაზღვრავთ I-კლასს, რომელიც ი ნ ა ხ ა ვ ს კ ო ნ ს ტ ი ტ უ ე ნ ტ ა 1-ს ერთეულოვან ნაკრებზე: $f\ (1{,}1) = 1$. 1-კლასს მიეკუთვნებიან კენტი ფუნქციები.

წ რ ფ ი ვ ი ფ უ ნ ქ ც ი ე ბ ი ს კლასი (II- კლასი) განისაზღვ-რება პოლინომიალური ფორმის წრფივობით. ასე, მაგალითად, ექ-ვივალენტობა არის წრფივი ფუნქცია, ვინაიდან $f_9 = 1 \oplus a \oplus b$, ხოლო პირსის ისარი ab შესაკრების არაწრფივობის გამო უკვე აღარ იქნება ასეთი: $f_8 = 1 \oplus a \oplus b \oplus ab$.

თ ვ ი თ ო რ ა დ უ ლ ი ფ უ ნ ქ ც ი ე ბ ი ს კლასი (III-კლასი) აიწერება ფორმულით: $f(a, b) = \bar{f}\left(a, \bar{b}\right)$. ასეთი ელემენტარული ფუნ-ქცია მხოლოდ ოთხია.

და ბოლოს, მ ო ნ ო ტ ო რ უ ლ ფ უ ნ ქ ც ი ა თ ა კლასი (IV - კლასი) განისაზღვრება უტოლობით:

$f(a,b) \le f'(a',b')$, როცა $a \le a'$ და $b \le b'$.

მაგალითად, ვთქვათ $a = 0,\ a' = 1,\ b = 1$ და $b' = 1$, მაშინ დიზიუნქციისათვის გვექნება:

$$(f_7 = a \vee b = 1) \le (f_7' = a' \vee b' = 1)$$

და a, a', b, b' - ის რა ნაკრებებიც არ უნდა ავიღოთ, თუ სრულდება პირობა $a \le a'$ და $b \le b'$, ყოველთვის ექნება ადგილი $f_7 \le f_7{}'$; რაც ნიშნავს, რომ დიზიუნქცია მონოტორული ფუნქციაა.

ელემენტარული ფუნქციების ამა თუ იმ კლასისათვის (K) მიკუთვნება აღნიშნულია ერთიანით ცხრილ 2.16-ში. ამ ცხრილით უკვე ადვილია ბ ა ზ ი ს უ რ ი ფ უ ნ ქ ც ი ე ბ ი ს სისტემის განსაზღვრა.

ცხრილი 2.16

K	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
2	1	0	0	1	0	1	1	0	0	1	1	0	1	0	0	1
3	0	0	0	1	0	1	0	0	0	0	1	0	1	0	0	0
4	1	1	0	1	0	1	0	1	0	0	0	0	0	0	0	1

ფუნქციათა სისტემა ბაზისურია, თუ ის ნულებით ფარავს 0-, 1-, 2-, 3-, 4-კლასის ყველა სტრიქონს. მაგალითად, სპნფ შექმნილია f_1, f_6, f_{15} ფუნქციებით. ამ სამი ფუნქციისათვის ნულები იმყოფებიან ცხრილ 2.16-ის ხუთივე სტრიქონში. გამომდინარე, სპნფ-ში შეიძლება წარმოდგენილი იყოს ნებისმიერი რაგინდ რთული ფუნქცია.

სდნფ და სკნფ შექმნილია f_1, f_7, f_{10} ფუნქციებით. ხუთივე კლასის ნულებით დაფარვა უკვე ორი ფუნქციით მიიღწევა: ან f_1

და f_{10}-ით, ან f_7 და f_{10}-ით ანუ ამ ფორმებში არის ფუნქციათა რაღაც მეტობა.

ფუნქციათა ხუთი კლასის თეორიის შედეგს წარმოადგენს პ ო ს ტ ი ს თ ე ო რ ე მ ა, რომელიც შემდეგში მდგომარეობს: იმისათვის, რომ ფუნქციათა სისტემა ბაზისური იყოს, აუცილებელია და საკმარისი რომ მასში შედიოდეს თუნდაც ერთი ფუნქცია, რომელიც არ მიეკუთვნება 0-, 1-, 2-, 3- და 4-კლასებს (ცხრილ 2.16-ის ყველა სტრიქონში ნულების არსებობის პირობა). მაგრამ ბაზისურ ფუნქციათა სისტემის დასაყენებლად სრულებითაც არაა სავალდებულო ფუნქციათა ხუთი კლასის შემოტანა: საკმარისია ოპერაციებს შორის ურთიერთკავშირების ცოდნა. ვინაიდან დადგენილი იყო, რომ ნებისმიერი ლოგიკური ოპერაცია შეიძლება წარმოდგენილ იქნას სამი ბულის ფუნქციით, ამიტომ საჭიროა ამ სამის ყველა დანარჩენით გამოსახვა. ჭეშმარიტების ცხრილების საფუძველზე შეიძლება დავრწმუნდეთ შემდეგი ტოლობების სამართლიანობაში:

$$a \to b = \overline{a - b},$$
$$\overline{a} = 1 - a = a \to 0 = 1 \oplus a = a \sim 0 = a|a = a \downarrow a,$$
$$a \wedge b = a - \overline{b} = (a|b)|(a|b) = (a \downarrow a) \downarrow (b \downarrow b),$$
$$a \vee b = \bar{a} \to b = (a|a)|(b|b) = (a \downarrow b) \downarrow (a \downarrow b),$$
$$a \oplus b = \bar{a} \oplus \overline{b} = \bar{a} \sim b = a \sim \overline{b} = \overline{a \sim b},$$
$$1 = a \to a = \bar{a} \vee a = a \sim a,$$
$$0 = a - a = \bar{a} \wedge a = a \oplus a.$$

2.4. ლოგიკის ალგებრის ფუნქციების მინიმიზაცია.

ლოგიკის ალგებრის ფუნქციათა (ლაფ) მ ი ნ ი მ ი ზ ა ც ი ა - ესაა ლაფ-ის ყველაზე მარტივი წარმოდგენის მიღების პროცედურა იმ ფუნქციათა სუპერპოზიციის სახით, რომლებიც ქმნიან ფუნქციონალურად სრულ სისტემას, ერთდროულად მისი ტექნიკური რეალიზაციის ოპტიმიზაციისას გარკვეული კრიტერიუმებით რიგი შეზღუდვების პირობებში. ოპტიმიზაციის კრიტერიუმები შეიძლება იყოს მოწყობილობათა მოცულობა, გაბარიტები, წონა, ენერგომოხმარება, ღირებულება, სწრაფქმედება, საიმედობა. შეზღუდვების სახით შეიძლება გამოდიოდნენ სისტემაში გამოსაყენებლად დაშვებული ელემენტები, კორპუსში ელემენტების რაოდენობა, შესასვლელთა მიერ გაერთიანების კოეფიციენტები და ლოგიკურ ელემენტთა გამოსასვლელების განშტოებები, ლაფ სისტემის რეალიზაციის აუცილებლობა, აგრეთვე ზემოთ ჩამოთვლილი ოპტიმიზაციის კრიტერიუმების ნაწილი.

ლაფ-ის მინიმიზაციის ამოცანის სრული მოცულობით გადაჭრა რთული პრობლემაა, თუნდაც იმიტომ, რომ ოპტიმიზაციის კრიტერიუმების ნაწილი წინააღმდეგობაშია ერთმანეთთან, მაგალითად, ენერგომოხმარების შემცირებისა და სწრაფქმედების ერთდროული ამაღლება. პრაქტიკაში ჩვეულებრივ წყდება ოპტიმიზაციის ამოცანა რამდენიმე ან თუნდაც ერთი კრიტერიუმის მიხედვით. დღეისათვის ყველაზე ფართო გავრცელება მოიპოვა ინვერსიის, კონიუნქციისა და დიზიუნქციისაგან აგებულმა ბაზისმა, ვინაიდან მასში შემავალი ფუნქციები მათემატიკური გარდაქმნებისა და ტექნიკური რეალიზაციის თვალსაზრისით ყველაზე მარტივებია და ამ დროს უმრავლეს შემთხვევაში კმაყოფილდება მინიმალური გაბარიტების, წონის, ენერგომოხმარების, ღირებულების, აგრეთვე სწრაფქმედებისა და საიმედობის ამაღლების მოთხოვნები. გარდა ამისა, მისგან ადვილია სხვა ნებისმიერ ბაზისში გადასვლა. ზოგჯერ შემო-

ისაზღვრებიან ლაფ-ის წარმოდგენის კიდევ უფრო მარტივი ამოცა-ნით დიზიუნქციურ ან კონიუნქციურ ფორმაში, რომელიც შეიცავს ასოთა შესაძლოდ მცირე რაოდენობას, როცა, მაგალითად, დიზიუნქ-ციური ფორმისათვის, გამოსახულებაში არსებობს იმ შესაკრებთა რაც შეიძლება ნაკლები რაოდენობა, რომლებიც ელემენტარულ ნამ-რავლებს წარმოადგენენ, თავად ისინი კი რაც შეიძლება ნაკლებ თა-ნამამრავლებს შეიცავენ. ასეთ ამოცანას ლ ა ფ - ის მ ი ნ ი მ ი ზ ა ც ი-ი ს კ ა ნ ო ნ ი კ უ რ ა მ ო ც ა ნ ა ს უ წ ო დ ე ბ ე ნ.

უმრავლესი მეთოდებისათვის საწყის ფორმას წარმოადგენს ან ჭეშმარიტების ცხრილი, ან ერთ-ერთი სრულყოფილი ფორმა: სდნფ ან სკნფ. თუ ლაფ მოცემულია სხვა სახით, მაშინ იგულისხმება, რომ ის ჯერ გადადის სდნფ-ში ან სკნფ-ში ბულის ალგებრის ძირითადი კანონების გამოყენებით.

მოცემული ფუნქციის ნორმალურ ფორმას (დიზიუნქციურს და კონიუნქციურს) ეწოდება მ ი ნ ი მ ა ლ უ რ ი, თუ იმ ასოთა რაოდენობა, რომელსაც ის შეიცავს, არ იქნება იმაზე მეტი, ვიდრე ნებისმიერ მის ნორმალურ ფორმაში.

ყურადღება მიაქციეთ იმას, რომ საუბარია ასოების მინიმა-ლურ რაოდენობაზე და არა ცვლადების. მაგალითად, $f(x,y,z) = xy \vee x\bar{y}z \vee yz$ გამოსახულება შეიცავს 7 ასოს და მხოლოდ 3 ცვლადს.

ზოგიერთ ფუნქციას რამდენიმე მინიმალური ფორმა გააჩნია. მათი პოვნა სპეციალური მეთოდებით შეიძლება, რომლებსაც ჩვენ ქვემოთ განვიხილავთ.

შემოვიტანოთ *რამდენიმე აუცილებელი ცნება.*

განვიხილოთ ფუნქცია $f(x,y) = \bar{x}\bar{y} \vee \bar{x}y \vee xy$. თითოეული შე-საკრები შეესაბამება მხოლოდ ერთ ერთიანს მოცემული ფუნქციის ჭეშმარიტების ცხრილში. ამბობენ, რომ თითოეული შესაკრები ფა-რავს ფუნქციის ერთიანს, ხოლო მთლიანობაში ისინი ფარავენ მოცე-მულ ფუნქციას, ანუ წარმოადგენენ მის გ ა დ ა ფ ა რ ვ ე ბ ს. მაგრამ შევნიშნოთ, რომ ფუნქციის გამარტივების შემდეგ $f(x,y) = x \vee \bar{y}$,

ვღებულობთ უფრო მარტივ გადაფარვას. ორივე წარმოდგენა შეესაბამება ფუნქციის ერთი და იგივე ჭეშმარიტების ცხრილს, ანუ გადადიან 1 და 0-ში x, y ცვლადების ერთი და იგივე ნაკრებებზე. თუ დავაკვირდებით მეორე წარმოდგენის ცალკეულ შესაკრებებს, ადვილი შესამჩნევია, რომ x ერთიანი ხდება ორ ნაკრებზე $(1,0),(1,1)$, ხოლო y - $(0,0),(1,0)$ ნაკრებზე; ერთად ისინი ერთიანებით ფარავენ მოცემული ფუნქციის ყველა ერთიანს. შევნიშნოთ, რომ ორივე შესაკრები x და $\bar{y}$ ერთდროულად ნული ხდებიან $(0,1)$ ნაკრებზე, ანუ იქ, სადაც $f(x,y)$ ფუნქცია ნულის ტოლია.

თუ φ ფუნქცია ნულის ტოლია ცვლადების იგივე ნაკრებებზე, რომლებზეც ნულის ტოლია მოცემული f ფუნქცია, მაშინ ამბობენ, რომ φ ფუნქცია შ ე დ ი ს ამ ფუნქციაში.

სხვა სიტყვებით, φ შედის f -ში, თუ ის ნულებით ფარავს f ფუნქციის ყველა ნულს, ანუ გააჩნია ნულების არა ნაკლები რაოდენობა.

φ ფუნქციას, რომელიც ელემენტალური ნამრავლია და შედის f ფუნქციაში, ი მ პ ლ ი კ ა ნ ტ ა ეწოდება.

მოცემული f ფუნქციის იმპლიკანტებს შორის გამოყოფენ ე. წ. მ ა რ ტ ი ვ ი მ პ ლ ი კ ა ნ ტ ე ბ ს, ანუ ისეთებს, რომლებიც თავად შედიან f - ში, ხოლო არავითარი მათი ნაწილი (ელემენტალური ნამრავლი, მიღებული მოცემული იმპლიკანტიდან ერთი ან რამდენიმე თანამამრავლის გამორიცხვით) f ფუნქციაში არ შედის.

მაგალითად: $$\left.\begin{array}{r} xy\bar{z} \subset f \\ y\bar{z} \subset f \\ y \not\subset f \\ \bar{z} \not\subset f \end{array}\right\} \rightarrow y\bar{z}$$ - არის მარტივი იმპლიკანტა.

მარტივი იმპლიკანტები მოცემულ ფუნქციაში შემავალი ყველაზე მოკლე ნამრავლებია. თუ რომელიმე ელემენტალური ნამრავ-

ლი შედის მოცემულ ფუნქციაში, მაშინ მასზე ნებისმიერი თანამამრავლის დამატების შემდეგ ახალი ნამრავლიც აგრეთვე შევა ამ ფუნქციაში, ვინაიდან ის ნული ხდება საწყის ნამრავლთან ერთად.

ბულის ნებისმიერი ფუნქცია ყველა თავისი მარტივი იმპლიკანტის დიზუნქციის ტოლია. ფუნქციის ასეთ წარმოდგენას *შემოკლებული* დიზიუნქციური ნორმალური ფორმა ეწოდება. შემოკლებული ფორმა იმით ხასიათდება, რომ მისი წევრები ყველაზე მოკლეებია, მისგან უკვე აღარ შეიძლება არც ერთი ასოს ამოგდება, მაგრამ შესაძლებელია ზოგიერთი იმპლიკანტის გადაგდება.

თუ შემოკლებული ფორმიდან გამოვრიცხავთ ყველა შესაძლო წევრს, მაშინ მივიღებთ *ჩიხურ* დიზიუნქციურ ნორმალურ ფორმას. ბულის ფუნქციას შეიძლება რამდენიმე ჩიხური ფორმა გააჩნდეს.

ჩიხურ ფორმას, რომელიც შეიცავს წევრთა უმცირეს რიცხვს, *უმოკლესი* დიზიუნქციური ნორმალური ფორმა ეწოდება. ზოგად შემთხვევაში უმოკლესი და ჩიხური ფორმები ერთმანეთს არ ემთხვევიან.

ნახ. 2.18-ზე მოყვანილია ბულის ფუნქციათა ფორმების გამარტივების სქემა.

ჭეშმარიტების ცხრილი

სდნფ

შემოკლებული დნფ

ჩიხური დნფ

მინიმალური დნფ | უმოკლესი დნფ

ნახ. 2.18.

ზოგად შემთხვევაში სდნფ-ით მოცემული ლაფ-ის მინიმიზაცია მოითხოვს შემდეგი სამი ეტაპის პროცედურათა შესრულებას:

I ეტაპი - სდნფ-დან *შემოკლებულ დნფ*-ზე (შდნფ) გადასვლა. შდნფ - ეს ლაფ-ის ფორმაა, რომლის წევრებიც წარმოად-

გენენ ი ზ ო ლ ი რ ე ბ უ ლ (შეუწებავ) ელემენტალურ ნამრავლებს. ეს ეტაპი დაფუძნებულია ჯერ ერთიანების ყველა შესაძლო კონსტიტუენტის ერთმანეთთან, ხოლო შემდეგ ნაკლები რანგის ნამრავლების (იმპლიკანტების) შეწებებაზე. ავღნიშნოთ, რომ არსებობენ ლაფები, რომლებისთვის სდნფ ემთხვევა შდნფ-ს.

II ეტაპი - შდნფ-დან ჩ ი ხ უ რ დ ნ ფ - ზე (ჩდნფ) გადასვლა. ჩდნფ - ეს ლაფ-ის ფორმაა, რომლის წევრებიც იმპლიკანტებია, რომელთა შორის არცერთი არაა ზედმეტი. ზ ე დ მ ე ტ ი მ პ ლ ი კ ა ნ ტ ა ს ლაფ-ის ისეთ წევრს ეძახიან, რომლის გამოსახულებიდან ამოღება არ ცვლის ლაფ-ს. შევნიშნოთ, რომ შესაძლოა ისეთი შემთხვევები, როცა შდნფ-ში არაა ზედმეტი იმპლიკანტები. ზოგჯერ ერთი სდნფ-იდან შეიძლება მივიღოთ რამდენიმე სხვადასხვა ჩდნფ. ტერმინი „ჩიხური“ იმაზე მეტყველებს, რომ შემდგომი მინიმიზაცია ნორმალური ფორმების ჩარჩოებში უკვე შეუძლებელია.

III ეტაპი - ჩდნფ-დან მ ი ნ ი მ ა ლ უ რ ი ფორმისკენ გადასვლა. ეს ეტაპი ეფუძნება ჯგუფური ინვერსიებისა და ფრჩხილების გამოყენებით, არაა სისტემატიური და ზევრადაა დამოკიდებული დამმუშავებლის გამოცდილებაზე.

დღეისათვის ლოგიკური სქემების დაპროექტების თეორიაში ყველაზე სრულადაა გამოკვლეული სწორედ დიზიუნქციური და კონიუნქციური ნორმალური ფორმების მინიმიზაციის ამოცანები, რომლებიც უზრუნველყოფენ კომბინაციური სქემების სინთეზის დროს რაციონალური გადაწყვეტილებების მიღებას.

არსებობს მინიმიზაციის ორი მიმართულება:

1. ჩანაწერის შემოკლებული ფორმა (მიზანი - თითოეული თერმის (წევრის) რანგის მინიმიზაცია). ამ დროს ვღებულობთ შემოკლებულ ფორმებს - შდნფ, შკნფ, შპნფ.

2. ჩანაწერის მინიმალური ფორმის მიღება (მიზანი - სიმბოლოების მინიმალური რიცხვის მიღება მთელი ფუნქციის ჩასაწერად).

საჭიროა გაითვალისწინოთ, რომ მ ი ნ ი მ ი ზ ა ც ი ი ს ა რ ც ე რ თ ი ხ ე რ ხ ი ა რ ა ა უ ნ ი ვ ე რ ს ა ლ უ რ ი!

ქვემოთ განვიხილავთ ლაფ-ის მინიმიზაციის რამდენიმე მეთოდს.

2.4.1. ლოგიკის ფუნქციათა უშუალო გარდაქმნის მეთოდი.

მინიმიზაციის ერთ-ერთ მარტივ მეთოდს უშუალო გარდაქმნათა მეთოდი წარმოადგენს, რომელიც სრულდება ბულის ალგებრის ძირითადი კანონებით.

ვთქვათ, გვსურს (2.1) გამოსახულებით წარმოდგენილი ლაფ-ის მინიმიზაცია. ამისათვის საჭიროა გავიაროთ შემდეგი ეტაპები:

$$y = \bar{x}_0 x_1 x_2 \vee x_0 \bar{x}_1 x_2 \vee \bar{x}_0 \bar{x}_1 x_2 \vee x_0 x_1 \bar{x}_2 \vee x_0 \bar{x}_1 \bar{x}_2 \tag{2.1}$$

I ეტაპი. ვასრულებთ ერთიანის კონსტიტუენტების შეწებების ოპერაციებს. ამ პროცედურის მოსაწესრიგებლად ჩავწეროთ (2.1) გამოსახულება რამდენიმე სტრიქონის სახით შემდეგი წესით: პირველი სტრიქონი - ეს საწყისი განტოლებაა, მეორე სტრიქონი - ეს მეორე კონსტიტუენტაა და ყველა მომდევნო, მესამე სტრიქონი - ეს მესამე კონსტიტუენტაა და ყველა მომდევნო და ა.შ. ეს დასაშვებია, ვინაიდან ბულის ალგებრაში მოქმედებს ტავტოლოგიის კანონი.

$$\begin{aligned} y = \bar{x}_0 x_1 x_2 \vee x_0 \bar{x}_1 x_2 \vee \bar{x}_0 \bar{x}_1 x_2 \vee x_0 x_1 \bar{x}_2 \vee x_0 \bar{x}_1 \bar{x}_2 \vee \\ \vee x_0 \bar{x}_1 x_2 \vee \bar{x}_0 \bar{x}_1 x_2 \vee x_0 x_1 \bar{x}_2 \vee x_0 \bar{x}_1 \bar{x}_2 \vee \\ \vee \bar{x}_0 \bar{x}_1 x_2 \vee x_0 x_1 \bar{x}_2 \vee x_0 \bar{x}_1 \bar{x}_2 \vee \\ \vee x_0 x_1 \bar{x}_2 \vee x_0 \bar{x}_1 \bar{x}_2 \end{aligned} \tag{2.2}$$

თითოეულ სტრიქონში ხორციელდება პირველი წევრის შეწებების შემოწმება იგივე სტრიქონის დანარჩენ წევრებთან. პირველ

სტრიქონში ხდება პირველი და მესამე კონსტიტუენტების შეწებება, მეორე სტრიქონში - პირველისა მეორესთან და მეოთხესთან, მესამე სტრიქონში პირველი კონსტიტუენტა დანარჩენებთან არ წებდება, ბოლო სტრიქონშიც კონსტიტუენტები არ ჭებდებიან. ვინაიდან ყველა კონსტიტუენტა მონაწილეობდა თუნდაც ერთი შეწებების ოპერაციაში, ამიტომ შდნფ-ში არცერთი კონსტიტუენტა არ იქნება. ამ პროცედურის შემდეგ ვღებულობთ შემდეგ გამოსახულებას:

$$y = \bar{x}_0 x_2 \vee \bar{x}_1 x_2 \vee x_0 \bar{x}_1 \vee x_0 \bar{x}_2 \quad (2.3)$$

შემდგომი შეწებება არ შეიძლება შესრულდეს, ვინაიდან (2.3) გამოსახულების ყველა წევრი იზოლირებულია.

II ეტაპი. აუცილებელია გამოსახულება (3)-ში ზედმეტი იმპლიკანტების გამოვლენა. ეს შეიძლება ორი ხერხით შესრულდეს. პირველი ხერხის დროს ერთ იმპლიკანტას ხსნიან ერთიანის კონსტიტუენტამდე, ხოლო შემდეგ აკვირდებიან, შთაინთქმებიან თუ არა ეს კონსტიტუენტები დანარჩენი იმპლიკანტებით. პირველი იმპლიკანტა იხსნება ჯამამდე

$$\bar{x}_0 x_2 = \bar{x}_0 \cdot 1 \cdot x_2 = \bar{x}_0 (x_1 \vee \bar{x}_1) x_2 = \bar{x}_0 x_1 x_2 \vee \bar{x}_0 \bar{x}_1 x_2,$$

ამასთან კონსტიტუენტა $\bar{x}_0 x_1 x_2$ არ შთაინთქმება არცერთი იმპლიკანტით, გამომდინარე, $\bar{x}_0 x_2$ არაა ზედმეტი. მეორე იმპლიკანტა $\bar{x}_1 x_2$ იხსნება $x_0 \bar{x}_1 x_2 \vee \bar{x}_0 \bar{x}_1 x_2$ ჯამამდე, ამასთან ორივე კონსტიტუენტა შთაინთქმება დანარჩენი იმპლიკანტებით, გამომდინარე, $\bar{x}_1 x_2$ იმპლიკანტა ზედმეტია. დავტოვოთ ჯერ $\bar{x}_1 x_2$ იმპლიკანტა გამოსახულება (2.3)-ში და გავაგრძელოთ ეს პროცედურა. იმპლიკანტა $x_0 \bar{x}_1$ იხსნება ჯამამდე $x_0 \bar{x}_1 x_2 \vee x_0 \bar{x}_1 \bar{x}_2$, ამასთან ორივე კონსტიტუენტა შთაინთქმება დანარჩენი იმპლიკანტებით, გამომდინარე, $x_0 \bar{x}_1$ იმპლიკანტა ზედმეტია. დავტოვოთ ეს იმპლიკანტაც გამოსახულება (2.3)-ში და გავაგრძელოთ პროცედურა. ბოლო იმპლიკანტის გახსნა გვაძლევს ჯამს $x_0 x_1 \bar{x}_2 \vee x_0 \bar{x}_1 \bar{x}_2$, რომელშიც კონსტიტუენტა $x_0 \bar{x}_1 \bar{x}_2$ არ შთაინთქმება არცერთი იმპლიკანტით, გამომდინარე, იმპლიკანტა $x_0 \bar{x}_2$ ზედმეტი არაა. გამოვლენილია ორი ზედმეტი იმპლიკანტა,

მაგრამ ეს იმას არ ნიშნავს, რომ ორივე მათგანი შეიძლება გადაგდე-ბულ იქნას, ვინაიდან თითოეული მათგანი მოწმდებოდა მეორის გა-მოსახულება (2.3)-ში არსებობისას. გამომდინარე, შეიძლება ერთი მათგანის გადაგდება, ხოლო შემდეგ ისევ ჩასატარებელია შემოწმება მეორის გადაგდების შესაძლებლობის შესახებ. თუ გადავაგდებთ $\bar{x}_1 x_2$ იმპლიკანტას, მაშინ შემოწმება გვიჩვენებს, რომ $x_0\bar{x}_1$ იმპლიკან-ტა ზედმეტი არაა, ხოლო თუ გადავაგდებთ $x_0\bar{x}_1$ იმპლიკანტას, მაშინ $\bar{x}_1 x_2$ იმპლიკანტა აღარ იქნება ზედმეტი. ამრიგად, შეიძლება გა-დავაგდოთ გამოვლენილი ორი იმპლიკანტიდან ერთ-ერთი და შე-დეგად მივიღებთ ერთნაირი სირთულის ორ ჩდნფ-ს, რომელთაგან თითოეული შედგება ექვს-ექვსი ასოსაგან:

$$y = \bar{x}_0 x_2 \vee x_0\bar{x}_1 \vee x_0\bar{x}_2 \tag{2.4}$$

$$y = \bar{x}_0 x_2 \vee \bar{x}_1 x_2 \vee x_0\bar{x}_2 \tag{2.5}$$

III ეტაპი. გამოსახულება (2.4) შეიძლება ასე ჩავწეროთ:

$$y = \bar{x}_0 x_2 \vee x_0(\bar{x}_1 \vee \bar{x}_2) \tag{2.6}$$

ის შეიცავს ხუთ ასოს და მოითხოვს სამ ინვერტორს. ორმაგი უარყო-ფის კანონისა და დე-მორგანის წესის გამოყენებით გამოსახულება (6) შეიძლება ასე გარდავქმნათ:

$$y = \bar{x}_0 x_2 \vee x_0\overline{\overline{(\bar{x}_1 \vee \bar{x}_2)}} = \bar{x}_0 x_2 \vee x_0 \cdot \overline{x_1 x_2} \tag{2.7}$$

ბოლო გამოსახულება შეიცავს ხუთ ასოს და მოითხოვს ორ ინ-ვერტორს.

ანალოგიურად შეიძლება გავამარტივოთ გამოსახულება (2.5):

$$y = x_2(\bar{x}_0 \vee \bar{x}_1) \vee x_0\bar{x}_2 = x_2 \cdot \overline{x_0 x_1} \vee x_0\bar{x}_2 \tag{2.8}$$

ზედმეტი იმპლიკანტების გამოვლენის მეორე ხერხი შემდეგში მდგომარეობს. ფუნქციის ჭეშმარიტების მნიშვნელობაზე გავლენას ახდენს მხოლოდ ის იმპლიკანტა, რომელიც თავადაა ერთის ტოლი. ნებისმიერი იმპლიკანტა იღებს მნიშვნელობა 1 თავისი არგუმენ-ტების მხოლოდ ერთ ნაკრებზე. მაგრამ თუ ზუსტად ამ ნაკრებზე დარჩენილი იმპლიკანტების ჯამი აგრეთვე ხდება 1, მაშინ განხილუ-

ლი იმპლიკანტა არ მოქმედებს ფუნქციის ჭეშმარიტების მნიშვნელობაზე ამ ერთადერთ შემთხვევაშიც კი, ანუ ის ზედმეტია.

გამოვიყენოთ ეს წესი (2.3) გამოსახულებისთვის. $\bar{x}_0x_2$ იმპლიკანტა ღებულობს მნიშვნელობა 1 ნაკრებზე $x_0 = 0,\ x_2 = 1$. ჩავსვათ ეს ნაკრები დარჩენილ ჯამში $\bar{x}_1x_2 \vee x_0\bar{x}_1 \vee x_0\bar{x}_2$, მივიღებთ $\bar{x}_1$, რაც იმაზე მეტყველებს, რომ პირველი იმპლიკანტა ზედმეტი არაა. $\bar{x}_1x_2$ ღებულობს მნიშვნელობა 1 ნაკრებზე $x_1 = 0,\ x_2 = 1$. ჩავსვათ ეს ნაკრები ჯამში $\bar{x}_0x_2 \vee x_0\bar{x}_1 \vee x_0\bar{x}_2$, მივიღებთ $x_0 \vee \bar{x}_0 = 1$, რაც იმაზე მეტყველებს, რომ $\bar{x}_1x_2$ იმპლიკანტა ზედმეტია.

დავტოვოთ ჯერ ეს იმპლიკანტა და გავაგრძელოთ სხვა იმპლიკანტების ანალიზი. $x_0\bar{x}_1$ იმპლიკანტა ღებულობს მნიშვნელობა 1 ნაკრებზე $x_0 = 1,\ x_1 = 0$. ჩავსვათ ეს ნაკრები ჯამში $\bar{x}_0x_2 \vee \bar{x}_1x_2 \vee x_0\bar{x}_2$, მივიღებთ $x_2 \vee \bar{x}_2 = 1$, რაც იმაზე მეტყველებს, რომ $x_0\bar{x}_1$ იმპლიკანტა ზედმეტია.

ვტოვებთ მას და ვაგრძელებთ პროცედურას. $x_0\bar{x}_2$ იმპლიკანტა ღებულობს მნიშვნელობა 1 ნაკრებზე $x_0 = 1,\ x_2 = 0$. ჩავსვათ ეს ნაკრები ჯამში $\bar{x}_0x_2 \vee \bar{x}_1x_2 \vee x_0\bar{x}_1$, მივიღებთ $\bar{x}_1$, რაც იმაზე მეტყველებს, რომ $x_0\bar{x}_2$ იმპლიკანტა ზედმეტი არაა.

როგორც პირველ შემთხვევაში ახლაც არ შეიძლება ორივე აღმოჩენილი ზედმეტი იმპლიკანტის გადაგდება, ვინაიდან თითოეული მათგანი მოწმდებოდა მეორის დარჩენილ ჯამში არსებობის დროს. გამოვტოვოთ მომდევნო პროცედურების განხილვა, რადგანაც ისინი ანალოგიურებია პირველი ხერხის დროს შესრულებული პროცედურების.

შეიძლება დავასკვნათ, რომ ასეთი მარტივი მაგალითის დროსაც კი მოგვიხდა საკმაოდ ბევრი ერთგვაროვანი მოქმედების შესრულება, რომლებიც მოითხოვენ ყურადღებასა და დროს, ამიტომ ლაფის ალგებრული მინიმიზაციის მეთოდი ძირითადად გამოიყენება ისეთი ლაფ-ებისათვის, რომლებიც დამოკიდებულებია ორ ან სამ ცვლადზე.

და ბოლოს, მაგალითებზე ვაჩვენოთ ლაფ-ის გამარტივების რამდენიმე ხერხი. აქვე შევნიშნოთ, რომ ლოგიკურ ფორმულათა ზოგიერთი გარდაქმნა მსგავსია ჩვეულებრივ ალგებრაში ფორმულათა გარდაქმნის (საერთო მამრავლის ფრჩხილებს გარეთ გატანა, კომუტატიურობისა და ასოციაციურობის კანონების გამოყენება და ა. შ.), მაშინ როცა სხვა გარდაქმნები დაფუძნებულია ისეთ თვისებებზე, რომლებიც ჩვეულებრივი ალგებრის ოპერაციებს არ გააჩნიათ (კონიუნქციისათვის დისტრიბუტიულობის კანონის, შთანთქმის, შეწებების, დე მორგანის და სხვა კანონების გამოყენება).

მ ა გ ა ლ ი თ ე ბ ი :

1. $\overline{x \vee y} \cdot (x \cdot \bar{y}) = \bar{x} \cdot \bar{y} \cdot (x \cdot \bar{y}) = \bar{x} \cdot x \cdot \bar{y} \cdot \bar{y} = 0 \cdot \bar{y} \cdot \bar{y} = 0 \cdot \bar{y} = 0$
(ალგებრის ლოგიკის კანონები გამოიყენებიან შემდეგი თანმიმდევრობით: დე მორგანის კანონები, ასოციაციურობის კანონი, ცვლადთან და მის ინვერსიასთან ოპერაციათა წესი და კონსტანტებთან ოპერაციათა წესი);

2. $\bar{x} \cdot y \vee \overline{x \vee y} \vee x = \bar{x} \cdot y \vee \bar{x} \cdot \bar{y} \vee x = \bar{x} \cdot (y \vee \bar{y}) \vee x = \bar{x} \vee x = 1$
(გამოიყენება დე მორგანის წესი, ფრჩხილებს გარეთ გამოდის საერთო მამრავლი, გამოიყენება ცვლადთან და მის ინვერსიასთან ოპერაციათა წესი);

3. $(x \vee y) \cdot (\bar{x} \vee y) \cdot (\bar{x} \vee \bar{y}) = (x \vee y) \cdot (\bar{x} \vee y) \cdot (\bar{x} \vee y) \cdot (\bar{x} \vee \bar{y}) = y \cdot \bar{x}$
(მეორდება მეორე მამრავლი, რაც დაშვებულია იდემპოტენციურობის კანონით; შემდეგ კომბინირდება პირველი ორი და ორი ბოლო თანამამრავლი და გამოიყენება შეწებების წესი).

4. $$x \cdot \bar{y} \vee \bar{x} \cdot y \cdot z \vee x \cdot z = x \cdot \bar{y} \vee \bar{x} \cdot y \cdot z \vee x \cdot z \cdot (y \vee \bar{y}) =$$
$$= x \cdot \bar{y} \vee \bar{x} \cdot y \cdot z \vee x \cdot y \cdot z \vee x \cdot \bar{y} \cdot z =$$
$$= (x \cdot \bar{y} \vee x \cdot \bar{y} \cdot z) \vee (\bar{x} \cdot y \cdot z \vee x \cdot y \cdot z) = x \cdot \bar{y} \vee y \cdot z$$
(შემოიტანება დამატებითი ლოგიკური მამრავლი $(y \vee \bar{y})$; შემდეგ კომბინირდება ორი კიდურა და ორი შუა ლოგიკური შესაკრები და გამოიყენება შთანთქმის კანონი);

5. $\overline{x \cdot y \vee \bar{z}} = \overline{x \cdot y} \cdot \bar{\bar{z}} = (\bar{x} \vee \bar{y}) \cdot z$

(დასაწყისში ვცდილობთ, რომ უარყოფის ნიშანი იდგეს მხოლოდ ცალკეულ ცვლადებთან და არა მათ კომბინაციებთან; ამისათვის ორჯერ გამოიყენება დე მორგანის წესი; შემდეგ გამოიყენება ორმაგი უარყოფის წესი);

6. $x \cdot y \vee x \cdot y \cdot z \vee x \cdot z \cdot p = x \cdot (y \cdot (1 \vee z) \vee z \cdot p) = x \cdot (y \vee z \cdot p)$

(ფრჩხილებს გარეთ გამოდიან საერთო მამრავლები; გამოიყენება კონსტანტებთან ოპერაციების წესი);

7. $x \vee \overline{y \cdot \bar{z}} \vee \overline{\bar{x} \vee y \vee \bar{z}} = x \vee \bar{y} \vee \bar{\bar{z}} \vee \bar{\bar{x}} \cdot \bar{y} \cdot \bar{\bar{z}} = x \vee \bar{y} \vee z \vee x \cdot \bar{y} \cdot z =$

$$= x \vee z \vee (\bar{y} \vee x \cdot \bar{y} \cdot z) = x \vee z \vee \bar{y}$$

(არაელემენტარული ფორმულების უარყოფებისადმი გამოიყენება დე მორგანის წესი; გამოიყენება ორმაგი უარყოფისა და შეწებების წესები);

8. $x \cdot \bar{y} \vee x \cdot y \cdot z \vee x \cdot \bar{y} \cdot z \vee x \cdot \overline{y \cdot z} = x \cdot (\bar{y} \vee y \cdot z \vee \bar{y} \cdot z \vee \overline{y \cdot z}) =$

$$= x \cdot \big((\bar{y} \vee \bar{y} \cdot z) \vee (y \cdot z \vee \overline{y \cdot z})\big) = x \cdot (\bar{y} \vee \bar{y} \cdot z \vee 1) = x \cdot 1 = x$$

(საერთო მამრავლი x გამოდის ფრჩხილებს გარეთ, ფრჩხილებში ხდება შესაკრებთა კომბინირება-პირველისა მესამესთან და მეორისა მეოთხესთან, $y \cdot z \vee \overline{y \cdot z}$ დიზიუნქციის მიმართ გამოიყენება ცვლადთან და მის ინვერსიასთან ოპერაციათა წესი);

9. $(x \cdot \bar{y} \vee z) \cdot (\bar{x} \vee y) \vee \bar{z} = x \cdot \bar{y} \cdot \bar{x} \vee x \cdot \bar{y} \cdot y \vee z \cdot \bar{x} \vee z \cdot y \vee \bar{z} =$

$$= 0 \vee 0 \vee z \cdot \bar{x} \vee z \cdot y \vee \bar{z} = z \cdot \bar{x} \vee (z \vee \bar{z}) \cdot (y \vee \bar{z}) = z \cdot \bar{x} \vee 1 \cdot (y \vee \bar{z}) =$$
$$= z \cdot \bar{x} \vee y \vee \bar{z} = (z \cdot \bar{x} \vee \bar{z}) \vee y = (z \vee \bar{z}) \cdot (\bar{x} \vee \bar{z}) \vee y = 1 \cdot (\bar{x} \vee \bar{z}) \vee y =$$
$$= \bar{x} \vee \bar{z} \vee y$$

(გამოიყენება დისტრიბუტიულობის კანონი დიზიუნქციისათვის, ცვლადთან და მის ინვერსიასთან ოპერაციათა წესი, კონსტანტებთან ოპერაციათა წესი, კომუტატიურობის კანონი და დისტრიბუტიულობის კანონი კონიუნქციისათვის);

10. $x \cdot y \cdot (\bar{x} \cdot z \vee \overline{\overline{x \cdot y} \cdot z} \vee z \cdot t) = x \cdot y \cdot (\bar{x} \cdot z \vee \overline{\overline{x \cdot y}} \vee \bar{z} \vee z \cdot t) =$

$$= x \cdot y \cdot (\bar{x} \cdot z \vee x \cdot y \vee \bar{z} \vee z \cdot t) = x \cdot y \vee x \cdot y \cdot \bar{z} \vee x \cdot y \cdot z \cdot t = x \cdot y$$

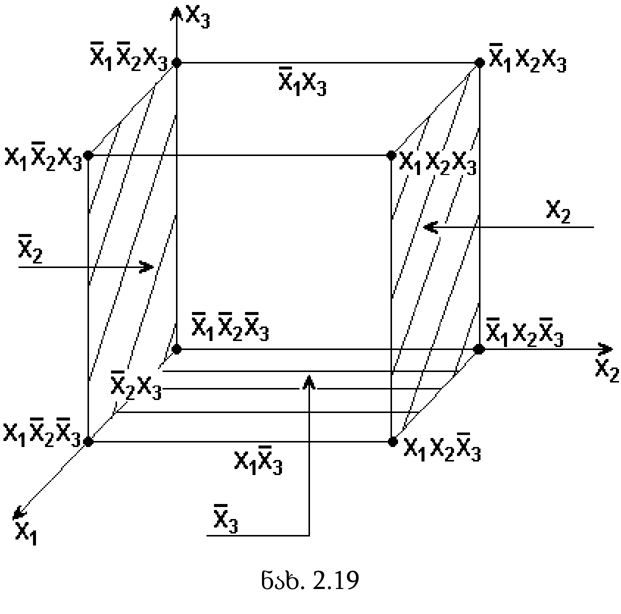
(გამოიყენება დე მორგანის წესი, ორმაგი უარყოფის კანონი და შთანთქმის კანონი).

ამ მაგალითებიდან ჩანს, რომ ლოგიკური ფორმულების გამარტივების დროს ყოველთვის არაა ცხადი, თუ ბულის ალგებრის რომელი კანონი უნდა გამოვიყენოთ ამა თუ იმ ნაბიჯზე. ჩვევები გამოცდილებით მოდის.

2.4.2. გეომეტრიული მეთოდი.

ძირითადად ეს მეთოდი გამოიყენება სამი ცვლადის ფუნქციათა მინიმიზაციისათვის, მაგრამ შესაძლებელია მისი განზოგადება ცვლადების ნებისმიერი რაოდენობისათვის. სამი ცვლადისათვის მეთოდი თვალსაჩინოა, ამიტომ ჩვენ შემოვისაზღვრებით მხოლოდ ამ შემთხვევით.

გამოვსახოთ სამი ცვლადის ნებისმიერი ბულის ფუნქციის განსაზღვრის არე - ესაა სამგანზომილებიანი კუბის წვეროები. შეიძლება დავამყაროთ კუბის ელემენტებსა და სხვადასხვა რანგის კონიუნქციებს შორის ურთიერთ-ცალსახა შესაბამისობა: კუბის წვეროები - მესამე რანგის კონიუნქციები, წიბოები - მეორე რანგის, ხოლო წახნაგები - პირველი რანგის.



ნახ. 2.19

ნაკლები განზომილების თითოეული გეომეტრიული ექვივალენტი იფარება უფრო დიდი განზომილების ყველა გეომეტრიული ექვივალენტებით. დიდი რანგის კონიუნქციები იფარებიან ნაკლები რანგის მქონე კონიუნქციებით (იხ. ნახ. 2.19).

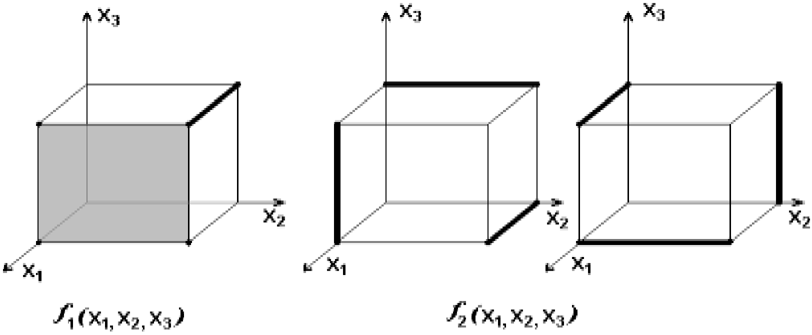
ასე, მაგალითად, $x_1x_2x_3$ და $x_1x_2\overline{x}_3$ იფარებიან x_1x_2 კონიუნქციით (ორი წვერო - წიბო);

$x_1\overline{x}_2\overline{x}_3$, $x_1\overline{x}_2x_3$, $\overline{x}_1\overline{x}_2x_3$, $\overline{x}_1\overline{x}_2\overline{x}_3$ კონიუნქციები იფარებიან ან ორი კონიუნქციით $x_1\overline{x}_2$ და $\overline{x}_1\overline{x}_2$, ან მხოლოდ $\overline{x}_2$-ით (ოთხი წვერო - ან ორი წიბო, ან ერთი წახნაგი).

ბულის ფუნქცია მოიცემა თავისი წვეროების სიმრავლით, ე. ი. ერთეულოვან მნიშვნელობათა სიმრავლით.

ბულის ფუნქცია მოიცემა თავისი წვეროების სიმრავლით, ანუ ერთეულოვან მნიშვნელობათა სიმრავლით. ფუნქციის ჩანაწერი რაღაც დნფ-ში შეესაბამება $T=\sum_{i=1}^{m} J_{r_i}$ დაფარვის პოვნას, სადაც r_i - დაფარვის ინტერვალების რანგებია. მინიმალური დნფ-ს პოვნის ამოცანა შეესაბამება ისეთი T დაფარვის პოვნას, რომელშიც ინტერვალით დაფარვის ყველა რანგების ჯამი მინიმალურია, ანუ $R=\sum_{i=1}^{m} r_i$ - მინიმალურია, რადგან r_i რანგი ემთხვევა J_{r_i} -ში შემავალ ასოთა რიცხვს.

$f_1(x_1,x_2,x_3)$ და $f_2(x_1,x_2,x_3)$ ფუნქციებისათვის ნახ. 2.20-ზე მოყვანილ ნახაზებზე მინიმალური ფორმები იქნებიან $f_1=x_1\vee x_2x_3$, $f_2=x_1\overline{x}_2\vee\overline{x}_1x_3\vee x_2\overline{x}_3$ ან $f_2=x_1\overline{x}_3\vee\overline{x}_1x_2\vee\overline{x}_2x_3$. მეორე ფუნქციისათვის ამოცანა ცალსახად არ იხსნება.



ნახ. 2.20

მ ა გ ა ლ ი თ ი 1: მოვახდინოთ ჭეშმარიტების ცხრილით (ცხრილ 2.17) წარმოდგენილი ფუნქციის მინიმიზაცია.

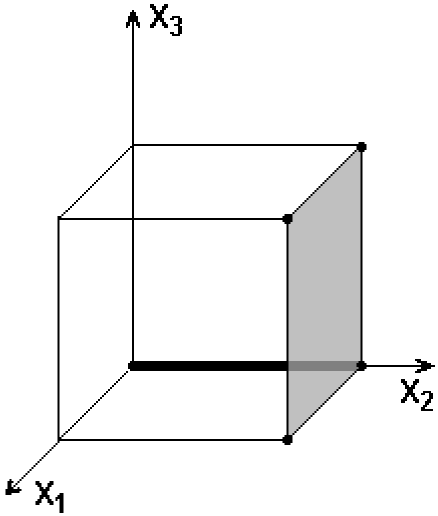
ცხრილი 2.17

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

მის გამოსახულებას სდნფ-ში შემდეგი სახე აქვს:

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3$$

ნახაზზე ავღნიშნოთ ის წვეროები, რომლებიც შეესაბამებიან მოცემული ფუნქციის სდნფ-ში შემავალ კონიუნქციებს (ნახ. 2.21). ავღნიშნოთ, რომ ოთხივე წვერო დევს ერთ წახნაგზე (x_2), ხოლო ორი ერთ წიბოზე ($\bar{x}_1\bar{x}_3$). საიდანაც გამომდინარეობს, რომ ფუნქციის მინიმალური ფორმა სწორედ არის ამ ინტერვალთა ჯამი $x_2 \vee \bar{x}_1\bar{x}_3$, ანუ $f(x_1, x_2, x_3) = x_2 \vee \bar{x}_1\bar{x}_3$. ამოხსნის სხვა ვარიანტი არ შეიძლება იყოს. ამოცანა ცალსახად იხსნება.



ნახ. 2.21

2.4.3. განუსაზღვრელ კოეფიციენტთა მეთოდი.

ეს მეთოდი შეიძლება გამოყენებულ იქნას ცვლადების ნებისმიერი რაოდენობის მქონე ბულის ფუნქციებისათვის, ოღონდ, მისი აღწერის სიმარტივისათვის განვიხილოთ სამ ცვლადზე დამოკიდებული ფუნქციის მინიმიზაცია.

წარმოვიდგინოთ $f(x_1,x_2,x_3)$ ფუნქცია შემდეგი დნფ-ს სახით:

$$\begin{aligned} f(x_1x_2x_3) = {} & K_1^1x_1 \vee K_1^0\overline{x}_1 \vee K_2^1x_2 \vee K_2^0\overline{x}_2 \vee K_3^1x_3 \vee K_3^0\overline{x}_3 \vee \\ & \vee K_{12}^{11}x_1x_2 \vee K_{12}^{10}x_1\overline{x}_2 \vee K_{12}^{01}\overline{x}_1x_2 \vee K_{12}^{00}\overline{x}_1\overline{x}_2 \vee \\ & \vee K_{13}^{11}x_1x_3 \vee K_{13}^{10}x_1\overline{x}_3 \vee K_{13}^{01}\overline{x}_1x_3 \vee K_{13}^{00}\overline{x}_1\overline{x}_3 \vee \\ & \vee K_{23}^{11}x_2x_3 \vee K_{23}^{10}x_2\overline{x}_3 \vee K_{23}^{01}\overline{x}_2x_3 \vee K_{23}^{00}\overline{x}_2\overline{x}_3 \vee \\ & \vee K_{123}^{111}x_1x_2x_3 \vee K_{123}^{110}x_1x_2\overline{x}_3 \vee K_{123}^{101}x_1\overline{x}_2x_3 \vee K_{123}^{100}x_1\overline{x}_2\overline{x}_3 \vee \\ & \vee K_{123}^{011}\overline{x}_1x_2x_3 \vee K_{123}^{010}\overline{x}_1x_2\overline{x}_3 \vee K_{123}^{001}\overline{x}_1\overline{x}_2x_3 \vee K_{123}^{000}\overline{x}_1\overline{x}_2\overline{x}_3 \end{aligned}$$

აქ წარმოდგენილია ყველა შესაძლო კონიუნქციური წევრები, რომლებიც შეიძლება შედიოდნენ $f(x_1,x_2,x_3)$ ფუნქციაში. სხვადასხვა ინდექსების მქონე K კოეფიციენტები განუსაზღვრელებია და ისე შეირჩევიან, რომ მიღებული ფორმა მინიმალური იყოს. თუ მივუთითებთ არგუმენტების ნაკრებებს, ჩავსვამთ ფორმულაში და გავუტოლებთ მიღებულ გამოსახულებებს (ნულოვანი კონიუნქცების გადაგდებით) ფუნქციის მნიშვნელობას არჩეულ ნაკრებებზე, მაშინ K კოეფიციენტების განსასაზღვრავად მივიღებთ განტოლებათა სისტემას. ზოგად შემთხვევაში სისტემაში 2^n განტოლება იქნება (n - ფუნქციის არგუმენტთა რიცხვია.

თუ $f(x_1,x_2,x_3)$ ფუნქცია ცხრილის სახითაა წარმოდგენილი, მაშინ შესაბამისი განტოლებების მარჯვენა ნაწილში ნოლები და ერთიანები იქნებიან. იმ განტოლების დასაკმაყოფილებლად, რომლის მარჯვენა ნაწილში დგას ნული, აუცილებელია ნულს გავუტოლოთ

განტოლებაში შემავალი ყველა ყველა K კოეფიციენტი (ეს დიზიუნქციის განმარტებიდან გამომდინარეობს).

$$(2.9)\quad \begin{cases} K_1^1 \vee K_2^1 \vee K_3^1 \vee K_{12}^{11} \vee K_{13}^{11} \vee K_{23}^{11} \vee K_{123}^{111} = f(1,1,1) \\ K_1^1 \vee K_2^1 \vee K_3^0 \vee K_{12}^{11} \vee K_{13}^{10} \vee K_{23}^{10} \vee K_{123}^{110} = f(1,1,0) \\ K_1^1 \vee K_2^0 \vee K_3^1 \vee K_{12}^{10} \vee K_{13}^{11} \vee K_{23}^{01} \vee K_{123}^{101} = f(1,0,1) \\ K_1^1 \vee K_2^0 \vee K_3^0 \vee K_{12}^{10} \vee K_{13}^{10} \vee K_{23}^{00} \vee K_{123}^{100} = f(1,0,0) \\ K_1^0 \vee K_2^1 \vee K_3^1 \vee K_{12}^{01} \vee K_{13}^{01} \vee K_{23}^{11} \vee K_{123}^{011} = f(0,1,1) \\ K_1^0 \vee K_2^1 \vee K_3^0 \vee K_{12}^{01} \vee K_{13}^{00} \vee K_{23}^{10} \vee K_{123}^{010} = f(0,1,0) \\ K_1^0 \vee K_2^0 \vee K_3^1 \vee K_{12}^{00} \vee K_{13}^{01} \vee K_{23}^{01} \vee K_{123}^{001} = f(0,0,1) \\ K_1^0 \vee K_2^0 \vee K_3^0 \vee K_{12}^{00} \vee K_{13}^{00} \vee K_{23}^{00} \vee K_{123}^{000} = f(0,0,0) \end{cases}$$

ყველა იმ განტოლების განხილვის შემდეგ, რომელთა მარჯვენა ნაწილში ნულები დგანან, და ამ განტოლებების ყველა კოეფიციენტის ნულთან გატოლებით, დანარჩენ განტოლებებში ვშლით მათში შესულ ნულოვან კოეფიციენტებს. მოხერხებულია მიღებული სისტემის გადაწერა უფრო შემოკლებულ ფორმაში, მასში იმ განტოლებათა დატოვებით, რომელთა მარჯვენა ნაწილშიც ერთიანები დგანან, ამასთან ამ განტოლებებიდან ნულოვანი კოეფიციენტების წაშლა. შემდეგ სისტემაში ვირჩევთ ყველაზე მოკლე განტოლებებს. ამ განტოლებებში ერთიანებს ვუტოლებთ იმ კოეფიციენტებს, რომლებიც განსაზღვრავენ შესაძლოდ უმცირესი რანგის კონიუნქციებს (ეს შესაძლებელია, ვინაიდან დიზიუნქცია ერთის ტოლია თუნდაც ერთი წევრის ერთიანად ქცევისას). ამასთან უმცირესი რანგის ისეთი კონიუნქციები უნდა შევარჩიოთ, რომლებიც უფრო ხშირად გვხვდებიან სისტემის განტოლებებში. დანარჩენი კოეფიციენტები შეიძლება 0 ან 1-ის ტოლი დავტოვოთ. შემდეგ ვათვალიერებთ დარჩენილ განტოლებებს და მათში ვირჩევთ კოეფიციენტებს, რომლებიც შეესაბამებიან უმცირესი რანგის კონიუნქციებს იგივე პრინციპით და ა. შ.

ნაპოვნი ერთეულოვანი კოეფიციენტები განსაზღვრავენ კონიუნქციებს ასოების უმცირესი რიცხვით, ხოლო ამ კოეფიციენტებით ჩაწერილი ფორმა განსაზღვრავს მოცემული ფუნქციის მინიმალურ დნფ-ს.

მ ა გ ა ლ ი თ ი 2: მოვახდინოთ ფუნქციის მინიმიზაცია (იხ. მაგ.1). $f(x_1,x_2,x_3)=\bar{x}_1\bar{x}_2\bar{x}_3\vee\bar{x}_1x_2\bar{x}_3\vee\bar{x}_1x_2x_3\vee x_1x_2\bar{x}_3\vee x_1x_2x_3$

შევადგინოთ სისტემა (ყურადღება მიაქციეთ იმას, რომ მას სტანდარტული სახე აქვს, მხოლოდ მარჯვენა ნაწილში იცვლებიან მნიშვნელობები ფუნქციის ჭეშმარიტების ცხრილისაგან დამოკიდებულებით). სისტემის ჩაწერის მოხერხებულობისათვის მარცხნიდან ათავსებენ წვეროების კოორდინატებს (ფუნქციის განსაზღვრის არე). ვახდენთ კოეფიციენტების ზედა ინდექსების კომბინირებას წვეროების ჩაწერილი კოორდინატების შესაბამისად აღებული ქვედა ინდექსების გათვალისწინებით. მაგალითად, მეორე წვეროსათვის (0,0,1) K_{12} კოეფიციენტისათვის ზედა ინდექსი იქნება 00; K_{13} - ისათვის 01 და ა. შ.

$$
\begin{array}{c|l}
000 & K_1^0\vee K_2^0\vee K_3^0\vee K_{12}^{00}\vee K_{13}^{00}\vee K_{23}^{00}\vee K_{123}^{000}=1\\
001 & K_1^0\vee K_2^0\vee K_3^1\vee K_{12}^{00}\vee K_{13}^{01}\vee K_{23}^{01}\vee K_{123}^{001}=0\\
010 & K_1^0\vee K_2^1\vee K_3^0\vee K_{12}^{01}\vee K_{13}^{00}\vee K_{23}^{10}\vee K_{123}^{010}=1\\
011 & K_1^0\vee K_2^1\vee K_3^1\vee K_{12}^{01}\vee K_{13}^{01}\vee K_{23}^{11}\vee K_{123}^{011}=1\\
100 & K_1^1\vee K_2^0\vee K_3^0\vee K_{12}^{10}\vee K_{13}^{10}\vee K_{23}^{00}\vee K_{123}^{100}=0\\
101 & K_1^1\vee K_2^0\vee K_3^1\vee K_{12}^{10}\vee K_{13}^{11}\vee K_{23}^{01}\vee K_{123}^{101}=0\\
110 & K_1^1\vee K_2^1\vee K_3^0\vee K_{12}^{11}\vee K_{13}^{10}\vee K_{23}^{10}\vee K_{123}^{110}=1\\
111 & K_1^1\vee K_2^1\vee K_3^1\vee K_{12}^{11}\vee K_{13}^{11}\vee K_{23}^{11}\vee K_{123}^{111}=1
\end{array}
$$

2, 5, 6 განტოლებებიდან დიზიუნქციის თვისებიდან გამომდინარეობს, რომ

$$K_1^0=K_2^0=K_3^0=K_1^1=K_3^1=K_{12}^{00}=K_{12}^{10}=K_{13}^{01}=K_{13}^{10}=K_{13}^{11}=K_{23}^{00}=K_{23}^{01}=K_{123}^{101}=K_{123}^{001}=K_{123}^{100}=0$$

მოსახერხებელია წავშალოთ განტოლება, რომლის მარჯვენა ნაწილში ნულებია, ხოლო დანარჩენ განტოლებებში წავშალოთ ნულის ტოლი კოეფიციენტები.

ამის შემდეგ სისტემა მიიღებს შემდეგ სახეს:

$$(2.10)\quad \begin{cases} K_{13}^{00} \vee K_{123}^{000} = 1 \\ K_2^1 \vee K_{12}^{01} \vee K_{13}^{00} \vee K_{23}^{10} \vee K_{123}^{010} = 1 \\ K_2^1 \vee K_{12}^{01} \vee K_{23}^{11} \vee K_{123}^{011} = 1 \\ K_2^1 \vee K_{12}^{11} \vee K_{23}^{10} \vee K_{123}^{110} = 1 \\ K_2^1 \vee K_{12}^{11} \vee K_{23}^{11} \vee K_{123}^{111} = 1 \end{cases}$$

სისტემა (2.10)-ში დიზიუნქციის თვისების თანახმად ($1 \vee A = 1$) შეიძლება ერთს გავუტოლოთ კოეფიციენტი $K_2^1 = 1$, მაშინ ამ სისტემის 2, 3, 4 და 5 განტოლებები იგივეობები ხდებიან, სისტემის პირველი განტოლებიდან კი ავიღოთ $K_{13}^{00} = 1$. ყველა განტოლებაში დანარჩენი კოეფიციენტები ნულს გავუტოლოთ.

$$K_{12}^{01} = K_{12}^{11} = K_{13}^{00} = K_{23}^{10} = K_{23}^{11} = K_{123}^{000} = K_{123}^{010} = K_{123}^{011} = K_{123}^{110} = K_{123}^{111} = 0$$

შ ე ნ ი შ ვ ნ ა: ყურადღება მიაქციეთ იმ ფაქტს, რომ ერთიანს უტოლებენ კოეფიციენტებს, რომლებიც პასუხობენ კონიუნქციებს, შეიცავენ ცვლადების უმცირეს რაოდენობას; გარდა ამისა, ყველაზე ხშირად გვხვდებიან განტოლებათა გამარტივებულ სისტემაში.

ამგვარად, ჩვენ ვიპოვეთ $K_2^1 = 1,\quad K_{13}^{00} = 1$, დანარჩენი კოეფიციენტები ნულის ტოლია. აქედან მოცემული ფუნქციის მინიმალური ფორმა:

$$f(x_1, x_2, x_3) = x_2 \vee \bar{x}_1 \bar{x}_3$$

ეს მეთოდი შრომატევადია, ნაკლებად გამოიყენება, მაგრამ ჩვენ ის განვიხილეთ მომდევნო მეთოდების დასაბუთების მიზნით.

2.4.4. კარნოს ბარათების მეთოდი.

ეს მეთოდი არსებითად წარმოადგენს იგივე განუსაზღვრელ კოეფიციენტთა მეთოდს, მხოლოდ ჩაწერილს უფრო მოხერხებულ ფორმაში.

განვიხილოთ შემდეგი ცხრილი.

ცხრილი 2.18

x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$
x_1	x_2	$\bar{x}_3$	x_1x_2	$x_1\bar{x}_3$	$x_2\bar{x}_3$	$x_1x_2\bar{x}_3$
x_1	$\bar{x}_2$	x_3	$x_1\bar{x}_2$	x_1x_3	$\bar{x}_2x_3$	$x_1\bar{x}_2x_3$
x_1	$\bar{x}_2$	$\bar{x}_3$	$x_1\bar{x}_2$	$x_1\bar{x}_3$	$\bar{x}_2\bar{x}_3$	$x_1\bar{x}_2\bar{x}_3$
$\bar{x}_1$	x_2	x_3	$\bar{x}_1x_2$	$\bar{x}_1x_3$	x_2x_3	$\bar{x}_1x_2x_3$
$\bar{x}_1$	x_2	$\bar{x}_3$	$\bar{x}_1x_2$	$\bar{x}_1\bar{x}_3$	$x_2\bar{x}_3$	$\bar{x}_1x_2\bar{x}_3$
$\bar{x}_1$	$\bar{x}_2$	x_3	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_3$	$\bar{x}_2x_3$	$\bar{x}_1\bar{x}_2x_3$
$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1\bar{x}_3$	$\bar{x}_2\bar{x}_3$	$\bar{x}_1\bar{x}_2\bar{x}_3$

ეს ცხრილი ემსახურება განუსაზღვრელ კოეფიციენტთა მეთოდის განტოლებათა (2.9) სისტემის უფრო კომპაქტურ ჩაწერას, სადაც K კოეფიციენტების ნაცვლად შესაბამის უჯრედში თავად კონიუნქციების ჩაწერა ხდება. ცხრილ 2.18-ის თითოეული სტრიქონი ცვლის შესაბამისად (2.9) სისტემის პირველ, მე-2, ... , მე-8 განტოლებას. ცხრილის სტრიქონის ყველა ელემენტთა დიზიუნქცია არის ფუნქციის მნიშვნელობა შესაბამისი ცვლადებით განსაზღვრულ წვეროში. ასე, მაგალითად, პირველი სტრიქონი არის ფუნქციის მნიშვნელობა (x_1, x_2, x_3) წვეროში, მეოთხე $(x_1, \bar{x}_2, \bar{x}_3)$ წვეროში ან კოორდინატებზე გადაყვანით შესაბამისად (1, 1, 1), (1, 0, 0) - ში.

შეიძლება ვაჩვენოთ, რომ თუ მოცემული ფუნქციის სდნფ-ში არ შედის ბოლო სვეტის რვა კონიუნქციიდან რომელიმე, მაშინ ამ ფუნქციის მინიმალურ ფორმაში არ შეიძლება შედიოდეს ცხრილის შესაბამისი სტრიქონის კონიუნქციებიდან არც ერთი.

დავუშვათ, მაგალითად, რომ სდნფ-ში არ შედის $x_1x_2x_3$ კონიუნქცია, მაშინ მინიმალურ ფორმაში არ შედის, მაგალითად, x_1 წევრი (ანალოგიურად მე-3 სტრიქონის სხვა კონიუნქციებიც):

$$x_1 = \underline{x_1x_2} \vee \underline{\underline{x_1\bar{x}_2}} = \underline{x_1x_2x_3 \vee x_1x_2\bar{x}_3} \vee \underline{\underline{x_1\bar{x}_2x_3 \vee x_1\bar{x}_2\bar{x}_3}}\,.$$

ამრიგად, თუ მინიმალურ ფორმაში შევიდოდა x_1 წევრი, მაშინ მასში აუცილებლად იქნებოდა $x_1\bar{x}_2x_3$ წევრიც, რაც დაშვებას ეწინააღმდეგება.

ცხრილ 2.18-ს მინიმიზირების ბარათი ჰქვია. ჩვეულებრივ ეს ბარათები დაბეჭდილია ცვლადების შესაბამისი რიცხვისთვის.

ფუნქციის მინიმიზაცია შემდეგი წესებით ხორციელდება:

1. ცხრილის ყველა სტრიქონი, რომელიც შეესაბამება ბოლო სვეტის კონიუნქციებს და არ არიან მოცემული ფუნქციის სდნფ-ში, წაიშლება.
2. დარჩენილი სტრიქონების სვეტებში ახდენენ ყველა იმ ელემენტის წაშლას, რომელიც წაშლილ სტრიქონებში მოხვდა.
3. თითოეულ წაუშლელ სტრიქონში ირჩევენ წაუშლელ კონიუნქციას, რომელიც ასოთა მინიმალურ რიცხვს შეიცავს (სასურველია, რომ არჩეული კონიუნქციები უფრო ხშირად გვხდებოდნენ ყველა დარჩენილ სტრიქონში).
4. აიღებენ რა თითო კონიუნქციას ყველა წაუშლელი სტრიქონიდან და ჩაწერენ მათ დიზიუნქციას, მიიღებენ მინიმალურ ფორმას.

შევნიშნოთ, რომ მდნფ-ს მიღება ცალსახა არაა, ვინაიდან ნებისმიერია ნებისმიერია სტრიქონებში მინიმალური კონიუნქციების

არჩევა. ოღონდ, ამ მეთოდით მიღებული ყველა მდნფ იქნება „ერთნაირად მინიმალური“.

მ ა გ ა ლ ი თ ი 3: მოვახდინოთ ფუნქციის მინიმიზაცია (იხ. მაგ.1).

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3$$

ფუნქციისათვის ვაგებთ მინიმიზირების ბარათს

$\cancel{x_1}$	$\boxed{x_2}$	$\cancel{x_3}$	$\boxed{x_1 x_2}$	$\cancel{x_1 x_3}$	$\boxed{x_2 x_3}$	$\boxed{x_1 x_2 x_3}$
$\cancel{x_1}$	$\boxed{x_2}$	$\cancel{\bar{x}_3}$	$\boxed{x_1 x_2}$	$\cancel{x_1 \bar{x}_3}$	$\boxed{x_2 \bar{x}_3}$	$\boxed{x_1 x_2 \bar{x}_3}$
$\cancel{x_1}$	$\cancel{\bar{x}_2}$	$\cancel{x_3}$	$\cancel{x_1 \bar{x}_2}$	$\cancel{x_1 x_3}$	$\cancel{\bar{x}_2 x_3}$	$\cancel{x_1 \bar{x}_2 x_3}$
$\cancel{x_1}$	$\cancel{\bar{x}_2}$	$\cancel{\bar{x}_3}$	$\cancel{x_1 \bar{x}_2}$	$\cancel{x_1 \bar{x}_3}$	$\cancel{\bar{x}_2 \bar{x}_3}$	$\cancel{x_1 \bar{x}_2 \bar{x}_3}$
$\cancel{\bar{x}_1}$	$\boxed{x_2}$	$\cancel{x_3}$	$\boxed{\bar{x}_1 x_2}$	$\cancel{\bar{x}_1 x_3}$	$\boxed{x_2 x_3}$	$\boxed{\bar{x}_1 x_2 x_3}$
$\cancel{\bar{x}_1}$	$\boxed{x_2}$	$\cancel{\bar{x}_3}$	$\boxed{\bar{x}_1 x_2}$	$\boxed{\bar{x}_1 \bar{x}_3}$	$\boxed{x_2 \bar{x}_3}$	$\boxed{\bar{x}_1 x_2 \bar{x}_3}$
$\cancel{\bar{x}_1}$	$\cancel{\bar{x}_2}$	$\cancel{x_3}$	$\cancel{\bar{x}_1 \bar{x}_2}$	$\cancel{\bar{x}_1 x_3}$	$\cancel{\bar{x}_2 x_3}$	$\cancel{\bar{x}_1 \bar{x}_2 x_3}$
$\cancel{\bar{x}_1}$	$\cancel{\bar{x}_2}$	$\cancel{\bar{x}_3}$	$\cancel{\bar{x}_1 \bar{x}_2}$	$\boxed{\bar{x}_1 \bar{x}_3}$	$\cancel{\bar{x}_2 \bar{x}_3}$	$\boxed{\bar{x}_1 \bar{x}_2 \bar{x}_3}$

ბოლო სვეტიდან დაწყებული ავღნიშნოთ ის კონიუნქციები, რომლებიც შედიან მოცემული ფუნქციის სდნფ-ში. წავშალოთ აღუნიშნავი სტრიქონები (წესი 1), შემდეგ წავშალოთ დარჩენილ სტრიქონებში (ვმოქმედებთ სვეტით) ის ელემენტები, რომლებიც მოხვდნენ წაშლილ სტრიქონებში (წესი 2). მე-2 სვეტში (ერთი ცვლადით) დავუშვათ, რომ $x_2 = 1$, ამასთან სტრიქონთა დანარჩენი ელემენტები (1, 2, 5, 6 სტრიქონები), სადაც დგას x_2 ელემენტი, დავუშვათ ნულის

ტოლი. მე-8 სტრიქონში დავუშვათ, რომ ელემენტი $\bar{x}_1\bar{x}_3=1$, $\bar{x}_1\bar{x}_2\bar{x}_3=0$.

ამრიგად, მივიღეთ მოცემული ფუნქციის მდნფ:

$$f(x_1,x_2,x_3)=x_2\vee\bar{x}_1\bar{x}_3$$

შეადარეთ გეომეტრიული მეთოდით და განუსაზღვრელ კოეფიციენტთა მეთოდით მიღებულ შედეგებს.

მ ა გ ა ლ ი თ ი 4: მოვახდინოთ ფუნქციის მინიმიზაცია.

$$f(x_1,x_2,x_3)=x_1x_2\bar{x}_3\vee x_1\bar{x}_2x_3\vee x_1\bar{x}_2\bar{x}_3\vee\bar{x}_1x_2\bar{x}_3\vee\bar{x}_1\bar{x}_2x_3\vee\bar{x}_1x_2x_3$$

პირველი და მეორე წესების შესაბამისად გადავხაზოთ კონიუნქციები:

x_1	x_2	x_3	x_1x_2	x_1x_3	x_2x_3	$x_1x_2x_3$
x_1	x_2	$\bar{x}_3$	x_1x_2	$x_1\bar{x}_3$	$x_2\bar{x}_3$	$x_1x_2\bar{x}_3$
x_1	$\bar{x}_2$	x_3	$x_1\bar{x}_2$	x_1x_3	$\bar{x}_2x_3$	$x_1\bar{x}_2x_3$
x_1	$\bar{x}_2$	$\bar{x}_3$	$x_1\bar{x}_2$	$x_1\bar{x}_3$	$\bar{x}_2\bar{x}_3$	$x_1\bar{x}_2\bar{x}_3$
$\bar{x}_1$	x_2	x_3	$\bar{x}_1x_2$	$\bar{x}_1x_3$	x_2x_3	$\bar{x}_1x_2x_3$
$\bar{x}_1$	x_2	$\bar{x}_3$	$\bar{x}_1x_2$	$\bar{x}_1\bar{x}_3$	$x_2\bar{x}_3$	$\bar{x}_1x_2\bar{x}_3$
$\bar{x}_1$	$\bar{x}_2$	x_3	$\bar{x}_1\bar{x}_2$	$\bar{x}_1x_3$	$\bar{x}_2x_3$	$\bar{x}_1\bar{x}_2x_3$
$\bar{x}_1$	$\bar{x}_2$	$\bar{x}_3$	$\bar{x}_1\bar{x}_2$	$\bar{x}_1\bar{x}_3$	$\bar{x}_2\bar{x}_3$	$\bar{x}_1\bar{x}_2\bar{x}_3$

მოხერხებულობისათვის დარჩენილ კონიუნქციათა ცხრილი ცალკე ამოვხაზოთ, 1-3 სვეტებისა და 1, 8 სტრიქონების გადაგდებით.

	$\cancel{x_1\bar{x}_3}$	$x_2\bar{x}_3$	$\cancel{x_1x_2\bar{x}_3}$		$\vee$	
$x_1\bar{x}_2$		$\cancel{\bar{x}_2x_3}$	$\cancel{x_1\bar{x}_2x_3}$	$\vee$		
$x_1\bar{x}_2$	$\cancel{x_1\bar{x}_3}$		$\cancel{x_1\bar{x}_2\bar{x}_3}$	$\vee$		
$\bar{x}_1x_2$	$\bar{x}_1x_3$		$\bar{x}_1x_2x_3$			$\vee$
$\bar{x}_1x_2$		$x_2\bar{x}_3$	$\bar{x}_1x_2\bar{x}_3$			$\vee$
	$\bar{x}_1x_3$	$\cancel{\bar{x}_2x_3}$	$\cancel{\bar{x}_1\bar{x}_2x_3}$		$\vee$	

მე-2 სტრიქონში დავუშვათ $x_1\bar{x}_2$ 1-ის ტოლი, შემოვავლოთ ჩარჩო, დანარჩენი წევრები დავუშვათ ნულის ტოლი. გადავხაზოთ ნულოვანი წევრები $\bar{x}_2x_3$ მე-6 სტრიქონში, $x_1\bar{x}_3$ პირველ სტრიქონ-ში. ამოვირჩიოთ დარჩენილი სტრიქონებიდან ყველაზე მოკლეები, პირველი და მე-6 სტრიქონები. დავუშვათ მათში შესაბამისად $x_2\bar{x}_3 = 1,\quad \bar{x}_1x_3 = 1$, დანარჩენი წევრები კი ნულის ტოლი. მე-4 და მე-5 სტრიქონებში იქნება 1-ის ტოლი თითო-თითო წევრი. ამრიგად, ცხრილის თითოეულ სტრიქონში არის ერთის ტოლი თითო წევრი, გამომდინარე, ფუნქციის მინიმალური ფორმა იქნება:

$$f(x_1, x_2, x_3) = x_1\bar{x}_2 \vee x_2\bar{x}_3 \vee \bar{x}_1x_3$$

შესაძლებელია მინიმალური ფორმის მეორე ვარიანტი. გან-ვიხილოთ ცხრილზე.

	$\boxed{x_1\bar{x}_3}$	$\cancel{x_2\bar{x}_3}$	$x_1x_2\bar{x}_3$		$\vee$	
$x_1\bar{x}_2$		$\boxed{\bar{x}_2x_3}$	$x_1\bar{x}_2x_3$			$\vee$
$x_1\bar{x}_2$	$\boxed{x_1\bar{x}_3}$		$x_1\bar{x}_2\bar{x}_3$			$\vee$
$\boxed{\bar{x}_1x_2}$	$\cancel{\bar{x}_1x_3}$		$\cancel{\bar{x}_1x_2x_3}$	$\vee$		
$\boxed{\bar{x}_1x_2}$		$\cancel{x_2\bar{x}_3}$	$\cancel{\bar{x}_1x_2\bar{x}_3}$	$\vee$		
	$\cancel{\bar{x}_1x_3}$	$\boxed{\bar{x}_2x_3}$	$\bar{x}_1\bar{x}_2x_3$		$\vee$	

დავუშვათ მე-4 სტრიქონში $\bar{x}_1x_2 = 1$, ხოლო დანარჩენი წევრე-ბი ნულის ტოლია. მაშინ მე-5 სტრიქონში: $\bar{x}_1x_2 = 1$, $x_2\bar{x}_3$ და $\bar{x}_1x_2\bar{x}_3$ შეიძლება ნულის ტოლი დავუშვათ. გადავხაზოთ პირველ და მე-6 სტრიქონებში (ისინი სხვებზე მოკლეებია) $x_2\bar{x}_3$ და $\bar{x}_1x_3$, დავუშვათ შესაბამისად $x_1\bar{x}_3 = 1$, $\bar{x}_2x_3 = 1$. მაშინ მე-2 და მე-3 სტრიქონებში გვექნება ერთიანის ტოლი თითო-თითო წევრი. ამრიგად ფუნქციის მინიმალური ფორმა იქნება:

$$f(x_1, x_2, x_3) = \bar{x}_1x_2 \vee x_1\bar{x}_3 \vee \bar{x}_2x_3$$

განუსაზღვრელ კოეფიციენტთა და კარნოს ბარათებს მივყავართ უზარმაზარ ჩანაწერებთან (n ცვლადის ფუნქციისათვის ცხრილის ცტრიქონთა რაოდენობა 2^n-ის ტოლია, ხოლო სვეტების $2^n - 1$ - ის). ამ მეთოდების გამოყენება 8-10 რიგის n-ისათვის ძლიერ რთულდება.

2.4.5. ქუაინის მეთოდი.

ეს მეთოდი გამოიყენება იმ ფუნქციებისათვის, რომლებიც ჩაწერილია სდნფ-ში. ფუნქციის მინიმიზაციის მეთოდი ეტაპობრივად სრულდება.

<u>I ეტაპი. პირველადი იმპლიკანტების პოვნა.</u>

მოცემული ფუნქციის სდნფ-ს ყველა კონიუნქციას ადარებენ ერთმანეთს წყვილებად და იყენებენ შეწებების კანონს $xA \vee \bar{x}A = A$. მოხერხებულია ფუნქციის წევრების გადანომვრა, ცხრილში მოთავსება (იხ. ნახ. 2.22-ზე მოყვანილი ფორმა).

$f(x_1, \ldots, x_n)$ - ის წევრები	I შეწებების შედეგები	II შეწებების შედეგები	....
1.	1.	1.	
2.	2.	2.	
3.	3.	3.	
....	....	....	
....	....	....	
....	....	....	

ნახ.2.22

თანმიმდევრობით უნდა შედარდეს პირველი წევრი ყველა დანარჩენთან. შეწებების შედეგები უნდა ჩაიწეროს მე-2 სვეტში, შეწებებული წევრების ნუმერაციის ფრჩხილებში მითითებით, ხოლო პირველი სვეტის შეწებებული წევრები აღინიშნონ ვარსკლავით (*). მიღებული კონიუნქციების რანგი ერთით ნაკლებია, ე. ი. ისინი შეიცავენ ერთი ცვლადით ნაკლებს. ხდება ამ კონიუნქციათა გადანომრვა, შემდეგ ოპერაციას ისევ იმეორებენ, შედეგის მე-3 სვეტში ჩაწერით და ა. შ. ამ პროცედურას ასრულებენ მაშინ, როცა ახლად მიღე-

ბული კონიუნქციები უკვე აღარ წებდებიან ერთმანეთთან. * ნიშნით ყველა მოუნიშნავ კონიუნქციას პირველადს (მარტივ იმპლიკანტას) ეძახიან. ყველა წევრი, რომელიც * ნიშნითაა მონიშნული, შთაინთქმება მარტივი იმპლიკანტებით შთანთქმის ოპერაციის საფუძველზე ($A \vee AB = A$). მოხერხებულობისათვის ცხრილში მარტივ იმპლიკანტებს ჩარჩო უკეთდებათ.

ყველა მარტივი იმპლიკანტის დიზიუნქცია იძლევა მოცემული ფუნქციის შემოკლებულ დნფ-ს. შემდეგ საჭიროა ჩიხურ დნფ-ზე გადასვლა. ეს მე-2 ეტაპია.

მანამდე მაგალითზე განვიხილოთ პირველი ეტაპი.

მ ა გ ა ლ ი თ ი 5: მოვახდინოთ ფუნქციის მინიმიზაცია (იხ. მაგ.1).

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3$$

მოვათავსოთ $f(x_1, x_2, x_3)$ წევრები ცხრილის პირველ სვეტში, გადავნომროთ ისინი. გამოვიყენოთ შეწებების კანონი, შედეგები ჩავწეროთ ცხრილის მე-2 სცეტში, ისევ გადავნომროთ ისინი, პირველი სვეტის შეწებებული წევრები ავღნიშნოთ ვარსკვლავებით, და ა. შ. არა შეწებებულ მარტივ იმპლიკანტებს ჩარჩოში ვათავსებთ. მათი დიზიუნქცია გვაძლევს შემოკლებულ დნფ-ს.

ცხრილი 2.19

	$f(x_1, \ldots, x_n)$ -ის წევრები	I შეწებების შედეგები	II შეწებების შედეგები
1.	$\bar{x}_1 \bar{x}_2 \bar{x}_3$ *	$\boxed{\bar{x}_1 \bar{x}_3 \quad (1, 2)}$	$\boxed{x_2 \quad (2, 5)}$
2.	$\bar{x}_1 x_2 \bar{x}_3$ *	$\bar{x}_1 x_2$ * (2, 3)	$\cancel{x_2 \quad (3, 4)}$
3.	$\bar{x}_1 x_2 x_3$ *	$x_2 \bar{x}_3$ * (2, 4)	
4.	$x_1 x_2 \bar{x}_3$ *	$x_2 x_3$ * (3, 5)	
5.	$x_1 x_2 x_3$ *	$x_1 x_2$ * (4, 5)	

მოცემულ მაგალითში პირველ ეტაპს მიზანთან მივყავართ: $f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_3 \vee x_2$ ფუნქციის მინიმალური ფორმაა. ზოგად შემთხვევაში შემოკლებული ფორმიდან საჭიროა ჩიხურზე გადასვლა, ხოლო შემდეგ მინიმალურზე.

მ ა გ ა ლ ი თ ი 6: ფუნქციის მინიმიზაცია. (I ეტაპი)

$$
\begin{aligned}
f(x_1, x_2, x_3, x_4) = {} & \bar{x}_1 \bar{x}_2 x_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee \\
& \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee \\
& \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4 \vee \\
& \vee x_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 x_4
\end{aligned}
$$

ჩავწეროთ ფუნქციის წევრები ცხრილის პირველ სვეტში, მათზე გამოვიყენოთ შეწებების კანონი, ჯერ პირველი წევრი თანმიმდევრობით შევაწებოთ ყველა დანარჩენთან, შემდეგ მე-2 ყველა დანარჩენთან და ა. შ. შედეგები ჩავწეროთ ცხრილის მე-2 სვეტში, გადავნომროთ ისინი და ფრჩხილებში მივუთითოთ შეწებებული წევრების ნომრები, ხოლო პირველ სვეტში შეწებებული წევრები ავღნიშნოთ ვარსკლავებით. გავიმეოროთ ეს პროცედურა მე-2 სვეტის წევრებთან და ა. შ. ის იმპლიკანტები, რომლებიც არ შეწებდნენ, ჩავსვათ ჩარჩოებში, სწორედ ისინი არიან მარტივი იმპლიკანტები.

შევნიშნოთ, რომ ცხრილის მე-2 სვეტის იმპლიკანტების შეწებებისას შედარების დროს აღებულ იმპლიკანტასთან შემდგომი იმპლიკანტებიდან უნდა ავირჩიოთ მხოლოდ ისინი, რომლებიც შეიცავენ მოცემული იმპლიკანტის ინდექსებთან შესაბამის ასოებს.

თუ შეწებების შედეგად მიიღება ერთნაირი იმპლიკანტები, მაშინ ტოვებენ მხოლოდ ერთ მათგანს.

ამრიგად, I ეტაპი („პირველადი იმპლიკანტების პოვნა“) დასრულდა. ესაა ყველა ის იმპლიკანტა, რომელიც ჩასმულია ჩარჩოში (ცხრ.2.20).

ცხრილი 2.20

	$f(x_1, x_2, x_3, x_4)$-ის წევრები	I შეწებების შედეგები	II შეწებების შედეგები
1.	$\bar{x}_1, \bar{x}_2, x_3, x_4$ *	$\boxed{\bar{x}_1, x_3, x_4 \quad (1, 4)}$	$\boxed{x_2 \bar{x}_3 \quad (3, 9)}$
2.	$\bar{x}_1, x_2, \bar{x}_3, \bar{x}_4$ *	$\boxed{\bar{x}_2, x_3, x_4 \quad (1, 6)}$	$\cancel{x_2 \bar{x}_3 \quad (4, 6)}$
3.	$\bar{x}_1, x_2, \bar{x}_3, x_4$ *	$\bar{x}_1, x_2, \bar{x}_3$ * (2, 3)	
4.	$\bar{x}_1, x_2, x_3, x_4$ *	$x_2, \bar{x}_3, \bar{x}_4$ * (2, 7)	
5.	$x_1, \bar{x}_2, \bar{x}_3, x_4$ *	$\boxed{\bar{x}_1, x_2, x_4 \quad (3, 4)}$	
6.	$x_1, \bar{x}_2, x_3, x_4$ *	$x_2, \bar{x}_3, x_4$ * (3, 8)	
7.	$x_1, x_2, \bar{x}_3, \bar{x}_4$ *	$\boxed{x_1, \bar{x}_2, x_4 \quad (5, 6)}$	
8.	$x_1, x_2, \bar{x}_3, x_4$ *	$\boxed{x_1, \bar{x}_3, x_4 \quad (5, 8)}$	
9.		$x_1, x_2, \bar{x}_3$ * (7, 8)	

<u>*II ეტაპი. ნიშნულების დაყენება.*</u>

ვადგენთ ცხრილს, რომლის სტრიქონების რაოდენობა ტოლია ნაპოვნი მარტივი იმპლიკანტების რიცხვის, ხოლო სვეტების რაოდენობა - მოცემული ფუნქციის სდნფ-ს წევრების რიცხვის. პირველ სვეტში იწერებიან პირველადი იმპლიკანტები, ხოლო პირველ

სტრიქონში - ფუნქციის წევრები. თუ ფუნქციის წევრში შედის პირველადი იმპლიკანტა, მაშინ მათ გადაკვეთაზე დგება ნიშნული (V).

მე-3 რიგის პირველადი იმპლიკანტებისათვის მოსახერხებელია ნიშნულები დავაყენოთ პირველი სვეტის შეწებებული წევრების ნომრების მიხედვით, რომლებიც მიწერილებია გვერდით მდგარ იმპლიკანტებზე (ფრჩხილებში), ხოლო მე-2 რიგის იმპლიკანტებს - პირველი სვეტის წევრების ნომრების მიხედვით. სტრიქონში ნიშნულების რაოდენობა დამოკიდებულია იმპლიკანტიდან ამოღებულ რიცხვთა რაოდენობიდან. k ამოღებული ასოსათვის ნიშნულთა რიცხვი ტოლია 2^k- ს.

განვიხილოთ მე-2 ეტაპი მე-6 მაგალითის საფუძველზე. შევადგინოთ ცხრილი (ცხრ.2.21).

შევნიშნოთ, რომ $x_2\overline{x}_3$ წევრი მივიღეთ მე-2 სვეტის მე-3 და მე-9 წევრების შერწყმით, ხოლო ისინი - პირველი სვეტის მე-2, მე-3 და მე-7, მე-8 წევრებისაგან. ამრიგად, $x_2\overline{x}_3$ პირველადი იმპლიკანტა შეესაბამება მოცემული ფუნქციის 2, 3, 7, 8 წევრებს. ამით დასრულდა ნიშნულთა ცხრილის შედგენა (ცხრ.2.21).

<u>III ეტაპი. არსებითი იმპლიკანტების პოვნა.</u>

თუ ნიშნულთა შედგენილი ცხრილის რომელიმე სვეტში არის მხოლოდ ერთი ნიშნული, მაშინ პირველადი იმპლიკანტა, რომელიც დგას შესაბამის სტრიქონში, მ ნ ი შ ვ ნ ე ლ ო ვ ა ნ ი ა. ის არ შეიძლება ამოღებულ იქნას ფუნქციის მინიმალური ფორმიდან, ვინაიდან მის გარეშე არაა შესაძლებელი მოცემული ფუნქციის იმპლიკანტების მთელი სიმრავლის დაფარვის მიღება. ნიშნულთა ცხრილიდან ხდება იმ სტრიქონებისა და სვეტების ამოღება, რომელთა გადაკვეთაზეც დგას ეს ერთადერთი ნიშნული.

ცხრილი 2.21

	$\bar{x}_1, \bar{x}_2, x_3, x_4$	$\bar{x}_1, x_2, \bar{x}_3, \bar{x}_4$	$\bar{x}_1, x_2, \bar{x}_3, x_4$	$\bar{x}_1, x_2, x_3, x_4$	$x_1, \bar{x}_2, \bar{x}_3, x_4$	$x_1, \bar{x}_2, x_3, x_4$	$x_1, x_2, \bar{x}_3, \bar{x}_4$	$x_1, x_2, \bar{x}_3, x_4$
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
$\bar{x}_1, x_3, x_4$	V			V				
$\bar{x}_2, x_3, x_4$	V					V		
$\bar{x}_1, x_2, x_4$			V	V				
$x_1, \bar{x}_2, x_4$					V	V		
$x_1, \bar{x}_3, x_4$					V			V
$\boxed{x_2, \bar{x}_3}$		V	V				V	V

ნიშნულთა ცხრილში (ცხრ. 2.21) სვეტებს ერთადერთი ნიშნულით წარმოადგენენ სვეტები (2), (7). შესაბამისი იმპლიკანტა $x_2\bar{x}_3$ არსებითია. ნიშნულს წრეს ავლებენ, არსებით იმპლიკანტებს - ჩარჩოს, ხოლო ერთადერთ ნიშნულიან სვეტებს შლიან ცხრილიდან. შთანთქმის კანონის მიხედვით სვეტში ნიშნულთა ნაკლებმა რაოდენობამ შეიძლება ამოშალოს უფრო მეტი. ასე, სვეტები (2) და (7) შესაბამისად შედიან (3) და (8) სვეტებში, ამიტომ ნიშნულთა ცხრილიდან ვშლით (3) და (8) სვეტებს. ამით მე-3 ეტაპი დასრულებულია.

IV ეტაპი. ზედმეტი სვეტების ამოშლა.

თუ ცხრილში მე-3 ეტაპის შემდეგ ორი ერთნაირი სვეტია (რომლებშიც ნიშნულები ერთნაირ სტრიქონებში დგანან), მაშინ ერთ-ერთი მათგანი ამოიშლება, რადგანაც დარჩენილი სვეტის დაფარვა განახორციელებს ამოშლილი საწყისი იმპლიკანტის დაფარვას. განხილულ მაგალითში ასეთი სვეტები არა გვაქვს (იხ. ცხრ.2.21).

V ეტაპი. ზედმეტი სტრიქონების ამოშლა.

თუ ცხრილში მე-4 ეტეპის შემდეგ აღმოჩდნენ სტრიქონები, რომლებშიც არაა არც ერთი ნიშნული, მაშინ მათ შლიან, ე. ი. მათი შესაბამისი პირველადი იმპლიკანტები იშლებიან ფუნქციის მინიმალური ფორმებიდან, ვინაიდან ისინი არ იფარებიან დარჩენილი საწყისი იმპლიკანტებით. განსახილველ მაგალითში ასეთი სტრიქონები არაა (იხ. ცხრ.2.21).

VI ეტაპი. მინიმალური დაფარვის არჩევა.

ცხრილი 2.22

	$\overline{x}_1, \overline{x}_2, x_3, x_4$	$\overline{x}_1, x_2, x_3, x_4$	$x_1, \overline{x}_2, \overline{x}_3, x_4$	$x_1, \overline{x}_2, x_3, x_4$
	(1)	(4)	(5)	(6)
$\boxed{\overline{x}_1, x_3, x_4}$	V	V		
$\overline{x}_2, x_3, x_4$	V			V
$\overline{x}_1, x_2, x_4$		V		
$\boxed{x_1, \overline{x}_2, x_4}$			V	V
$x_1, \overline{x}_3, x_4$			V	

გამოვიკვლიოთ ყველა წინა ეტაპის ჩატარების შემდეგ მიღებული ცხრილი 2.22. ხდება ისეთი პირველადი იმპლიკანტების ერთობლიობის ამორჩევა, რომელსაც ექნებოდა ნიშნულები ყველა სვეტში. უპირატესობა ენიჭება დაფარვის ვარიანტს ასოთა მინიმალური რიცხვით დაფარვის განმახორციელებელ პირველად იმპლიკანტებში. მოცემულ მაგალითში ყველა დარჩენილი პირველადი იმპლიკანტა შეიცავს ასოთა ერთნაირ რაოდენობას. ავირჩიოთ დაფარვა $\overline{x}_1, x_3, x_4$ და $x_1, \overline{x}_2, x_4$ იმპლიკანტებიდან, ვინაიდან ისინი ფა-

რავენ ოთხივე დარჩენილ საწყის იმპლიკანტას. სწორედ ისინია არსებითი იმპლიკანტები.

ამრიგად, დარჩენილი არსებითი იმპლიკანტებით ვწერთ მოცემული ფუნქციის ჩიხურ ფორმას.

$$f(x_1,x_2,x_3,x_4)=x_2\overline{x}_3\vee\overline{x}_1x_3x_4\vee x_1\overline{x}_2x_4$$

ის არ უნდა იყოს ჭარბი. მოცემულ მაგალითში სწორედ ესაა ფუნქციის მინიმალური ფორმა.

შ ე ნ ი შ ვ ნ ა 1: ქუაინის მეთოდში არის ერთი არსებითი უხერხულობა - პირველადი იმპლიკანტების პოვნის ეტაპზე სრული წყვილური შედარების აუცილებლობა. ფუნქციის არგუმენტებისა და სდნფ-ში მისი განმსაზღვრელი წევრების რიცხვის ზრდასთან ერთად იზრდება ამ შედარებათა რიცხვიც. ეს ზრდა ხასიათდება ფაქტორიალური ფუნქციით. ამიტომ ქუაინის მეთოდის გამოყენება რთულდება.

შ ე ნ ი შ ვ ნ ა 2. ქუაინის მეთოდით ჩიხური ფორმები მიიღება. ისინი შეიძლება რამდენიმე იყოს. მათ შორისაა საჭირო მინიმალური ფორმების ძიება. ყველა შესაძლო ჩიხური ფორმა შეიძლება ნაპოვნი იყოს პეტრიკის მეთოდით.

2.4.6. ყველა შესაძლო ჩიხური ფორმის ძიების პეტრიკის მეთოდი.

არსებითი იმპლიკანტების ძიების გარეშე, ავღნიშნოთ ყველა იმპლიკანტა ლათინური ასოებით. საწყისი ფუნქცია შეიძლება ჩაიწეროს მარტივი იმპლიკანტების დიზიუნქციის სახით, რაც შეესაბამება შემოკლებულ ფორმას (რომელიც ერთადერთია). ეს ფორმა აგრეთვე ჩიხურია. ჩიხური ფორმების მთელი სიმრავლის მოსაძებნად ჩავწეროთ იგივური ლოგიკური ფორმულა:

$$F = \bigwedge_{j=1}^{m} \Big(\bigvee_{i=1}^{k} a_i \Big)_j \equiv 1,$$

სადაც a_i - j - ური სვეტის ნიშნულების შესაბამისი მარტივი იმპლიკანტებია, k - ნიშნულების რაოდენობაა j - ურ სვეტში, m - ნიშნულთა ცხრილში სვეტების რაოდენობაა. F ფორმულა იძლევა ფუნქციის სრულ სდნფ-ს, ე. ი. $F \equiv 1$.

თუ F ფორმულაში შეგვხდებათ a_r და $a_r \vee a_s$ წევრები, მაშინ $a_r \vee a_s$ წევრი შეიძლება არ ჩაწეროთ, რადგანაც $a_r(a_r \vee a_s) = a_r \vee a_r a_s = a_r(1 \vee a_s) = a_r \cdot 1 = a_r$ (აი, ქუაინის მეთოდის მე-3 ეტაპზე რატომ გადავაგდეთ დიდი სვეტები).

F გამოსახულება საჭიროა გავამარტივოთ (გავხსნათ ფრჩხილები და გამოვიყენოთ ალგებრის ლოგიკის კანონები). მივიღებთ წევრების დიზიუნქციას, რომელთაგან თითოეული იძლევა მარტივ იმპლიკანტთა სიმრავლეს, რომლებიც შედიან ჩიხურ ფორმაში. შევადგინოთ ცხრილი ნახ. 2.23-ზე მოყვანილი ფორმის მიხედვით.

ჩიხური ფორმები ასოთა უმცირესი რიცხვით წარმოადგენენ მინიმალურ ფორმებს, ხოლო ჩიხური ფორმები წევრების უმცირესი

რიცხვით - უმოკლესი ფორმებია. ხშირად მინიმალური ფორმა არ ემთხვევა უმოკლესს.

№№	ჩიხური ფორმები	ჩიხურ ფორმაში ასოთა საერთო რიცხვი	ჩიხურ ფორმაში წევრების რიცხვი
1.			
2.			
3.			
.			
.			

ნახ. 2.23

განვიხილოთ მე-5 მაგალითზე ეს მეთოდი (იხ. ქუაინის მეთოდში მე-2 ეტაპის ცხრილი). დავხაზოთ ის აქ. პირველადი იმპლიკანტები ავღნიშნოთ ლათინური ასოებით a, b, c, d, e, f, ხოლო სვეტები ციფრებით (1), (2), ... , (8) (ცხრილი 2.23).

შევადგინოთ F ფუნქცია.

$$F = (a \vee b) f (c \vee f)(a \vee c)(d \vee e)(b \vee d) f (e \vee f) \equiv 1$$

$a \vee b$ დიზიუნქციის ჩართვა ხდება პირველი სვეტისა და ა. შ. ნიშნულების რეალიზაციისათვის $f(c \vee f) = f, \quad f(e \vee f) = f$ შთანთქმის კანონით, ამიტომ მე-3 და მე-8 წევრები შეიძლება არ ჩავწეროთ. გავამარტივოთ F :

$$F = (a \vee b) f \overline{(a \vee c)(d \vee e)(b \vee d)} f = (a \vee bc)(d \vee be) f \equiv 1$$

გავხსნათ ფრჩხილები:

$$F = afd \vee bcdf \vee abef \vee bcef \equiv 1$$

ცხრილი 2.23

		$\bar{x}_1, \bar{x}_2, x_3, x_4$	$\bar{x}_1, x_2, \bar{x}_3, \bar{x}_4$	$\bar{x}_1, x_2, \bar{x}_3, x_4$	$\bar{x}_1, x_2, x_3, x_4$	$x_1, \bar{x}_2, \bar{x}_3, x_4$	$x_1, \bar{x}_2, x_3, x_4$	$x_1, x_2, \bar{x}_3, \bar{x}_4$	$x_1, x_2, \bar{x}_3, x_4$
		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
$\bar{x}_1, x_3, x_4$	a	V			V				
$\bar{x}_2, x_3, x_4$	b	V					V		
$\bar{x}_1, x_2, x_4$	c			V	V				
$x_1, \bar{x}_2, x_4$	d					V	V		
$x_1, \bar{x}_3, x_4$	e					V			V
$x_2, \bar{x}_3$	f		V	V				V	V

F - ის თითოეული წევრი იძლევა მოცემული ფუნქციის ჩიხურ ფორმას. შევადგინოთ ცხრილი:

ცხრილი 2.24

№№	ჩიხური ფორმები	ჩიხურ ფორმაში ასოთა საერთო რიცხვი	ჩიხურ ფორმაში წევრების რაოდენობა
1.	adf	3+3+2=8	3
2.	$bcdf$	3+3+3+2=11	4
3.	$abef$	3+3+3+2=11	4
4.	$bcef$	3+3+3+2=11	4

ცხრილიდან გამომდინარეობს, რომ პირველი გადაწყვეტა მინიმალური ფორმაა (შეადარეთ შედეგი), ის აგრეთვე იძლევა უმოკლეს ფორმასაც. კიდევ ერთხელ შევნიშნოთ, რომ უმოკლესი და მინიმალური ფორმები შეიძლება არ ემთხვეოდნენ ერთმანეთს.

ამრიგად, $f(x_1, x_2, x_3, x_4) = \bar{x}_1 x_3 x_4 \vee x_1 \bar{x}_2 x_4 \vee x_2 \bar{x}_3$ მოცემული ფუნქციის მინიმალური ფორმაა.

შ ე ნ ი შ ვ ნ ა 3: დაფარვის ცხრილი შეიძლება არ შეიცავდეს არსებით იმპლიკანტებს. განვმარტოთ, თუ როგორ უნდა მოვიქცეთ ამ შემთხვევაში. ვთქვათ, ნიშნულთა ცხრილს შემდეგი სახე აქვს (იხ. ცხრ. 2.25).

წავშალოთ მე-2 და მე-7 სვეტები, ვინაიდან მე-3 და მე-5 სვეტები მათი ნაწილებია, ხოლო დარჩენილი სვეტებიდან ირჩევენ სვეტს ნიშნულთა უმცირესი რაოდენობით. აქ ყველა სვეტში მათი რაოდენობა ორ-ორია, ამიტომ ავიღოთ პირველი სვეტი, ფსევდოარსებით იმპლიკანტად ჯერ ავიღოთ a , ხოლო შემდეგ d.

ცხრილი 2.25

	1	2	3	4	5	6	7
a	v	v	v				v
b		v	v	v		v	
c		v		v	v		v
d	v				v	v	v

განვიხილოთ ორი კერძო შემთხვევა:

	1	3	4	5	6
a	v	v			
b		v	v		v
c			v	v	
d	v			v	v

1)

	1	3	4	5	6
a	v	v			
b		v	v		v
c			v	v	
d	v			v	v

2)

	1	3	4	5	6
a	v	v			
b		v	v		v
c			v	v	
d	v			v	v

გამოვრიცხოთ დიდი სვეტები, რომლებიც შეიცავენ არჩეულ ფსევდოსვეტებს. თითოეული ცხრილისათვის ჩავწეროთ ჩიხურ ფორმათა სიმრავლე. ეს შეიძლება შევასრულოთ პეტრიკის მეთოდით, მაგრამ აქ შეგვიძლია ყველა შესაძლო ვარიანტი გავარჩიოთ ცხრილით:

1): $a \vee b \vee c$ (1)

$a \vee b \vee d$ (2)

$a \vee c \vee d$ (3)

2): $a \vee c \vee d$ (4)

$b \vee d$ (5)

ერთობლივად განვიხილოთ ამოხსნათა სიმრავლე. ამოხსნა (2) შედის (5)-ში, (3) ემთხვევა (4)-ს, ხოლო (1)-ს არა აქვს შესატყვისი მე-2 ცხრილში; ყველაზე მარტივი ჩიხური ფორმაა (5). ამრიგად, ქვეცხრილებად დაყოფა ამარტივებს ჩიხური ფორმების პოვნას.

2.4.7. მაკ-კლასკის მეთოდი.

ეს მეთოდი ქუაინის მეთოდის მოდერნიზაციას წარმოადგენს (მისი პირველი ეტაპის). მაკ-კლასკიმ შემოგვთავაზა სდნფ-ში მოცემული ფუნქციის საწყისი იმპლიკანტების ჩაწერა მათი ორობითი კოდის სახით (თითოეულ წევრს ეთითება ცნობილი წესის შესაბამისად მისი საკუთარი წვერო). ასე ჩაწერილი იმპლიკანტების მთელი სიმრავლე იყოფა ჯგუფებად მათ კოდებში არსებული ერთიანების რიცხვის მიხედვით. ამასთან i -ურ ჯგუფში შევლენ კოდები, რომლებსაც თავიანთ ჩანაწერში i ერთეული აქვთ. იმპლიკანტების წყვილური შედარება საკმარისია ჩატარდეს მხოლოდ მეზობელ ჯგუფებს შორის, ვინაიდან მხოლოდ ეს ჯგუფები განსხვავდებიან მათში შემავალი წევრების კოდებში ერთი ნიშნით. მეზობელი ჯგუფების წევრების კოდების შედარებისას, იქმნებიან დაბალი რანგის წევრები. წაშლილი ასოს ადგილას მათში წერენ „ტირეს“. შემდეგ დაბალი რანგის წევრების მთელ ერთობლიობას ისევ ყოფენ ჯგუფებად ნიშან „ტირეს“ ადგილმდებარეობის მიხედვით. ისევ ადარებენ მეზობელი ჯგუფების წევრებს, მაგრამ ახლა უკვე ჯგუფების შიგნით, ქმნიან რა დაბალი რანგის წევრებს იგივე წესით და ა. შ.

შემდეც ყველაფერი სრულდება ქუაინის მეთოდით, მაგრამ იმპლიკანტების კოდურ მნიშვნელობებში.

$$f(x_1,x_2,x_3,x_4) = \bar{x}_1\bar{x}_2x_3x_4 \vee \bar{x}_1x_2\bar{x}_3\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_3x_4 \vee \bar{x}_1x_2x_3x_4 \vee$$
$$\vee\, x_1\bar{x}_2\bar{x}_3x_4 \vee x_1\bar{x}_2x_3x_4 \vee x_1x_2\bar{x}_3\bar{x}_4 \vee x_1x_2\bar{x}_3x_4$$

შევცვალოთ საწყისი იმპლიკანტები მათი კოდებით ორობით ცვლადებში: 0011, 0100, 0101, 0111, 1001, 1011, 1100, 1101.

საწყისი იმპლიკანტების კოდები დავყოთ ჯგუფებად, მოვათავსოთ ისინი ცხრილში. შემდეგ გამოვიყენოთ შეწებების კანონი მეზობელი ჯგუფების წევრების მიმართ, პირველი ჯგუფის თითოეული წევრის მე-2 ჯგუფის ყველა წევრთან შედარებით და ა. შ.

ყოველივე ამის შესრულება პირდაპირ ცხრილში შეიძლება:

ცხრილი 2.26

მოცემული ფუნქცია			I შეწებების შედეგები			I შეწებების შედეგები	
კოდები	ჯგუფები		კოდები	ჯგუფები		კოდები	ჯგუფები
0011	0-ნი	-	0-11	1	-011	-10-	
0100	1	0100 *	-011		-100 *	~~-10-~~	
0101	2	0011 *	010-		-101 *		
0111		0101 *	0100	2	0-11		
1001		1001 *	01-1		1-01		
1011		1100 *	-101	3	01-1		
1100	3	0111 *	10-1		10-1		
1101		1011 *	1-01	4	010- *		
		1101 *	110-		110- *		

შემდეგ იგება ნიშნულთა ცხრილი (ცხრ. 2.27), მაგრამ მათში იწერებიან საწყისი და პირველადი იმპლიკანტები ორობით კოდებ-ში. მიაქციეთ ყურადღება იმას, რომ პირველადი იმპლიკანტები ჩა-წერილია სხვა რიგით (ჯგუფების მიხედვით), ამიტომ ნიშნულთა ცხრილი გამოიყურება სხვაგვარად ვიდრე მე-5 მაგალითში:

ცხრილი 2.27

	0100	0011	0101	1001	1100	0111	1011	1101
-011		v					v	
0-11		v				v		
1-01				v				v
01-1			v			v		
10-1				v			v	
-10-	v		v		v			v

ნიშნულთა ცხრილის დამუშავება სრულდება ქუაინის მეთო-დით.

$$f = (-10-) \vee (0-11) \vee (10-1),$$

$$f(x_1, x_2, x_3, x_4) = x_2 \bar{x}_3 \vee \bar{x}_1 x_3 x_4 \vee x_1 \bar{x}_2 x_4$$

2.4.8. ინდექსების კომბინაციის მეთოდი.

ლოგიკური ფუნქციების მინიმიზაციის არა ნაკლებ ეფექტურ ხერხს წარმოადგენს ი ნ დ ე ქ ს ე ბ ი ს კ ო მ ბ ი ნ ა ც ი ი ს მეთოდი. მისი გადმოცემისათვის შევადგინოთ ცხრილი 2.28, რომლის სვეტებში ჩაწერილია ინდექსთა შესაძლო შეერთებები. ბოლო სვეტში ამოწერილია ფუნქციის მნიშვნელობები. ცხრილის ანალიზი იწყება მარცხნიდან სვეტების მიხედვით. i, j -კოდის გამორიცხვის პრინციპი შემდეგში მდგომარეობს. i -იური სვეტის, მაგალითად ინდექსთა კომბინაციით 23, და j - იური სტრიქონის, მაგალითად მე-3 -ის, გადაკვეთაზე მოთავსებულია კოდი 10, რაც შეესაბამება $x_2\bar{x}_3$ იმპლიკანტას. გამომდინარე, ამ სვეტში ყველგან, სადაც გვხვდება კოდი 10, ანუ 2, 3, 10 და 11 სტრიქონებში, ხდება ამ კოდების გამორიცხვა, ვინაიდან მე-3 სტრიქონში ფუნქციის მნიშვნელობა ნულის ტოლია.

ცხრილი 2.28

n	x_1	x_2	x_3	x_4	12	13	14	23	24	34	123	124	134	234	1234	F
0	0	0	0	0	00	00	00	00	00	00	000	000	000	000	0000	0
1	1	0	0	0	10	10	10	00	00	00	100	100	100	000	1000	1
2	0	1	0	0	01	00	00	10	10	00	010	010	000	100	0100	0
3	1	1	0	0	11	10	10	10	10	00	110	110	100	100	1100	0
4	0	0	1	0	00	01	00	01	00	10	001	000	010	010	0010	0
5	1	0	1	0	10	11	10	01	00	10	101	100	110	010	1010	0
6	0	1	1	0	01	01	00	11	10	10	011	010	010	110	0110	1
7	1	1	1	0	11	11	10	11	10	10	111	110	110	110	1110	1
8	0	0	0	1	00	00	01	00	01	01	000	001	001	001	0001	0
9	1	0	0	1	10	10	11	00	01	01	100	101	101	001	1001	1
10	0	1	0	1	01	00	01	10	11	01	010	011	001	101	0101	0
11	1	1	0	1	11	10	11	10	11	01	110	111	101	101	1101	0
12	0	0	1	1	00	01	01	01	01	11	001	001	011	011	0011	1
13	1	0	1	1	10	11	11	01	01	11	101	101	111	011	1011	1
14	0	1	1	1	01	01	01	11	11	11	011	011	011	111	0111	1
15	1	1	1	1	11	11	11	11	11	11	111	111	111	111	1111	1

ამრიგად, ყველა კოდი იმ სტრიქონებში, რომლებზეც ფუნქციას ნულოვანი მნიშვნელობა აქვს, ავტომატურად გამოირიცხება. თუ ეს კოდები ხვდებიან იმ სტრიქონებში, რომლებზეც ფუნქციას ერთეულოვანი მნიშვნელობა აქვს, მაშინ არც მათი გათვალისწინება არ

ხდება. რჩება მხოლოდ ის სტრიქონები, რომლებიც მოთავსებულებია სტრიქონებზე ფუნქციის ერთეულოვანი მნიშვნელობებით.

n	x_1	x_2	x_3	x_4	12	13	14	23	24	34	123	124	134	234	1234	F
1	1	0	0	0							100				1000	1
6	0	1	1	0				11			011			110	0110	1
7	1	1	1	0				11			111			110	1110	1
9	1	0	0	1							100	101			1001	1
12	0	0	1	1						11			011	011	0011	1
13	1	0	1	1						11		101	111	011	1011	1
14	0	1	1	1				11		11	011		011	111	0111	1
15	1	1	1	1				11		11	111		111	111	1111	1

შემდეგ ხელმძღვანელობენ შემდეგი წესით. იმისათვის, რომ ფუნქციამ მიიღოს ერთის ტოლი მნიშვნელობა, საკმარისია სტრიქონში რომელიმე ერთმა იმპლიკანტამ მიიღოს ერთეულოვანი მნიშვნელობა. უპირველესად ვტოვებთ მინიმალურ იმპლიკანტას x_2x_3, რომელიც ფარავს ერთიანებს მე-6, მე-7, მე-14 და მე-15 სტრიქონებში. შემდეგ, ბუნებრივია, გადავდივართ სვეტზე ინდექსების კომბინაციით 34. აქ ვტოვებთ კოდს 11, რაც ფარავს მე-12 სტრიქონის ერთიანს და შეესაბამება x_3x_4 იმპლიკანტას. იგივე იმპლიკანტაა პასუხისმგებელი მე-13 სტრიქონზე, რომელიც ერთიანით მთავრდება. დაგვრჩა პირველი და მე-9 სტრიქონების განხილვა. მათთვის საერთოა $x_1\bar{x}_2\bar{x}_3$ იმპლიკანტა. ამრიგად, სამი იმპლიკანტით ჩვენ გადავ-

n	x_1	x_2	x_3	x_4	12	13	14	23	24	34	123	124	134	234	1234	F
1	1	0	0	0							100				1000	1
6	0	1	1	0				11			011			110	0110	1
7	1	1	1	0				11			111			110	1110	1
9	1	0	0	1							100	101			1001	1
12	0	0	1	1						11			011	011	0011	1
13	1	0	1	1						11		101	111	011	1011	1
14	0	1	1	1				11		11	011		011	111	0111	1
15	1	1	1	1				11		11	111		111	111	1111	1

ფარეთ ფუნქციის ყველა ერთეულოვანი მნიშვნელობა და მივიღეთ მდნფ: $x_2x_3 \vee x_3x_4 \vee x_1\bar{x}_2\bar{x}_3$.

2.4.9. ვეიჩის დიაგრამების მეთოდი.

ეს მეთოდი ყველაზე მოხერხებულია საინჟინრო ამოცანათა გადასაჭრელად, ვინაიდან ამარტივებს შესაწებებელი წევრების ძიებას, მაგრამ ის შეზღუდულია მოცემული ფუნქციის არგუმენტების რიცხვით. პრაქტიკულად ვეიჩის ბარათების მეთოდით მინიმიზაცია ხორციელდება ფუნქციებისათვის, რომელთა არგუმენტების რიცხვი არ აღემატება რვას ($n \leq 8$).

ვეიჩის ბარათი წარმოადგენს ფუნქციის სპეციალურ ცხრილს.

განვიხილოთ ორი ცვლადის ფუნქციის ბარათი.

x_1	x_2	$f(x_1, x_2)$
0	0	1
0	1	0
1	0	1
1	1	1

	$x_2 = 0$	$x_2 = 1$
$x_1 = 0$	1	0
$x_1 = 1$	1	1

$x_1 \setminus x_2$	0	1
0	1	0
1	1	1

შესაძლებელია ბარათის გამარტივება, თუ არგუმენტებისათვის შემოვიტანთ სიმბოლურ აღნიშვნას ხაზით და მას დავაყენებთ იქ, სადაც ისინი ერთის ტოლია.

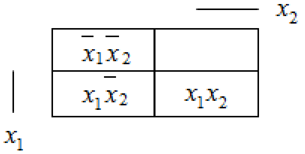
		x_2
	1	0
x_1	1	1

ბარათში შეაქვთ ცვლადების ნაკრების შესაბამისი ფუნქციის მნიშვნელობები.

ცხრილის უჯრედების განლაგებით ადვილად ხდება შესაწებებელი წევ-

რების განსაზღვრა. მეზობელი უჯრედები შეესაბამებიან ერთი ნიშ-ნით განსხვავებულ წევრებს და შეიძლება მათი შეწებება, თუ მათში ფუნქციის მნიშვნელობა ერთის ტოლია.

ფუნქციის სდნფ-ს შესაბამის უჯრედებში ჩაწერის შემდეგ ად-ვილია მისი წევრების დანახვა. მაგალითად ზემოთ მოყვანილ ბარათში.

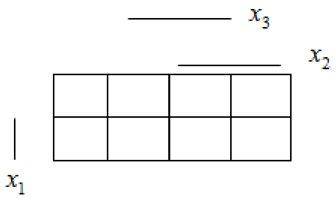


$$f(x_1,x_2)=\bar{x}_1\bar{x}_2 \vee x_1\bar{x}_2 \vee x_1x_2 = \bar{x}_2 \vee x_1$$

სვეტის წევრები წებდებიან იმ ცვლადით, რომელსაც შეესაბა-მება მთელი სვეტი, ხოლო სტრიქონები - მთელი სტრიქონი.

განვიხილოთ ვეიჩის ბარათი სამი ცვლადის ფუნქციისათვის. ბარათი ავაგოთ არგუმენტების სიმეტრიული განლაგებით, ერთი მათგანი მოვათავსოთ ერთი მხრიდან, ორი სხვა - მეორე მხრიდან.

გავყოთ ბარათი ორი ღერძით, რომელთა სიმეტრიულად გან-ვალაგებთ არგუმენტებს (იხ. ბარათი).



ბარათის თითოეული უჯრედი შეესაბამება ფუნქციის (რომე-ლიც 3 ნიშანს შეიცავს) სდნფ-ს წევრებს.

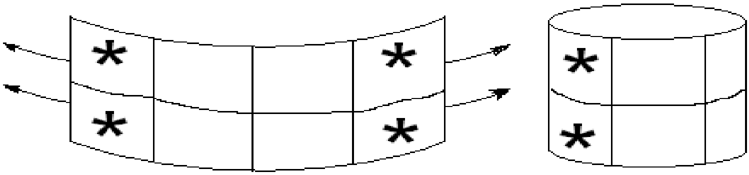
x_3

x_2

	1 $\bar{x}_1\bar{x}_2\bar{x}_3$	2 $\bar{x}_1\bar{x}_2x_3$	3 $\bar{x}_1x_2x_3$	4 $\bar{x}_1x_2\bar{x}_3$
x_1	5 $x_1\bar{x}_2\bar{x}_3$	6 $x_1\bar{x}_2x_3$	7 $x_1x_2x_3$	8 $x_1x_2\bar{x}_3$

ყურადღება მიაქციეთ იმას, რომ მეზობელი უჯრედების თითოეული წყვილი შეიძლება შეწებდეს, შეიძლება შეწებდეს ნებისმიერი ოთხი მეზობელი უჯრედი და ყველა რვა. ასე შეიძლება შევაწებოთ პირველი და მე-5, პირველი და მე-2 და ა. შ. უჯრედები, აგრეთვე 2, 3, 6, 7; 1, 5, 4, 8 და ა. შ.

თუ ბარათს ვერტიკალური ცილინდრის სახით შეკრულად წარმოვიდგენთ, მაშინ კიდურა უჯრედები აღმოჩნდებიან მეზობლად, მათი შეწებებაც შეიძლება.



განვიხილოთ რამდენიმე მაგალითი.

x_3

x_2

	1	1	1	
x_1			1	

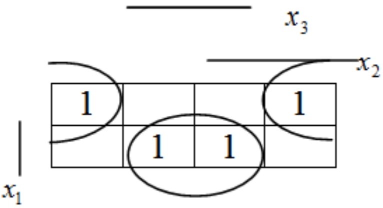
$$\bar{x}_1\bar{x}_2 \vee x_2x_3$$

x_3

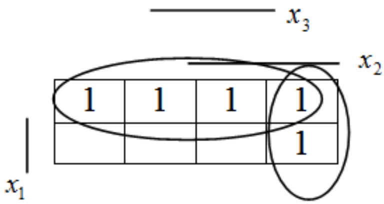
x_2

	1	1		
x_1		1		

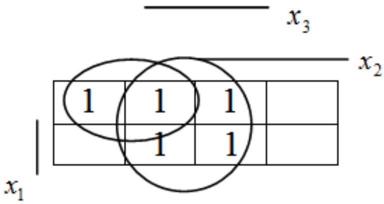
$$\bar{x}_1\bar{x}_2 \vee \bar{x}_2x_3$$



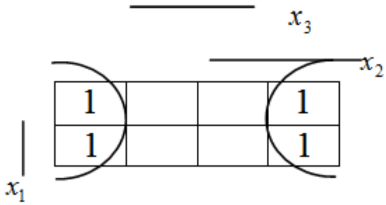
$\bar{x}_1 \bar{x}_3 \vee x_1 x_3$



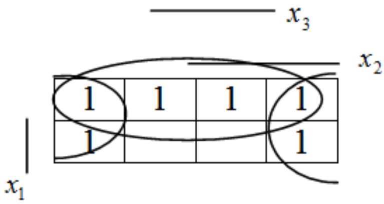
$\bar{x}_1 \vee x_2 \bar{x}_3$



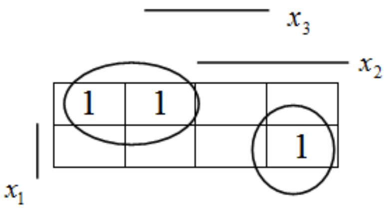
$\bar{x}_1 \bar{x}_2 \vee x_3$



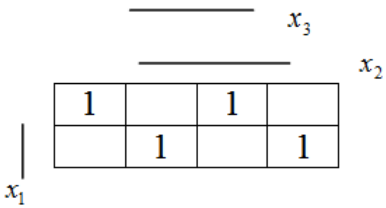
$\bar{x}_3$



$x_1 \vee \bar{x}_3$

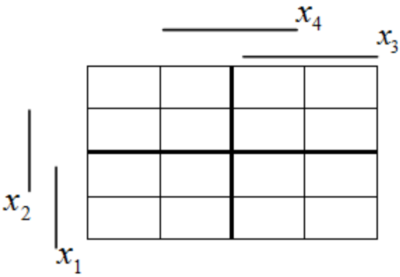


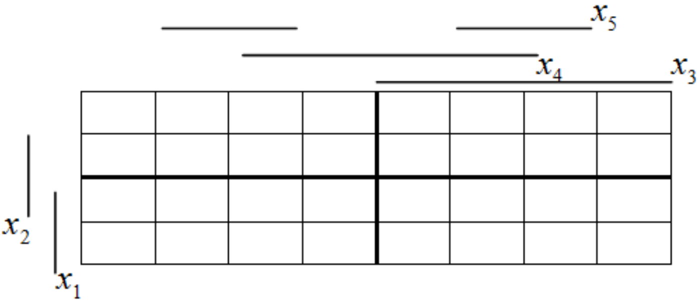
$\bar{x}_1 \bar{x}_2 \vee x_1 x_2 \bar{x}_3$

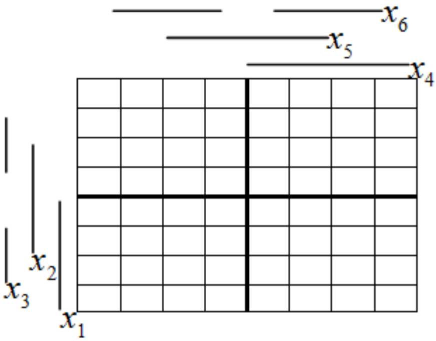


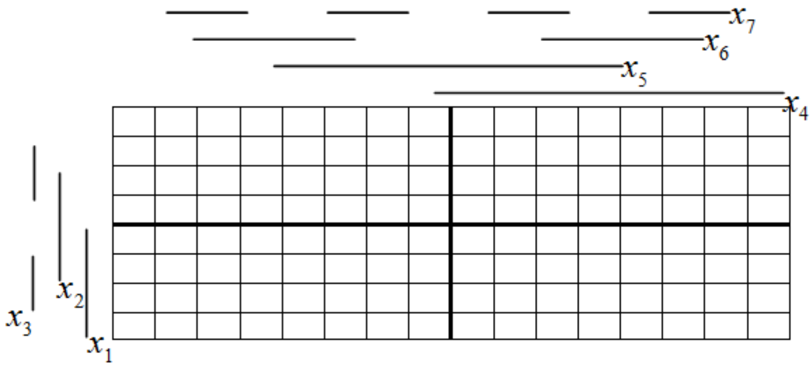
შეუძლებელია შეწებება

მოვიყვანოთ ვეიჩის ბარათების მაგალითები $n = 4, 5, 6, 7$ არგუმენტთა რიცხვისთვის.



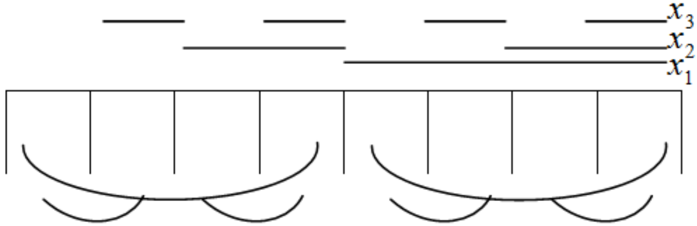




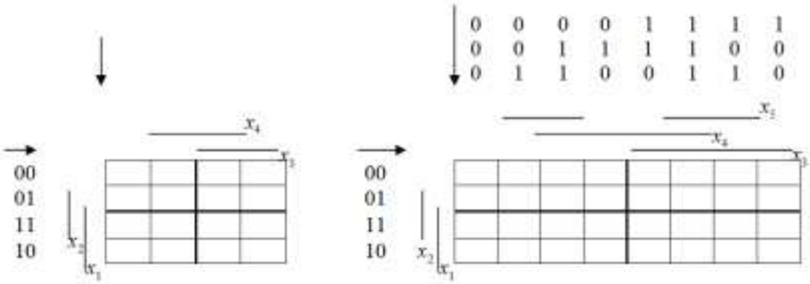


ვეიჩის ბარათში არგუმენტების განლაგება შეიძლება ნებისმიერი იყოს. უმჯობესია მათი ნახევარი მოვათავსოთ ერთ მხარეს, ხოლო მეორე ნახევარი - მეორე მხარეს. უფრო მოსახერხებელია ცენტრალური ღერძების მიმართ არგუმენტების სიმეტრიული განლაგების ვარიანტი.

შესაძლოა მეორე ხერხი, რომლიც დროსაც არგუმენტთა მნიშვნელობები ლაგდებიან მარჯვნივ ნახევარი ბარათის ფარგლებში, მეოთხედი ბარათის ფარგლებში და ა. შ.

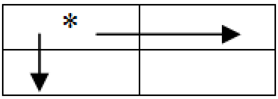


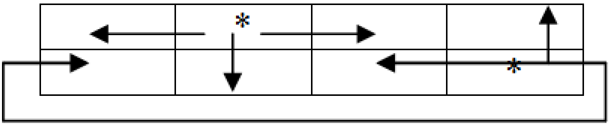
ვეიჩის ბარათში საჭირო უჯრედის ძიების გასამარტივებლად მიზანშეწონილია დამატებით დაინომროს სტრიქონები და სვეტები.

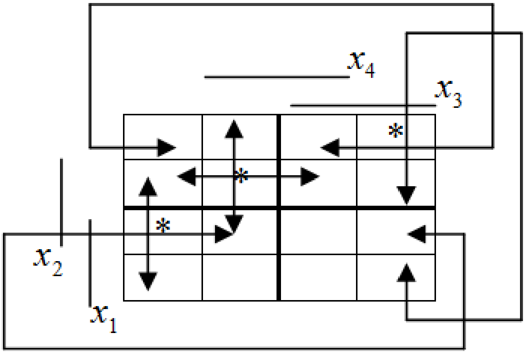


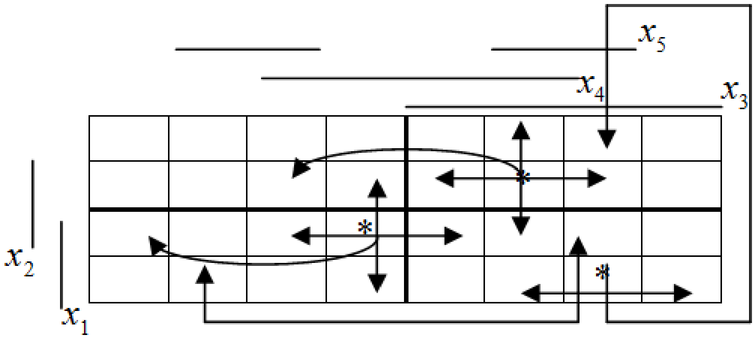
ბარათების დამუშავება.

არგუმენტების სიმეტრიული განლაგების მქონე ბარათების აგების ხერხისაგან დამოკიდებულებით ცხადია, რომ n არგუმენტის ფუნქციის თითოეული უჯრედს აქვს n მეზობელი უჯრედი, ანუ ის უჯრედები, რომლებთანაც შესაძლებელია შეწებების შესრულება.







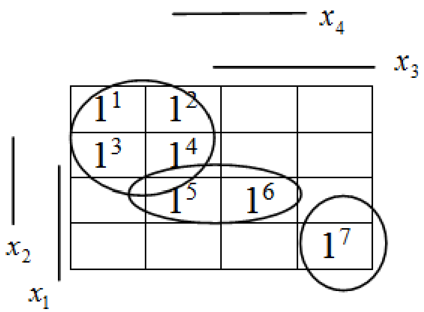


ღერძების მიმართ სიმეტრიულად განლაგებული უჯრედები მეზობლებია, ანუ შესაძლებელია მათი შეწებება. სიმეტრიის წესი არ ვრცელდება სხვა მეთოდებზე.

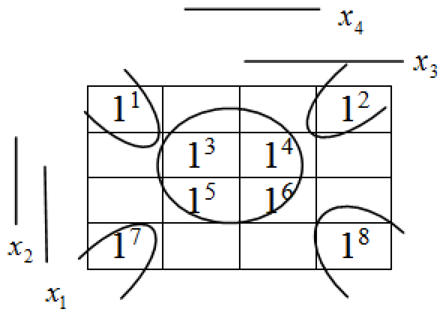
ბარათში იწერებიან ფუნქციის მხოლოდ ერთის ტოლი მნიშვნელობები, ნულები არ იწერებიან. შეიძლება 2^k $(k = 1,2,\dots,n)$ უჯრედის შეწებება, ანუ სრული სტრიქონების, სრული სვეტების, რომლებიც ბარათში გაივლიან, ნახევარ ბარათის, მეოთხედი ბარათის და ა. შ.

2^k უჯრედის შეწებების დროს ამოვარდება k ცვლადი, ანუ დარჩება $(n-k)$ ცვლადი.

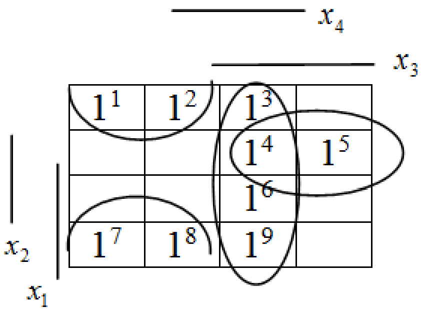
რთული არაა შევნიშნოთ, რომ მარტივი იმპლიკანტები შეესაბამებიან ზარათის მაქსიმალურ ზონებს, ანუ ისეთებს, რომლებიც არ შეიძლება გავზარდოთ.



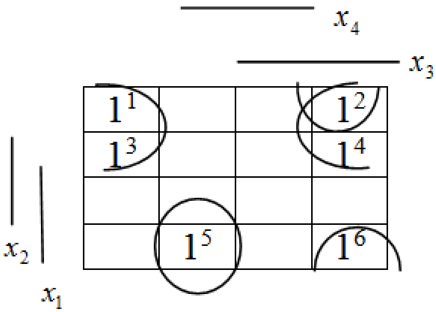
$$\bar{x}_1\bar{x}_3 \vee x_1x_2x_4 \vee x_1\bar{x}_2x_3\bar{x}_4$$



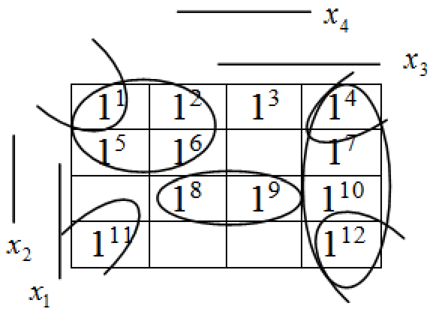
$$x_2x_4 \vee \bar{x}_2\bar{x}_4$$



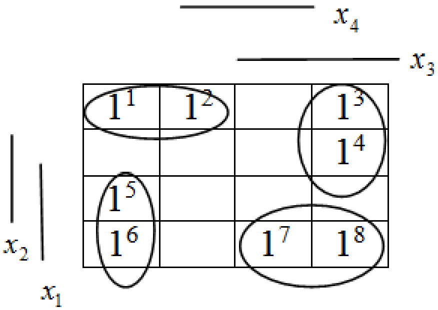
$$\bar{x}_2 \bar{x}_3 \vee x_3 x_4 \vee \bar{x}_1 x_2 x_3$$



$$\bar{x}_1 \bar{x}_4 \vee \bar{x}_2 x_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4$$

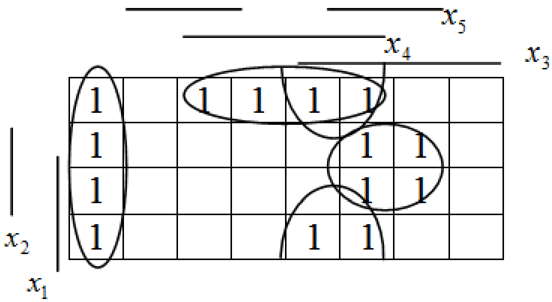


$$\bar{x}_1 \bar{x}_2 \vee \bar{x}_1 \bar{x}_3 \vee x_3 \bar{x}_4 \vee \bar{x}_2 \bar{x}_4 \vee x_1 x_2 x_4$$

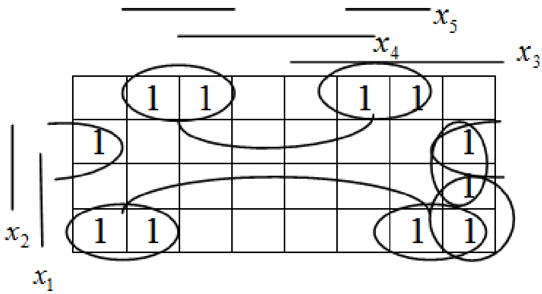


$$\bar{x}_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1 x_3 \bar{x}_4 \vee x_1\bar{x}_2 x_3 \vee x_1\bar{x}_3\bar{x}_4$$

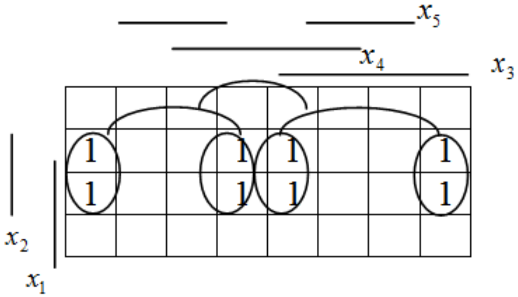
ყურადღება მიაქციეთ იმას, რომ ბოლო ბარათში აზრი არა აქვს 1, 3, 6, 8 უჯრედების გაერთიანებას, ვინაიდან მოგვიხდება დარჩენილი უჯრედების მათთანვე გაერთიანება.



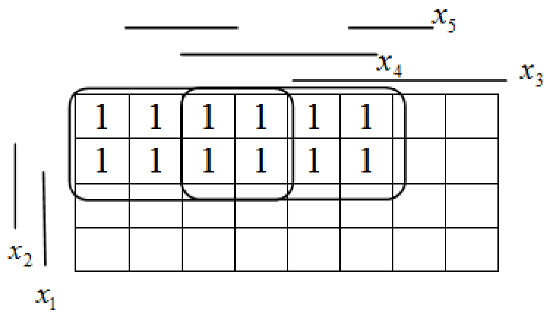
$$\bar{x}_3\bar{x}_4\bar{x}_5 \vee \bar{x}_1\bar{x}_2 x_4 \vee x_2 x_3 x_5 \vee \bar{x}_2 x_3 x_4$$



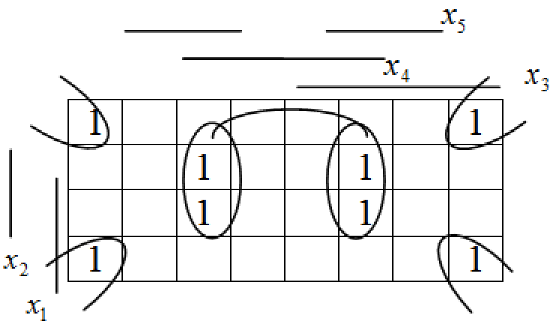
$$\bar{x}_1\bar{x}_2x_5 \vee x_1\bar{x}_2\bar{x}_4 \vee \bar{x}_1x_2\bar{x}_4\bar{x}_5 \vee \frac{x_2x_3\bar{x}_4\bar{x}_5}{x_1x_3\bar{x}_4\bar{x}_5}$$



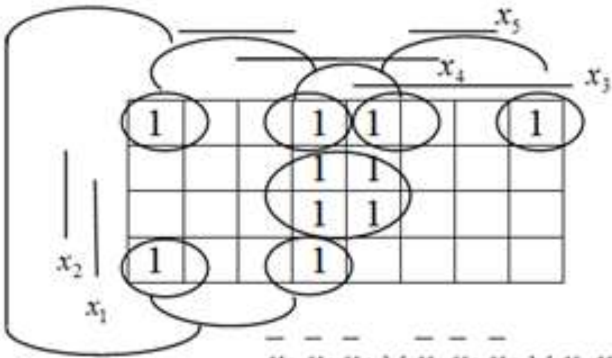
$$x_2\bar{x}_5$$



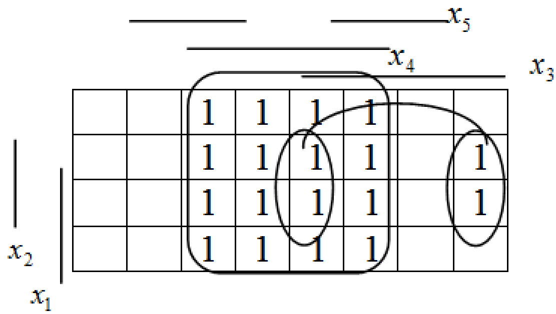
$$\bar{x}_1x_4 \vee \bar{x}_1\bar{x}_3$$



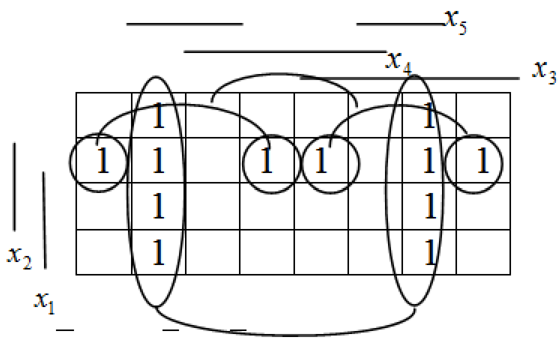
$$x_2 x_4 x_5 \vee \bar{x}_2 \bar{x}_4 \bar{x}_5$$



$$\bar{x}_2 \bar{x}_3 \bar{x}_5 \vee \bar{x}_1 \bar{x}_2 \bar{x}_5 \vee x_2 x_4 \bar{x}_5$$



$$x_4 \vee x_2 x_3 \bar{x}_5$$



$$\bar{x}_4 x_5 \vee \bar{x}_1 x_2 \bar{x}_5$$

ნაწილი III

ციფრული მოწყობილობის ლოგიკური ელემენტები.

3.1. კომბინაციური ლოგიკის ელემენტები.

თანამედროვე კომპიუტერების ელექტრონული ბლოკების მთელი სირთულის მიუხედავად მათ მიერ შესრულებული მოქმედებები ხორციელდება ტიპიური ლოგიკური კვანძების შედარებით მცირე რიცხვის კომბინაციათა მეშვეობით. მათგან ძირითადები შემდეგია:

- რეგისტრები;

- კოდების კომბინირებული გარდამქმნელები (კოდერი, დეკოდერი, მულტიპლექსორი და სხვა);

- მთვლელები (რგოლური, სინქრონული, ასინქრონული და სხვა);

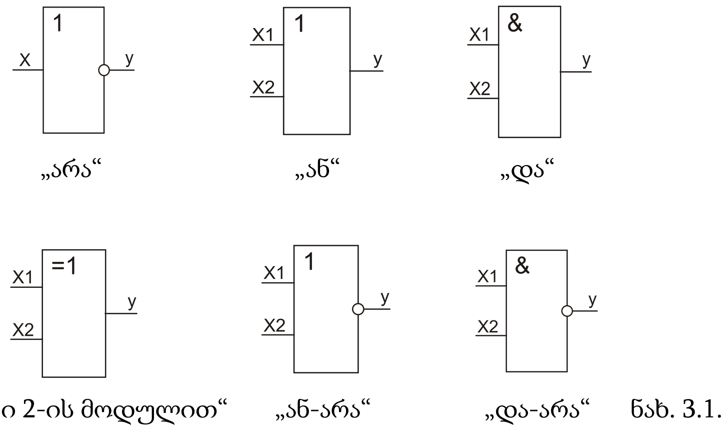
- არითმეტიკულ-ლოგიკური კვანძები (ამჯამავი, შედარების კვანძი და სხვა).

იმ კვანძებისაგან აგებენ ძალიან მაღალი დონის ინტეგრაციის იტეგრალურ მიკროსქემებს: მიკროპროცესორებს, ოპერაციული მეხსიერებების მოდულებს, გარე მოწყობილობათა კონტროლიორებს და ა. შ.

თავად აღნიშნულ კვანძებს ძირითადი ბაზური ლოგიკის ელემენტებით - როგორც მარტივების, რომლებიც ახორციელებენ „ან“, „და“, „არა“ , „და- არა“, „ ან-არა“ და მათ მსგავს ლოგიკურ ფუნქციებს (კომბინაციური ლოგიკის ელემენტებს, რომელთათვისაც გამოსასვლელზე ფუნქციის მნიშვნელობა ცალსახადაა განსაზღვრული დროის მოცემულ მომენტში შესასვლელი ცვლადების კომბინაცი-

ით), აგრეთვე უფრო რთულების, ისეთების როგორიცაა ტრიგერები (მიმდევრობითი ლოგიკის ელემენტები, რომელთათვისაც ფუნქციის მნიშვნელობა დამოკიდებულია არა მხოლოდ შესასვლელზე ცვლადების მიმდინარე მნიშვნელობებისაგან, არამედ მათი წინა მნიშვნელობებითაც).

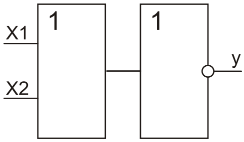
კომბინაციური ლოგიკის ძირითადი ელემენტების პირობითი აღნიშვნები მოყვანილია ნახ. 3.1-ზე, ცვლადების შესაბამისი მნიშვნელობები („ჭეშმარიტების ცხრილები“) - ცხრილ 3.1-ში.



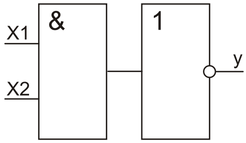
ნახ. 3.1.

როგორც წესი, ძირითად ლოგიკურ ელემენტებს აქვს ერთი გამოსასვლელი (y) და რამდენიმე შესასვლელი, რომელთა რაოდენობაც ტოლია არგუმენტების რიცხვის $(x_1, x_2, x_3, \dots, x_n)$. ელექტრულ სქემებზე ლოგიკური ელემენტები გამოისახებიან მართკუთხედებით გამოსასვლელებით შესასვლელი (მარცხნიდან) და გამოსასვლელი (მარჯვნიდან) ცვლადებისათვის. მართკუთხედის მარცხენა ზედა კუთხეში მოთავსებულია ელემენტის ფუნქციური დანიშნულების მაჩვენებელი პირობითი აღნიშვნა (სიმბოლო). ლოგიკური ელემენტის სქემის გამოსასვლელზე პატარა წრე ნიშნავს, რომ ელემენტი მართკუთხედის შიგნით მითითებული ოპერაციის შედეგის ლოგიკურ უარყოფას ახდენს.

პირსის ელემენტი („ან-არა“) შეგვიძლია წარმოვიდგინოთ „ან“ ელემენტისა და „არა“ ელემენტის მიმდევრობითი შეერთებით (ნახ. 3.2), ხოლო შეფერის ელემენტი („და -არა“) – „და“ ელემენტისა და „არა“ ელემენტის მიმდევრობითი შეერთებით (ნახ. 3.3).



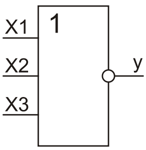
ნახ. 3.2



ნახ. 3.3

ლოგიკური ელემენტები, რომლებიც ახორციელებენ კონიუნქციისა და დიზიუნქციის ოპერაციებს, პირსისა და შეფერის ფუნქციებს, ზოგად შემთხვევაში შეიძლება n შესასვლელიანები იყვნენ. ასე მაგალითად, სამშესასლელიან ლოგიკურ ელემენტს, რომელიც ახორციელებს პირსის ფუნქციას აქვს ნახ. 3.4-ზე წარმოდგენილი სახე.

ჭეშმარიტების ცხრილში (ცხრილი 3.1) ადრე მოყვანილი ცხრილისაგან განსხვავებით გამოსასვლელ y ცვლადს რვა მნიშვნელობა აქვს. ეს რიცხვი განისაზღვრება შესასვლელ N ცვლადთა შესაძლო კომბინაციათა რიცხვით, რომელიც ზოგად შემთხვევაში ტოლია: $N = 2^n$, სადაც n შესასვლელ ცვლადთა რიცხვია.



ნახ. 3.4

ცხრილი 3.1

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

ლოგიკური ელემენტები გამოიყენებიან ისეთი ინტეგრალური მიკროსქემების ასაგებად, რომლებიც ასრულებენ სხვადასხვა ლოგიკურ და არითმეტიკულ ოპერაციებს და გააჩნიათ სხვადასხვა ფუნქციური დანიშნულება.

ნებისმიერი სირთულის ლოგიკური ალგებრის ფუნქცია(ლაფ) შეიძლება განვახორციელოთ აღნიშნული ლოგიკის ელემენტების დახმარებით. მაგალითის სახით განვიხილოთ შემდეგი ალგებრული ფორმით წარმოდგენილი ლაფ:

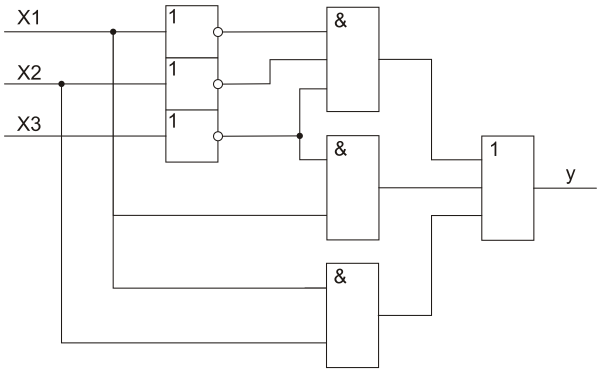
$$y = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2 \cdot \bar{x}_3 \vee x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2 \cdot x_3 \quad (3.1)$$

გავამარტივოთ მოცემული ლაფ ლოგიკის კანონების დახმარებით. მივიღებთ:

$$y = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2 \cdot \bar{x}_3 \vee x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2 \cdot x_3 =$$
$$= \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2(\bar{x}_3 \vee x_3) \vee x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 =$$
$$= \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2 \vee x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1(x_2 \vee \bar{x}_2 \cdot \bar{x}_3) =$$
$$= \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1(x_2 \vee \bar{x}_2)(x_2 \vee \bar{x}_3) = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1(x_2 \vee \bar{x}_3) =$$
$$= \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2 \vee x_1 \cdot \bar{x}_3 \quad (3.2)$$

აღნიშნულ ოპერაციის ლაფ-ის მინიმიზაცია ჰქვია და ემსახურება შესაბამისი ციფრული მოწყობილობის ფუნქციური სქემების აგების პროცედურის გამარტივებას.

განხილული ლაფ-ის შესაბამისი მოწყობილობის ფუნქციონალური სქემა მოყვანილია ნახ. 3.5-ზე



ნახ. 3.5

უნდა აღინიშნოს, რომ გარდაქმნების შედეგად მიღებული ფუნქცია (3.2) არაა სრულად მინიმიზირებული. სრული მინიმიზა-ცია დამატებით უნდა შესრულდეს.

3.2. მიმდევრობითი ლოგიკის ელემენტები.

3.2.1. ტრიგერები.

ციფრულ ავტომატებში ფუნქციის მნიშვნელობა დამოკიდებუ-ლია არა მხოლოდ ცვლადების მნიშვნელობებისაგან დროის მო-ცემულ მომენტში (მოცემულ ტაქტში), არამედ მათი თანმიმდევრო-ბისგანაც წინა მომენტებში (ტაქტებში). ამიტომ ლოგიკის ალგებრის იმ ნაწილს, რომელიც აღწერს ციფრული ავტომატების მუშაობას, რომლებსაც გააჩნიათ მეხსიერება, უწოდებენ მიმდევრობით ლოგი-კას. მიმდევრობითი ლოგიკის ძირითად ელემენტს წარმოადგენს მეხსიერების ტრიგერული ელემენტი, ან უბრალოდ ტრიგერი.

ტ რ ი გ ე რ ი მოწყობილობაა წონასწორობის ორი მდგრადი მდგომარეობით და მისი დანიშნულებაა ორობითი ინფორმაციის ჩა-წერა და დახსომება. შესასვლელი სიგნალების ზემოქმედებით ტრი-გერი შეიძლება გადაერთოს წონასწორობის ერთი მდგრადი მდგომა-რეობიდან მეორეში. ამ დროს ძაბვა მის გამოსასვლელზე ნახტომისე-ბურად იცვლება. როგორც წესი, ტრიგერს გააჩნია ორი გამოსასვლე-ლი: პირდაპირი Q და ინვერსიული $\bar{Q}$. შესასვლელთა რაოდენობა დამოკიდებულია შესასვლელი ფუნქციებისაგან.

ტრიგერების კლასიფიკაცია ხორციელდება ლოგიკური ფუნქ-ციონირების და ინფორმაციის ჩაწერის ხერხის მიხედვით.

ლოგიკური ფუნქციონირების მიხედვით განასხვავებენ *RS, D, T, JK* ტრიგერებს, კომბინირებულ და რთული შესავალი ლოგიკის მქონე ტრიგერებს:

ინფორმაციის ჩაწერის ხერხის მიხედვით ტრიგერები იყოფიან ორ ჯგუფად: ა ს ი ნ ქ რ ო ნ უ ლ ე ბ ა დ (არატაქტირებულებად) და ს ი ნ ქ რ ო ნ უ ლ ე ბ ა დ (ტაქტირებულებად). ასინქრონულ ტრიგერებში ინფორმაცია შეიძლება იცვლებოდეს დროის ნებისმიერ დისკრეტულ მომენტში შესასვლელი სიგნალების ცვლილებისას, ხოლო სინქრონულ ტრიგერებში - დროის განსაზღვრულ მომენტებში, რომელიც განისაზღვრება დამატებითი ტაქტური (სინქრონიზაციის) იმპულსით. სინქრონიზაციის იმპულსის აღქმის მიხედვით განასხვავებენ სტატიკურ და დინამიურ ტრიგერებს.

ტრიგერის ლოგიკურ სტრუქტურაში გამოყენებული მეხსიერების უჯრედების რაოდენობის მიხედვით განასხვავებენ ე რ თ ს აფ ე ხ უ რ ი ა ნ და ო რ ს ა ფ ე ხ უ რ ი ა ნ ტრიგერებს.

ტრიგერთა მთელი ეს მრავალსახეობა სხვადასხვა ფუნქციური შესაძლებლობებით ხასიათდება. ოღონდ ყველა სქემას საფუძვლად უდევს ბაზური ასინქრონული *RS*-ტრიგერი.

ასინქრონული RS-ტრიგერი.

ეს ტრიგერი შეიძლება აიგოს ორ ლოგიკურ „ან-არა“ ან „და-არა“ ელემენტებზე (ნახ. 3.6). ელემენტები მოცულნი არიან უკუ კავშირის წრედებით, რისთვისაც თითოეული ელემენტის გამოსასვლელი მიერთებულია მეორე ელემენტის ერთ-ერთ შესასვლელზე. ტრიგერს ორი შესასვლელი გააჩნია: S - ერთის მდგომარეობაში დაყენების შესასვლელი (ინგლისური სიტყვიდან *set* -დაყენება) და R -ნულის მდგომარეობაში დაყენების შესასვლელი (ინგლისური სიტყვიდან *reset* -ჩამოყრა). როგორც ნახ. 3.6,ა-დან გამომდინარეობს $S = 1$ და $R = 0$ -ის დროს გამოსასვლელზე გვექნება:

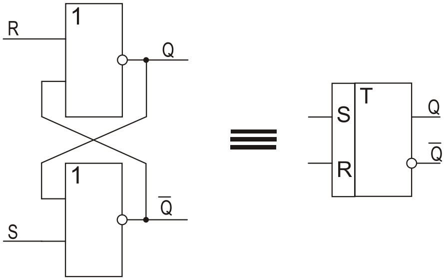
$$\bar{Q} = \overline{S \vee Q} = \overline{1 \vee Q} = 0,$$

$$Q = \overline{R \vee \bar{Q}} = \overline{0 \vee 0} = 1,$$

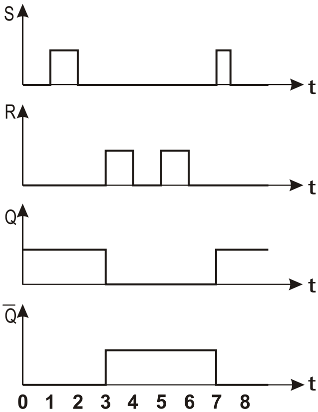
ხოლო $S = 0$ და $R = 1$ -ის დროს კი:

$$Q = \overline{R \vee \bar{Q}} = \overline{1 \vee \bar{Q}} = 0,$$
$$\bar{Q} = \overline{S \vee Q} = \overline{0 \vee 0} = 1.$$

შესასვლელი სიგნალების მოხსნის შემდეგ, ანუ როცა $S = R = 0$, ხდება გამოსასვლელი სიგნალის შენარჩუნება, 1-ის ან 0-ის იმისგან დამოკიდებულებით თუ რომელ შესასვლელზე (შესაბამისად S-ზე ან R-ზე) იყო მანამდე ერთიანი. ყველაფერი ზემოთ აღნიშნული შეიძლება განვიხილოთ (ნახ. 3.6,ბ)- ზე წარმოდგენილ დიაგრამებზე:



ა)



ბ) ნახ. 3.6.

0-ვანი ტაქტი. R და S სიგნალები (შესასვლელი) 0-ის ტოლია. ტრიგერი ამ მომენტში იმყოფება ერთის მდგომარეობაში ($Q = 1$). აუ-

ცილებელია დავაზუსტოთ, რომ დროის საწყის მომენტში ტრიგერის მდგომარეობა შემთხვევითი სიდიდეა და ჩვენ მხოლოდ განსაზღვრულობის მიზნით ვიწყებთ ანალიზს $Q = 1$ -იდან.

<u>I ტაქტი.</u> შესასვლელი სიგნალი $S = 1$. ტრიგერი იძულებით დგება ერთის მდგომარეობაში, მაგრამ ვინაიდან Q უკვე უდრის 1-ს, ამიტომ ტრიგერის გამოსასვლელებზე მდგომარეობა არ იცვლება.

<u>მე - 2 ტაქტი.</u> $S = R = 0$. წინა ტაქტში ჩაწერილი ინფორმაციის შენახვის რეჟიმი: $Q = 1,\ \bar{Q} = 0$.

<u>მე - 3 ტაქტი.</u> $S = 0,\ R = 1$. ტრიგერი იძულებით დგება ნულოვან მდგომარეობაში. ამასთან Q და $\bar{Q}$ მდგომარეობები საწინააღმდეგოზე იცვლებიან.

<u>მე - 4 ტაქტი.</u> შენახვის რეჟიმი.

<u>მე - 5 ტაქტი.</u> $S = 0,\ R = 1$. იძულებითი დაყენება 0-ის მდგომარეობაში. მაგრამ ვინაიდან $\bar{Q}$ უკვე 0-ის ტოლი იყო, ამიტომ ტრიგერის გამოსასვლელებზე მდგომარეობები არ იცვლება.

<u>მე - 6 ტაქტი.</u> $\bar{S} = R = 0$. შენახვის რეჟიმი.

<u>მე - 7 ტაქტი.</u> $S = 1,\ R = 0$. იძულებითი დაყენება 1-ის მდგომარეობაში. ტრიგერის გამოსასვლელებზე ჩნდებიან სიგნალები $Q = 1$ და $\bar{Q} = 0$.

წარმოდგენილი ანალიზის საფუძველზე შეგვიძლია დავასკვნათ:

1) $S = 1$ და $R = 0$ -ის დროს ხდება ტრიგერის დაყენება $Q = 1$ და $\bar{Q} = 0$ მდგრად მდგომარეობაში (ერთის ჩაწერა);

2) $S = 0$ და $R = 1$-ის დროს ხდება ტრიგერის დაყენება $Q = 0$ და $\bar{Q} = 1$ -ის მდგრად მდგომარეობაში (ნულის ჩაწერა);

3) $S = R = 0$ -ის დროს ტრიგერი ინარჩუნებს იმ მდგრად მდგომარეობას, რომელიც მას ჰქონდა ამ სიგნალის მოწოდებამდე (დახსომების რეჟიმი).

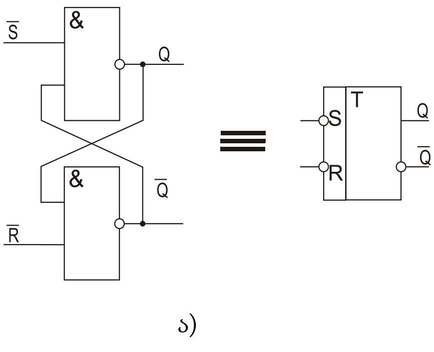
ამრიგად, ტრიგერის მდგომარეობა (ჩაწერილი ინფორმაცია) შეგვიძლია განვსაზღვროთ ან Q გამოსასვლელზე სიგნალით, ან $\bar{Q}$ გა-

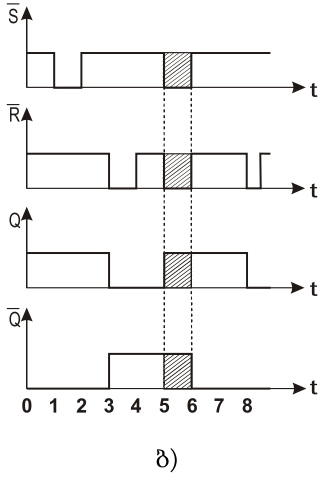
მოსასვლელზე ჩაწერილი სიგნალის ინვერსიით. მაგრამ $S = R = 1$ -ის დროს ორივე გამოსასვლელი სიგნალი

$$Q = \overline{R \vee \bar{Q}} = \overline{1 \vee \bar{Q}} = 0,$$
$$\bar{Q} = \overline{S \vee Q} = \overline{0 \vee Q} = 0$$

ტოლია ნულის, რაც საშუალებას არ გვაძლევს ცალსახად განვსაზღვროთ სისტემის მდგომარეობა. ამიტომ შესასვლელ სიგნალთა $S = R = 1$ კომბინაცია აკრძალულია.

დავალება: ჩაატარეთ დამოუკიდებლად მსგავსი ანალიზი „და-არა“ ელემენტებზე აგებულ RS-ტრიგერის სქემაზე (ნახ. 3.7,ა) და დარწმუნდით, რომ მეორე სქემა მუშაობს პირველის მსგავსად შესასვლელი სიგნალების ინვერსიულებზე შეცვლის დროს 0-ის ლოგიკური დონით. დიაგრამებზე (ნახ. 3.7,ბ) დაშტრიხული უბნებით გამოსახულია დრო, რომლის განმავლობაში RS-ტრიგერის შესასვლელებზე მოქმედებს შესასვლელი სიგნალების აკრძალული კომბინაცია: $\bar{S} = \bar{R} = 0$.





ნახ. 3.7.

RS -ტრიგერის მუშაობის აღწერა ჩვენ ჩავატარეთ ანალიზური და გრაფიკული ხერხებით, ოღონდ ამის გაკეთება შესაძლებელია გადართვის ცხრილის დახმარებით (ცხრ. 3.2).

დამოუკიდებელი მოწყობილობის სახით RS -ტრიგერის ფართო გავრცელებას ხელს უშლის მისთვის დამახასიათებელი სერიოზული ნაკლოვანებები: შესასვლელ სიგნალთა აკრძალული კომბინაციის არსებობა, ინფორმაციის მიწოდება ორი განცალკევებული წრედით (R, S), შეშფოთებებისადმი დაბალი მდგრადობა.

ცხრილი 3.2

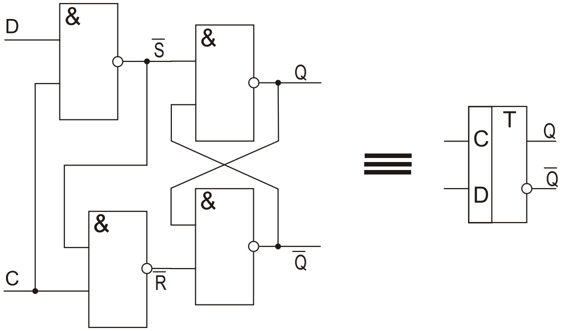
შესასვლელი				გამოსასვლელი		მუშაობის რეჟიმი
„ან-არა“		„და-არა“		Q	$\bar{Q}$	
S	R	$\bar{S}$	$\bar{R}$			
0	0	1	1	0	0	დახსომება
1	0	0	1	1	0	1-ის ჩაწერა
0	1	1	0	0	1	0-ის ჩაწერა
1	1	0	0	X	X	აკრძალული $(Q = \bar{Q})$

სინქრონული D - ტრიგერი.

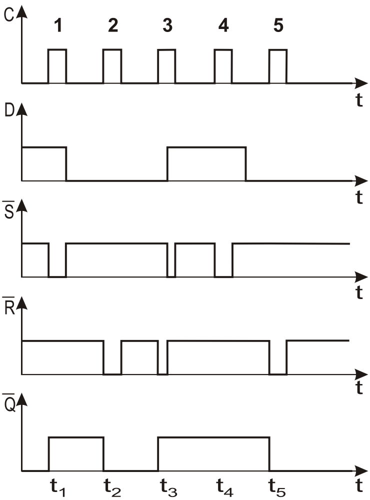
სინქრონული D - ტრიგერი თავისუფალია RS -ტრიგერის ნაკლოვანებისაგან. ის შედგება RS -ტრიგერისა და ორ ლოგიკურ ელემენტზე აგებული კომბინაციური სქემისაგან (ნახ. 3.8,ა). ტრიგერში შესატანად გამიზნული სიგნალები მიეწოდებიან ინფორმაციულ D შესასვლელს. სინქრონიზაციის C შესასვლელზე მიეწოდებიან სინქრონიზაციის იმპულსები, რომლებიც განსაზღვრავენ ინფორმაციის ჩაწერის მომემტს. ტრიგერის მუშაობის აღწერა შესასვლელ სიგნალთა სხვადასხვა კომბინაციებისათვის მოყვანილია ცხრილ 3.3-ში.

ცხრილი 3.3

C	D	$\bar{S} = \overline{C \cdot D}$	$\bar{R} = \overline{C \cdot \bar{S}}$	Q	მუშაობის რეჟიმი
0	0	$\overline{0 \cdot 0} = 1$	$\overline{0 \cdot 1} = 1$	წინა მნიშვნელობა	დახსომება
0	1	$\overline{0 \cdot 1} = 1$	$\overline{0 \cdot 1} = 1$	წინა მნიშვნელობა	დახსომება
1	0	$\overline{1 \cdot 0} = 1$	$\overline{1 \cdot 1} = 0$	0	0-ის ჩაწერა
1	1	$\overline{1 \cdot 1} = 0$	$\overline{1 \cdot 0} = 1$	1	1-ის ჩაწერა



ა)

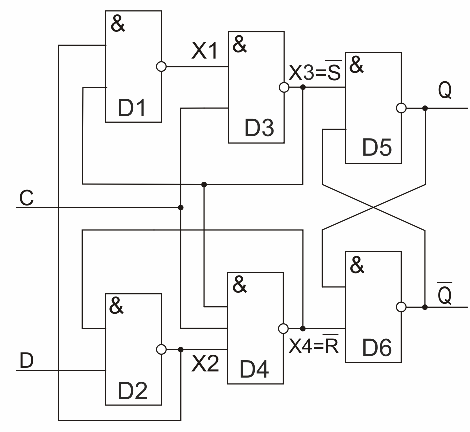


ბ) ნახ.3.8.

ცხრილიდან და დროითი დიაგრამებიდან (ნახ. 3.8,ბ), რომლე-ბიც აღწერენ ინფორმაციის შენახვისა და ჩაწერის პროცესს, ჩანს, რომ D - ტრიგერი იმყოფება შენახვის რეჟიმში როცა $C = 0$ ჩაწერის რეჟიმში $C = 1$ -ის დროს. ასეთი ტრიგერი აყოვნებს გამოსავალ სიგ-ნალს იმ ტაქტის დასრულებამდე, რომელშიც ის იქნა ჩაწერილი. ასე, შესასვლელი სიგნალი $D = 1$ მთავრდება პირველ და მეორე, მეოთხე და მეხუთე სინქრონიზაციის იმპულსებს შორის, ხოლო მდგომარეო-ბა $Q = 1$ შენარჩუნებულია მეორე და მეხუთე ტაქტების ბოლომდე. სწორედ აქედან წარმოიშვა დასახელება D - ტრიგერი (ინგლისური-დან delay - დაყოვნება). თუ სიგნალი შესასვლელზე შეიცვლება სინ-ქრონიზაციის იმპულსის მოქმედების განმავლობაში, მაშინ ტრიგერ-ში ჩაწერილი აღმოჩნდება ის ინფორმაცია, რომელიც წინ უძღოდა სინქრონიზაციის იმპულსის დასრულებას - მომენტი t_3 (ნახ. 3.8,ბ). ამ თვისების წყალობით (ინფორმაციის ცვლილება მთელი დროის განმავლობაში, სანამ $C = 1$) განხილულ ტრიგერს უწოდებენ ს ტ ა - ტ ი კ უ რ ს ი ნ ქ რ ო ნ უ ლ D - ტ რ ი გ ე რ ს. სტატიკური D - ტრი-გერის ნორმალური მუშაობისათვის აუცილებელია, რომ ინფორ-მაციის ცვლილება D -შესასვლელზე ხდებოდეს მხოლოდ $C = 0$ -ის დროს.

დინამიური სინქრონული D - ტრიგერი.

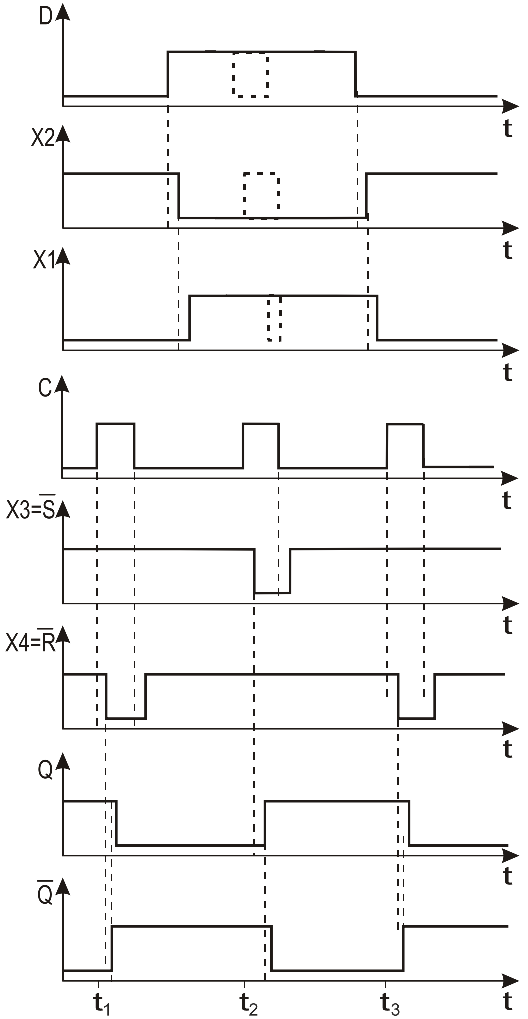
დინამიური სინქრონული D - ტრიგერი გამორიცხავს სინქრო-ნიზაციის იმპულსის მოქმედების განმავლობაში სიგნალის გამჭოლ გადაცემას D-შესასვლელიდან მის გამოსასვლელზე. დინამიური მართვის მქონე ტრიგერში ინფორმაცია იწერება მხოლოდ ძაბვის გა-დავარდნის მომენტში სინქრონიზაციის შესასვლელზე. დინამიური D - ტრიგერის სქემა მოყვანილია ნახ. 3.9,ა-ზე, ხოლო დროითი დიაგ-რამა სიგნალებისათვის ტრიგერის სხვადასხვა წერტილებში-ნახ. 3.9,ბ-ზე.



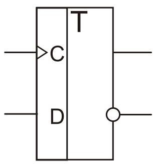
ნახ. 3.9, ა)

განვიხილოთ ტრიგერის მუშაობა იმის გათვალისწინებით, რომ დროის ყოველ მომენტში ლოგიკურ ელემენტთა გამოსასვლელ სიგნალთა მნიშვნელობები ტოლია: $X_2 = \overline{D \cdot X_4}$, $X_1 = \overline{X_2 \cdot X_3}$, $X_3 = \overline{C \cdot X_1}$, $X_4 = \overline{C \cdot X_2 \cdot X_3}$. სანამ C სიგნალი ნულის ტოლია, $X_3 = X_4 = 1$. ამიტომ სინქრონიზაციის იმპულსებს შორის პაუზებში გამოსასვლელი ტრიგერი იმყოფება ინფორმაციის შენახვის რეჟიმში, ხოლო სიგნალები პირველი ორი ელემენტის გამოსასვლელებზე $(D1, D2)$ მთლიანად განისაზღვრება შესასვლელი ინფორმაციული სიგნალით: $X_2 = \overline{D \cdot 1} = \overline{D}$ და $X_1 = \overline{\overline{D} \cdot 1} = D$. X_2 და X_1 სიგნალები ინვერსიულებია ერთმანეთის მიმართ, ამიტომ $C = 1$-ის გამოჩენისას მხოლოდ ერთ-ერთი მათგანი აძლევს ნებას სინქრონიზაციის იმპულსის გავლას $D3$ ან $D4$ ელემენტში:

$C = 1$ და $D = 1$,	გვაქვს	$\bar{S} = X_3 = \overline{C \cdot D} = \overline{1 \cdot 1} = 0,$ $\bar{R} = X_4 = \overline{C \cdot \overline{D} \cdot X_3} = \overline{1 \cdot 0 \cdot 0} = 1.$	- ერთის ჩაწერის რეჟიმი
$C = 1$ და $D = 0$,	გვაქვს	$\bar{S} = X_3 = \overline{1 \cdot 0} = 1,$ $\bar{R} = X_4 = \overline{1 \cdot 1 \cdot 1} = 0.$	- ნულის ჩაწერის რეჟიმი



ბ)



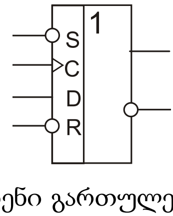
გ)

ნახ. 3.9.

დროითი დიაგრამები (ნახ. 3.9,ბ) აგებულია თითოეულ ლოგიკურ ელემენტში სიგნალთა გავრცელების დაყოვნების გათვალისწი-

ნებით (ნაჩვენებია წვრილი ხაზებით). წყვეტილი ხაზებით დიაგრამებზე D, X_1 და X_2-ისათვის ნაჩვენებია შემთხვევა, როდესაც ინფორმაციული D სიგნალის ცვლილება ხდება სინქრონიზაციის იმპულსების მოქმედების დროს. ეს სიგნალი გადის მხოლოდ პირველი ელემენტების გამოსასვლელებზე და არ გაიშვება $D3$ და $D4$ ელემენტებით. ამრიგად, დინამიური მართვის მქონე განხილულ ტრიგერში სიგნალის ჩაწერა ხდება სინქრონიზაციის იმპულსის ფრონტით. დინამიური ტრიგერის პირობითი გრაფიკული გამოსახულება (ნახ. 3.9,გ) განსხვავდება სტატიკური ტრიგერის გამოსახულებისაგან (ტრიგერი პოტენციური მართვით) იმით, რომ სინქრონიზაციის დინამიური შესასვლელი სქემაზე გამოსახულია სამკუთხედით. თუ სამკუთხედის წვერო მიმართულია მიკროსქემის მიმართულებით, მაშინ ტრიგერი ირთვება სინქრონიზაციის იმპულსის წინა ფრონტით, თუ პირიქით - უკანა ფრონტით.

დინამიური D - ტრიგერი შედგება სამი სტატიკური RS-ტრიგერისაგან. პირველი ორი ელემენტი, შესაბამისად აგებულები $D1, D2$ და $D3, D4$ ელემენტებზე, ახორციელებს ინფორმაციის მომზადებას. მესამე ტრიგერი ($D5, D6$) იწერს ლოგიკურად უკვე წინასწარ დამუშავებულ ინფორმაციას. დინამიური ტრიგერის სწორედ ასეთმა ორსაფეხუროვანმა აგებამ უზრუნველყო სინქრონიზაციის იმპულსის მოქმედების განმავლობაში D შესასვლელიდან ტრიგერის გამოსასვლელზე სიგნალის პირდაპირი გასვლის თავიდან აცილება.



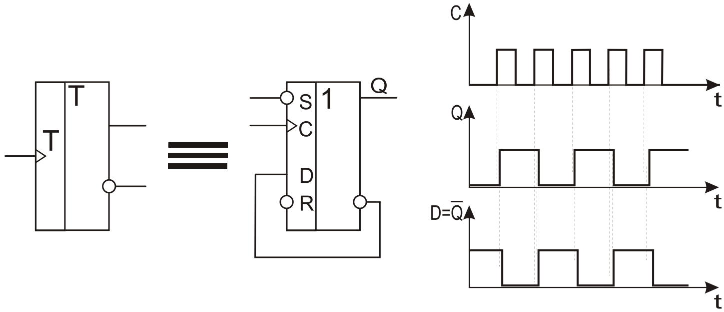
ნახ. 3.10

სქემის მცირეოდენი გართულებით (ნახ. 3.9,ა) შეიძლება მივიღოთ უნივერსალური D - ტრიგერი (ნახ. 3.10), რომელიც ასრულებს როგორც RS-ტრიგერის, ისე დინამიური D - ტრიგერის ფუნქციებს. მოდერნიზაცია მდგომარეობს ყველა ორშესასვლელიანი „და-არა“

ელემენტის შეცვლაში სამშესასვლელიან „და-არა“ სქემებზე. $D1$ და $D5$ ელემენტების გაჩენილი დამატებითი შესასვლელები წარმოადგენენ $\bar{S}$ სიგნალისთვის შესასვლელს, ხოლო $D2$, $D3$ და $D6$ ელემენტები - $\bar{R}$ სიგნალის შესასვლელებს. სანამ სიგნალი $\bar{S}$-ზე და $\bar{R}$-ზე ერთის ტოლია, უნივერსალური ტრიგერი მუშაობს როგორც დინამიური D - ტრიგერი D და C შესასვლელებით. როგორც კი რომელიმე შესასვლელზე ($\bar{S}$ ან $\bar{R}$-ზე) მიეწოდება ნულის ტოლი სიგნალი, ტრიგერი მაშინვე წყვეტს რეაგირებას C და D სიგნალებზე და ღებულობს $\bar{S}$ და $\bar{R}$ სიგნალებით განსაზღვრულ მდგომარეობას. მდგომარეობა $\bar{S} = \bar{R} = 0$ ისევ აკრძალულია.

მთვლელი T - ტრიგერი.

T - ტრიგერს გააჩნია ერთი მართვადი შესასვლელი და ორი გამოსასვლელი (ნახ. 3.11,ა). ასეთი ტრიგერის გამოსასვლელებზე ინფორმაცია იცვლება საწინააღმდეგოზე ძაბვის ყოველი დადებითი გადავარდნისას სათვლელ T შესასვლელზე, ამიტომ მთვლელ ტრიგერს იყენებენ შესასვლელი სიხშირის გამყოფად. ამ ტიპის ტრიგერი შეიძლება აიგოს დინამიური მართვის მქონე D - ტრიგერიდან, თუ მის ინვერსიულ გამოსასვლელს მივუერთებთ ინფორმაციულ შესასვლელს (ნახ. 3.11,ბ). ამასთან, თუ დროის საწყის მომენტში Q გამოსასვლელზე ნულოვანი დონეა, მაშინ D შესასვლელზე - $\bar{Q} = 1$ დონეა. სინქრონიზაციის პირველი იმპულსის წინა ფრონტით ერთიანი D-შესასვლელიდან გადაიწერება (დაგვიანებით, რაც ერთი ლოგიკური ელემენტის დაყოვნების ტოლია: ნახ. 3.11,გ და ნახ. 3.9,ბ) Q გამოსასვლელზე. შესაბამისად $\bar{Q}$ გამოსასვლელზე და D შესასვლელზე შეიქმნება ნულოვანი დონე (დაყოვნებით, რაც ორი ლოგიკური ელემენტის დაყოვნების ტოლია). შემდეგ ტაქტზე Q გამოსასვლელზე გადაიწერება ნულოვანი მნიშვნელობა D შესასვლელიდან და ა. შ.



ა)　　　　　　　ბ)　　　　　　　გ)

ნახ. 3.11

მთვლელი ტრიგერის ასე აგება სტატიკური D - ტრიგერის ბაზაზე (უკუ კავშირით $\bar{Q}$ გამოსასვლელიდან D შესასვლელზე) შეუძლებელია. ვინაიდან სტატიკურ ტრიგერს აქვს პოტენციური მართვა, მაშინ $C = 1$ ძაბვა გამოსასვლელზე უკუ კავშირის ზემოქმედების გამო მუდმივად იცვლება საწინააღმდეგოზე, ანუ შეიქმნებიან მაღალსიხშირული რხევები. ამაში ადვილად დავრწმუნდებით, თუ ავაგებთ ნახ.3.8,გ-ზე მოყვანილი დროითი დიაგრამების მსგავსებს ლოგიკურ ელემენტებში სიგნალების დაყოვნების გათვალისწინებით.

ტრიგერთა სხვა ტიპები.

როგორც ზემოთ ავღნიშნეთ, არსებობს ტრიგერთა დიდი მრავალფეროვნება, რომლებიც განხილულებისაგან განსხვავდებიან სხვა შესასვლელი ლოგიკით. ასე, მაგალითად, ცნობილია სინქრონული RS -ტრიგერები, DV -ტრიგერები ორი მართვადი შესასვლელით, უნივერსალური JK -ტრიგერები და სხვა. კიდევ ერთხელ გავუსვათ იმას ხაზი, რომ ტრიგერთა ყველა სახე დაფუძნებულია მიმდევრობითი ლოგიკის ბაზური ელემენტის - ასინქრონული RS - ტრიგერის - გამოყენებაზე.

ჩვენს მიერ განხილული ტრიგერთა ტიპები: სტატიკური ასინქრონული RS -ტრიგერი (პოტენციური მართვით), სტატიკური სინქრონული D- ტრიგერი (დინამიური მართვით), მთვლელი T - ტრიგერი (დინამიური მართვით) - ძირითადებია ინტეგრალურ მიკროსქემათა სერიებში და საშუალებას იძლევიან მათ ბაზაზე აიგოს ნებისმიერი ციფრული კვანძები და მიმდევრობითი ტიპის მოწყობილობები.

3.3. ციფრული ტექნიკის ძირითადი ოპერაციული ელემენტები.

რთულ სისტემებში ციფრული ინფორმაციის დამუშავება ხდება ცალკეული ელემენტალური ოპერაციების მიმდევრობითი შესრულების სახით. ეს ელემენტალური ოპერაციები სრულდებიან ოპერაციული ელემენტებით. ციფრული მოწყობილობების ოპერაციული ელემენტების, ან კვანძების, აგება ხდება კომბინაციური ან მიმდევრობითი ლოგიკის ლოგიკური ელემენტებისაგან.

ელემენტალურ ოპერაციათა ძირითადი ნაკრები დიდი არ არის.

დ ა ყ ე ნ ე ბ ა - ოპერაციულ ელემენტში რაიმე კონსტანტის ორობითი კოდის ჩაწერა. მაგალითი - მთვლელის ყველა თანრიგში ნულის დაყენება.

გ ა დ ა ც ე მ ა - მ ი ღ ე ბ ა - რიცხვის კოდის გადაწერა ერთი ოპერაციული ელემენტიდან მეორეში.

წ ა ნ ა ც ვ ლ ე ბ ა - კოდის თანრიგთა მდგომარეობის შეცვლა პირვანდელ მდგომარეობასთან შედარებით.

თ ვ ლ ა - ოპერაციული ელემენტის გამოსასვლელზე რიცხვის კოდის გაზრდა ან შემცირება მის შესასვლელზე იმპულსთა თანმიმდევრობის მიწოდებისას.

გ ა რ დ ა ქ მ ნ ა - რიცხვის კოდის გადაყვანა კოდირების ერთი სისტემიდან მეორეში.

გ ა ნ ა წ ი ლ ე ბ ა - ბევრი წყაროდან ერთ მომხმარებელზე ან ერთი წყაროდან რამდენიმე მომხმარებელზე სიგნალების მისამართების მიხედვით გადაცემა.

შ ე კ რ ე ბ ა - ორობით კოდში წარმოდგენილი ორი რიცხვის ჯამის პოვნა.

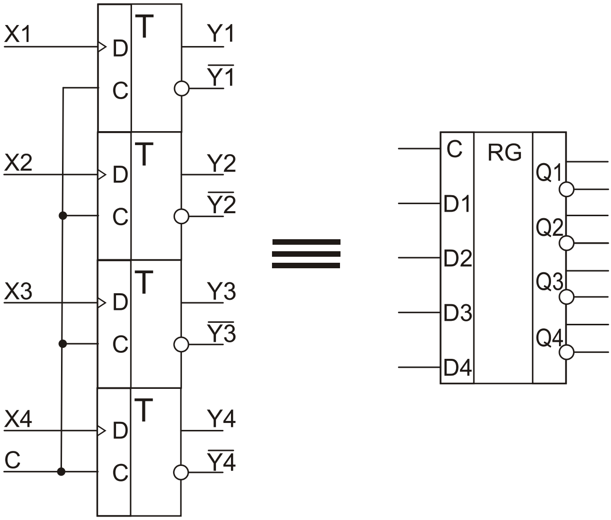
კვანძები, რომლებიც ასრულებენ ძირითად ელემენტალურ ოპერაციებს, ასევე იწოდებიან ციფრულ მოწყობილობათა ძირითად კვანძებად. მათ მიეკუთვნებიან რეგისტრები, მთვლელები, კოდების გარდამქნელები, მულტიპლექსორები და ამჯამავები. განვიხილოთ ეს მოწყობილობები ცალ-ცალკე.

3.3.1. რეგისტრები.

რეგისტრი ეწოდება ტრიგერებისაგან აგებულ ოპერაციულ ელემენტს, რომლის ძირითადი დანიშნულებაა რიცხვების მიღება და შენახვა. ამ რიცხვთა თანრიგთა ციფრები წარმოდგენილი არიან ორობით კოდში. მაგრამ ზოგიერთი სახის რეგისტრის საშუალებით შეიძლება შესრულდეს შემდეგი ელემენტალური ოპერაციები: დაყენება, წანაცვლება, გარდაქმნა. რეგისტრთა ძირითად ტიპებს წარმოადგენენ პარალელური და მიმდევრობითი რეგისტრები.

სინქრონულ D- ტრიგერზე აგებულ პ ა რ ა ლ ე ლ უ რ რ ე - გ ი ს ტ რ შ ი დამახსოვრებული რიცხვის კოდი მიეწოდება ყველა ტრიგერის ინფორმაციულ შესასვლელს და იწერება რეგისტრში ტაქტური იმპულსის მოსვლის დროს. გამოსასვლელი ინფორმაცია იცვლება ახალი შესასვლელი სიტყვის მიწოდებით და ჩაწერის შემდეგი იმპულსის მოსვლით. ასეთ რეგისტრებს იყენებენ ოპერაციული მეხსიერების სისტემებში. მათში ტრიგერთა რაოდენობა ტოლია შესანა-

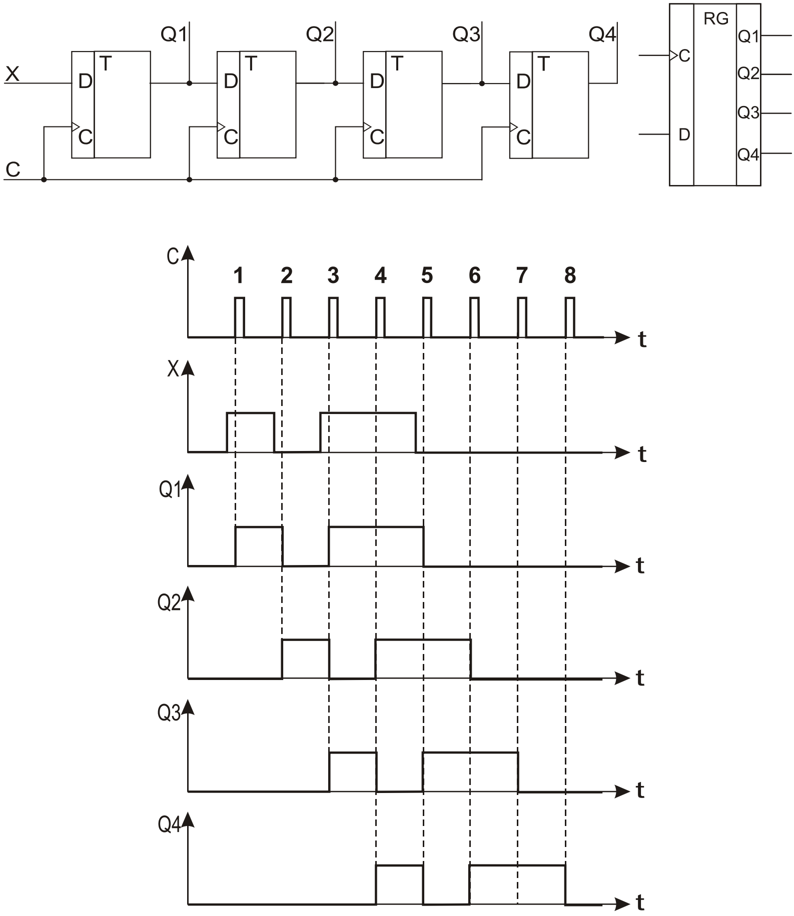
ხი სიტყვის მაქსიმალური თანრიგიანობის. ნახ. 3.12-ზე მოყვანილია ოთხთანრიგიანი რეგისტრი. C შესასვლელების გაერთიანებით ელემენტალურად მივიღებთ უფრო მეტ თანრიგიან რეგისტრებს (8, 12 და ა. შ.).



ნახ. 3.12.

დინამიური მართვის მქონე D- ტრიგერებზე აგებული მიმდევრობითი რეგისტრის სქემა, აღნიშვნა და მისი მუშაობის დროითი დიაგრამა მოყვანილია ნახ. 3.13 - ზე. ტაქტური C იმპულსის მოსვლასთან ერთად პირველი ტრიგერი იწერს X კოდს (0-ს ან 1-ს), რომელიც ამ მომენტში იმყოფება მის D- შესასვლელზე, ხოლო ყოველი შემდეგი ტრიგერი გადაირთვება იმ მდგომარეობაში, რომელშიც მანამდე იმყოფებოდა წინამდებარე ტრიგერი. ასე იმის გამო ხდება, რომ ჩასაწერი სიგნალი გადის D- ტრიგერის შესასვლელიდან მის Q გამოსასვლელზე ისეთი დაყოვნებით, რომელიც მეტია ტაქტური იმპულსის წინა ფრონტის დაყოვნებაზე (რომლის განმავლობაშიც

ხდება ჩაწერა). თითოეული ტაქტური იმპულსი მიმდევრობით



ნახ. 3.13

ძრავს რიცხვის კოდს რეგისტრში ერთი თანრიგით. ამიტომ *n* - თანრიგიანი კოდის ჩასაწერად საჭიროა *n* ტაქტური იმპულსი. დიაგრამიდან გამომდინარეობს, რომ ოთხნიშნა 1011 რიცხვი ჩაწერილი იყო რეგისტრის შესაბამის თანრიგებში (1 - Q4-ში, 0 – Q3-

ში, 1 – Q2-ში, 1 – Q1-ში) მეოთხე ტაქტური იმპულსის მოსვლის შემდეგ. შემდეგი ტაქტური იმპულსის მოსვლამდე ეს რიცხვი ინახება რეგისტრში პარალელური კოდის სახით Q4, …, Q1 გამოსასვლელებზე. შენახული ინფორმაციის მიმდევრობით კოდში მიღების აუცილებლობის შემთხვევაში, მას იღებენ Q4 გამოსასვლელიდან შემდეგი ოთხი იმპულსის (5 – 8) მისვლის დროს. ამ რეჟიმს მ ი მ დ ე ვ რ ო ბ ი თ ი წ ა კ ი თ ხ ვ ი ს რ ე ჟ ი მ ი ეწოდება.

ძალიან მოხერხებულებია უ ნ ი ვ ე რ ს ა ლ უ რ ი რ ე გ ი ს ტ- რ ე ბ ი, რომლებიც საშუალებას იძლევიან მოვახდინოთ როგორც მიმდევრობითი, ისე პარალელური ჩაწერა და წაკითხვა. ასეთი რე- გისტრები შეიძლება გამოვიყენოთ პარალელური კოდის მიმდევრო- ბითში და პირიქით გარდასაქმნელად. უნივერსალური რეგისტრის ბაზაზე შესაძლებელია რევერსიული ძვრის რეგისტრის აგება. ამი- სათვის საჭიროა D1, D2 და D3 შესასვლელების Q2, Q3 და Q4 გამო- სასვლელებთან მიერთება.

3.3.2. კოდების კომბინაციური გარდამქნელები.

კოდების კომბინაციური გარდამქნელის დანიშნულებაა m - თანრიგიანი პარალელური კოდის გარდაქმნა n - თანრიგიან პარა- ლელურ კოდში. შესასვლელ და გამოსასვლელ სიგნალებს შორის კავშირი შეიძლება მოცემული იყოს ჭეშმარიტების ცხრილით ან ლო- გიკური ფუნქციებით. ქვემოთ მოყვანილია ყველაზე მეტად გავ- რცელებული კოდთა გარდამქნელების სახეები.

კოდერი

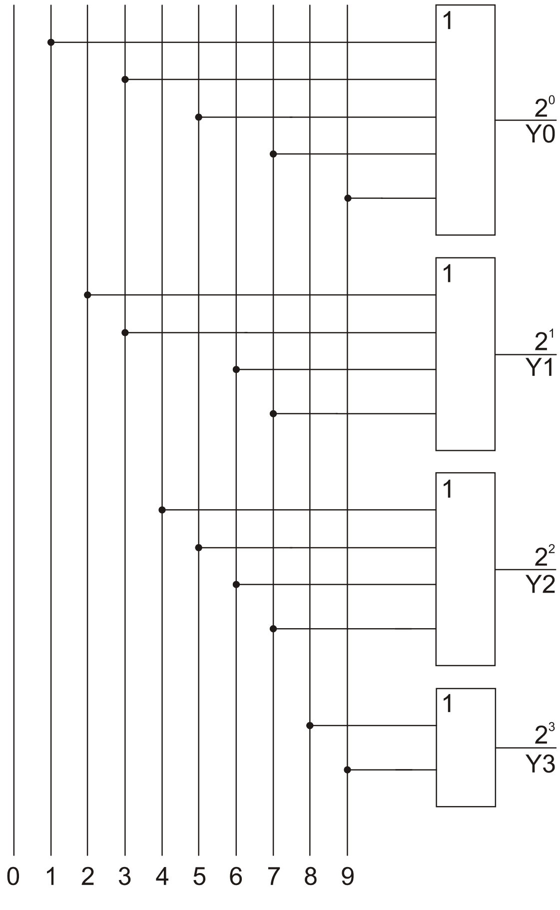
კოდერი გარდაქმნის ერთ-ერთ შესასვლელზე მოსულ ერთეულოვან სიგნალს *n* - თანრიგიან ორობით კოდში. ყველაზე ფართო გავრცელება მან მოიპოვა ინფორმაციის შეტანის მოწყობილობებში (მართვის პულტებში) ათობითი რიცხვების თვლის ორობით სისტემაში გარდასაქმნელად. დავუშვათ, პულტზე 10 კლავიშაა 0...9 ციფრებით. თითოეულ მათგანზე ხელის დაჭერისას კოდერის შესასვლელს მიეწოდება ერთეულოვანი სიგნალი (X0, … ,X9). კოდერის გამოსასვლელზე გამოვა ამ რიცხვის შესაბამისი ორობითი კოდი

ცხრილი 3.4

კ ო დ ე რ ი				
ათობითი რიცხვი X	ორობითი კოდი			
	Y3	Y2	Y1	Y0
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
Y	X3	X2	X1	X0
დ ე კ ო დ ე რ ი				

(Y0, … ,Y3). როგორც ჭეშმარიტების ცხრილიდან ჩანს (ცხრ. 3.4) ამ შემთხვევაში გარდამქმნელი ათი შესასვლელით და ოთხი გამოსასვლელით. Y0 გამოსასვლელზე ერთიანი უნდა მივიღოთ ნებისმიერ კენტ კლავიშზე (X1, X3, X5, X7, X9) ხელის დაჭერით, ანუ $Y0 =$

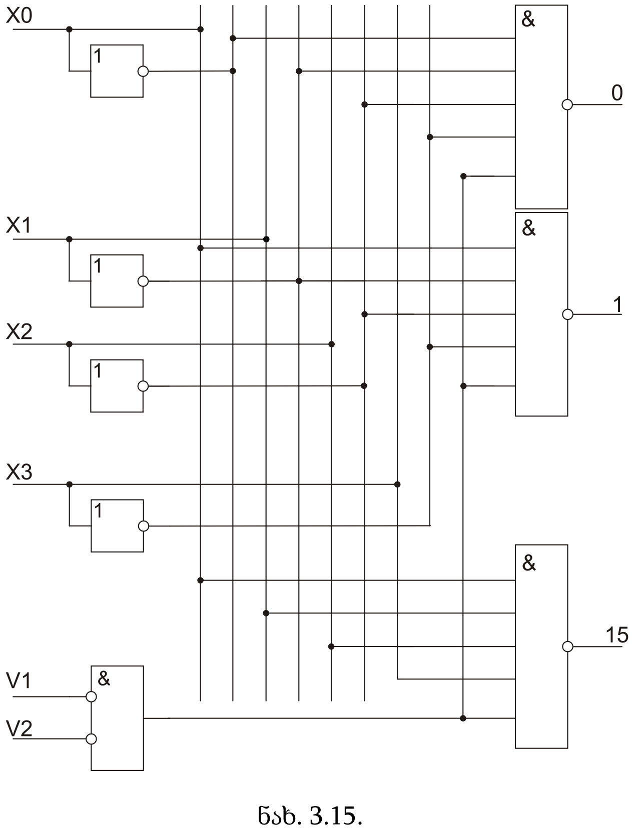
$X1 \vee X3 \vee X5 \vee X7 \vee X9$. სხვა გამოსასვლელთა მდგომარეობა განისაზღვრება ლოგიკური ფუნქციებით: $Y1 = X2 \vee X3 \vee X6 \vee X7$, $Y2 = X4 \vee X5 \vee X6 \vee X7$, $Y3 = X8 \vee X9$. შესაბამისად, კოდერის ასაგებად საჭიროა ოთხი „ან“ სქემა: ხუთშესასვლელიანი, ორი ოთხშესასვლელიანი და ერთიც ორშესასვლელიანი (ნახ. 3.14).



ნახ. 3.14.

დეკოდერი

დეკოდერი (დეკოდერი) გარდაქმნის მის შესასვლელებზე მიწოდებულ კოდს სიგნალში მის ერთ გამოსასვლელზე. დეშიფრატორები ფართოდ გამოიყენებიან მართვის მოწყობილობებში, ინდიკაციის სქემებში იმპულსების განმანაწილებლების ასაგებად და ა. შ.



ნახ. 3.15.

ორობითი n - თანრიგა კოდის დეშიფრატორს აქვს 2^n გამოსასვლელი (ვინაიდან შესასვლელი კოდის 2^n მნიშვნელობიდან ყოველს უნდა შეესაბამებოდეს ერთეულოვანი სიგნალი დეკოდერის ერთ-

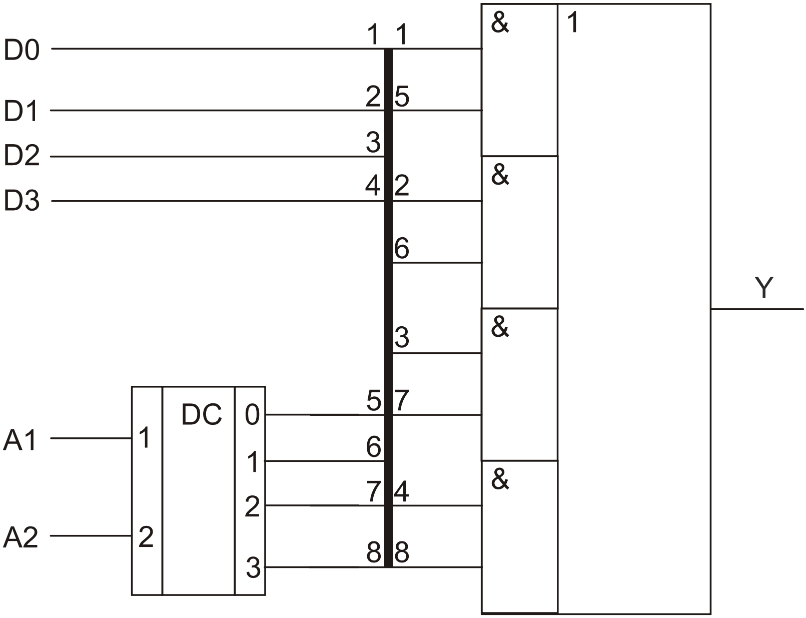
ერთ გამოსასვლელზე). ჭეშმარიტების ცხრილი ორთანრიგა ორობითი კოდის ათობით ციფრებში გარდამქნელი დეკოდერისათვის შეიძლება მიღებულ იქნას ცხრილ 3.4 -დან, თუ ჩავთვლით ორობით კოდს X3, … , X0 შესატყვის სიტყვად, ხოლო ათობითს - გამოსასვლელად. ასეთი დეკოდერის ლოგიკური ფუნქციები ფრიად მარტივია: $Y0 = \overline{X3} \cdot \overline{X2} \cdot \overline{X1} \cdot \overline{X0}$, $Y1 = \overline{X3} \cdot \overline{X2} \cdot \overline{X1} \cdot X0$, $Y2 = \overline{X3} \cdot \overline{X2} \cdot X1 \cdot \overline{X0}$, … , $Y15 = X3 \cdot X2 \cdot X1 \cdot X0$. ორობითი კოდის კოდ „1 თექვსმეტიდან" დეკოდერის სქემა მოყვანილია ნახ. 3.15 -ზე.

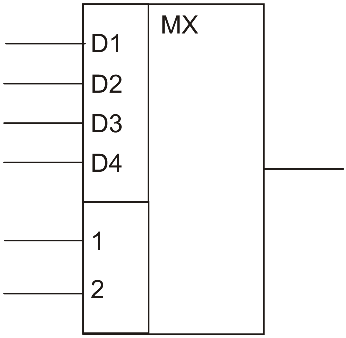
<u>მულტიპლექსორი</u>

მულტიპლექსორის დანიშნულებაა პარალელური კოდის მიმდევრობითში გარდაქმნა. ოთხშესასვლელიანი მულტიპლექსორის სქემა და პირობითი აღნიშვნა მოყვანილია ნახ. 3.16-ზე. მულტიპლექსორს გააჩნია ორი სახის შესასვლელები: ინფორმაციული (D) და სამისამართო (A). ინფორმაციული ხაზის არჩევა ხდება სამისამართო შესასვლელზე მიწოდებული კოდით. ამიტომ ასეთი მოწყობილობის Q გამოსასვლელზე გადაეცემა იმ D_i ინფორმაციული შესასვლელის ლოგიკური დონე, რომლის ნომერი i შეესაბამება $A1$ და $A2$ სამისამართო შესასვლელებზე ორობით კოდს. მულტიპლექსორის სქემიდან გამომდინარეობს, რომ

$$Y = D0 \cdot \overline{A2} \cdot \overline{A1} \vee D1 \cdot \overline{A2} \cdot A1 \vee D2 \cdot A2 \cdot \overline{A1} \vee D3 \cdot A2 \cdot A1.$$

ინორმაციული შესასვლელების ზრდა იწვევს მისამართის თანრიგების ზრდას.





ნახ. 3.16.

3.3.3. მთვლელები.

მთვლელი ეწოდება მიმდევრობითი მოქმედების ოპერაციულ ელემენტს, რომელიც ახორციელებს მის შესასვლელზე მიწოდებული იმპულსების თვლას. თვლის შედეგი ინახება მთვლელში შემდეგი იმპულსის მოსვლამდე. თვლის შედეგების წაკითხვა შესაძლებელია მთვლელ იმპულსებს შორის შუალედში.

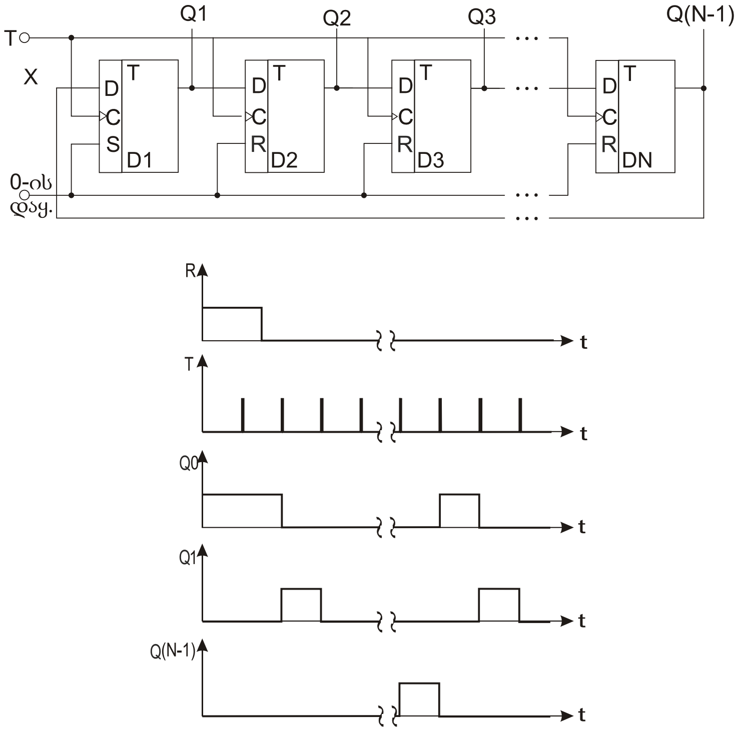
მთვლელები, ისევე როგორც ძვრის რეგისტრები, შედგებიან მიმდევრობით ჩართულ ტრიგერთა მწკრივიდან. მთვლელის თანრიგიანობა და, აქედან გამომდინარე, ტრიგერთა N რაოდენობა, განისაზღვრება იმ მაქსიმალური რიცხვით, სანამდეც უნდა დაითვალოს მთვლელმა. ამ რიცხვს თვლის კოეფიციენტი - $k_{მთ}$ ჰქვია. თუ შესასვლელ იმპულსთა რაოდენობა $n > k_{მთ}$, მაშინ ყოველი $k_{მთ}$ იმპულსის შემდეგ მთვლელი ბრუნდება საწყის მდგომარეობაში და იწყებს იმპულსების თავიდან თვლას.

მთვლელთა ტიპების მრავალფეროვნება გამოწვეულია გამოთვლით ტექნიკაში მათი ფართო გამოყენებით. ისინი გამოიყენებიან ბრძანებათა მისამაღთების თანმიმდევრობის შესაქმნელად, შესრულებულ ოპერაციათა ციკლების რიცხვის დასათვლელად, ანალოგიურ-ციფრული გარდამქმნელების კოდის დასახსომებლად და ა. შ. ქვემოთ განხილულია მთვლელთა ყველაზე მეტად გავრცელებული ტიპები.

რგოლური მთვლელი

რგოლური მთვლელი შეიძლება მივიღოთ ძვრის რეგისტრიდან, თუ ბოლო ტრიგერის გამოსასვლელს შევაერთებთ პირველის D -შესასვლელთან. ასეთი N თანრიგიანი მთვლელის სქემა მოყვანილია ნახ. 3.17-ზე. თვლის დაწყებამდე, საწყისი დაყენების იმპულსით, მთვლელის ნულოვან თანრიგში ($Q0$) იწერება ლოგიკური 1,

ხოლო დანარჩენ თანრიგებში - ლოგიკური 0. თვლის დაწყების შემდეგ თითოეული მთვლელი T იმპულსი გადაწერს 1-ს შემდეგ ტრიგერში, ხოლო მიწოდებულ იმპულსთა რაოდენობა განისაზღვრება იმ გამოსასვლელის ნომრით, რომელშიც არის კოდი „1“. ბოლოს წინა $(n-1)$ -იმპულსი გადაიყვანს ერთის მდგომარეობაში ბოლო ტრიგერს, ხოლო n -ური იმპულსი გადაწერს ერთეულოვან მდგომარეობას ნულოვანი ტრიგერის გამოსასვლელზე, და თვლა თავიდან დაიწყება. ამრიგად, შეგვიძლია ავაგოთ რგოლური მთვლელი ნებისმიერი თვლის $k_{მთ}$ კოეფიციენტით, მხოლოდ ტრიგერთა N რაოდენობის შეცვლით მწკრივში, ვინაიდან $k_{მთ} = N$.



ნახ. 3.17.

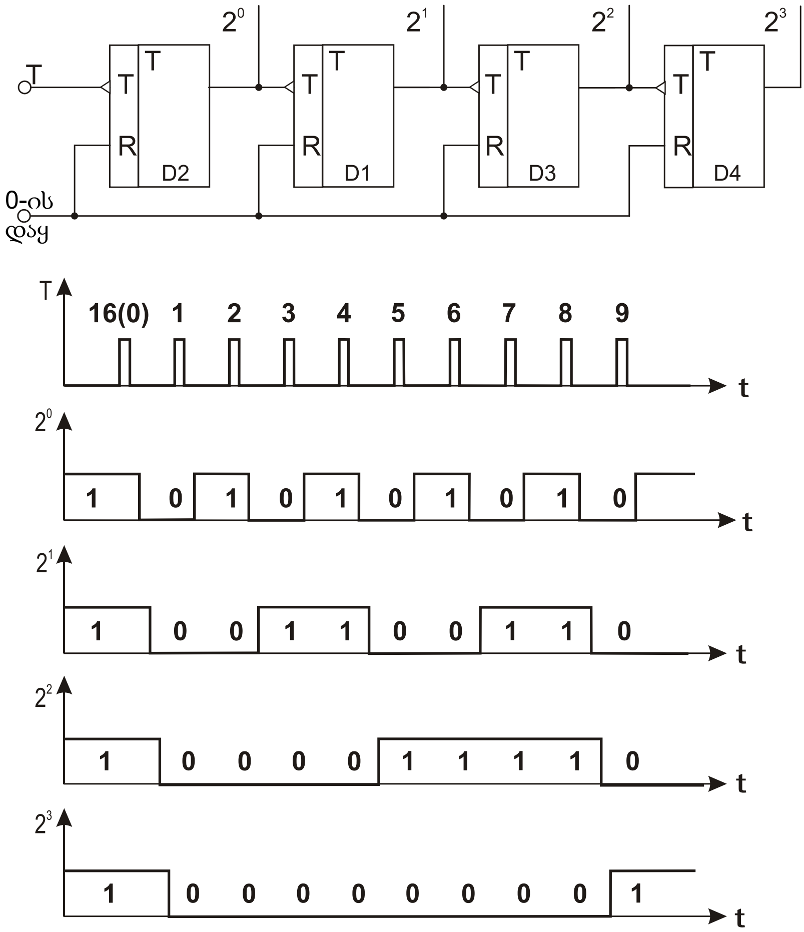
რგოლური მთვლელების ძირითადი გამოყენების არეა იმპულსთა გამანაწილებლები, რომლებიც ქმნიან მართვად სიგნალთა აუცილებელ დროით თანმიმდევრობას. სხვა შემთხვევებში ძირითადად გამოიყენებიან მთვლელ ტრიგერებზე აგებული მთვლელები, რადგანაც ისინი საშუალებას იძლევიან მივიღოთ თვლის საჭირო კოეფიციენტი ტრიგერთა მნიშვნელოვნად მცირე რაოდენობის დროს.

ასინქრონული (მიმდევრობითი) ორობითი მთვლელი.

ეს მთველი აგებულია მიმდევრობით შეერთებული მთვლელი ტრიგერებისაგან. თვლის შედეგი გამოისახება $Q(N-1),\dots,Q0$ გამოსასვლელებზე პარალელული ორობითი კოდის სახით, რომელიც შეესაბამება დათვლილ იმპულსებს. ვინაიდან გამოსასვლელ ცვლადთა რაოდენობა ტოლია ტრიგერთა N რიცხვის და ყოველი ცვლადის მხოლოდ ორ მნიშვნელობა, ამიტომ შესაძლო მდგომარეობათა რიცხვი (თვლის კოეფიციენტი) ტოლია:$k_{მთ}=2^N$. რადგანაც 2^N მდგომარეობიდან ერთი მოდის ნულოვან მდგომარეობაზე, ამიტომ ის მაქსიმალური რიცხვი, რომლის დროსაც მთვლელი მთლიანად შეივსება ერთიანებით, ტოლია (2^N-1).

უმარტივესი ერთთანრიგიანი მთვლელი $k_{მთ}=2$-ით, განხილული ადრე, იცვლის თავის მდგომარეობას საწინააღმდეგოზე ყოველი შესასვლელი სიგნალის მოსვლისას. ამის გამო, ტრიგერის გამოსასვლელზე ძაბვის გადავარდნებს ორჯერ უფრო ნაკლები სიხშირე აქვთ, ვიდრე შესასვლელზე. ამ ვარდნებით ხდება შემდეგ ტრიგერთა გაშვება, და მის გამოსასვლელზე მდგომარეობის ცვლილება ხდება უკვე 4-ჯერ უფრო ნაკლებად, ვიდრე პირველი ტრიგერის შესასვლელზე.

ნახ. 3.18-ზე მოყვანილია T-ტრიგერებზე აგებული ოთხთანრიგიანი ორობითი მთვლელი, რომელიც ირთვება შესასვლელი სიგ-



ნახ. 3.18.

ნალის უკანა ფრონტით და მუშაობის აღმწერი დიაგრამები. დიაგრამები იწყებიან იმ მომენტიდან, როცა მთვლელი მთლიანად შევსებუ-

ლია, ანუ ყველა მის გამოსასვლელზე იმყოფებოდნენ ერთეულები. მთვლელის მიერ დათვლილ იმპულსთა რაოდენობა ტოლია:

$$1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 15$$

რაც შეესაბამება ოთხთანრიგიანი მთვლელის ბოლო მდგომარეობას $(2^4 - 1)$. მე-16 იმპულსის უკანა ფრონტით მიმდევრობით შეიცვლიან მდგომარეობას ყველა ტრიგერები და მთვლელი გადავა საწყის ნულოვან მდგომარეობაში. ამიტომ მე-16 იმპულს ასევე ეძახიან ნულოვანს. ყოველი შემდგომი იმპულსის მოსვლით მთვლელის გამოსასვლელის პარალელური ორობითი კოდი ერთით იწყებს ზრდას, სანამ ისევ არ მოხდება მთვლელის გადავსება, რომლის დროსაც ტრიგერები ნულის მდგომარეობაში გადგებიან.

ნახ. 3.18-ის დიაგრამები გამოსახავენ გამოსასვლელ სიგნალებს თითოეულ ტრიგერში მათი დაყოვნების გათვალისწინებით. კარგად ჩანს, რომ მთვლელის გამოსასვლელზე ჭეშმარიტი ინფორმაცია დგება მხოლოდ ტაქტური იმპულსის უკანა ფრონტის შემდეგ $N \cdot t^{1.0}_{დაყ,თან}$ - გავლილი დროის შემდეგ. აქ $t^{1.0}_{დაყ,თან}$ იმპულსის თითოეულ ტრიგერში სიგნალის გავრცელების დაყოვნებაა. თანრიგიანობის შემდგომი ზრდისას ჯამურმა დაყოვნებამ შეიძლება მიგვიყვანოს მთვლელში ინფორმაციის დამახინჯებასთან. ამიტომ ტრიგერიდან ტრიგერში მთვლელი იმპულსების მიმდევრობითი გადატანით მომუშავე მრავალტანრიგიან მთვლელებს შეუზლიატ იმუშაონ მხოლოდ დაბალ სიხშირეებზე, იმპულსებს შორის დიდი დაყოვნების დროის დროს.

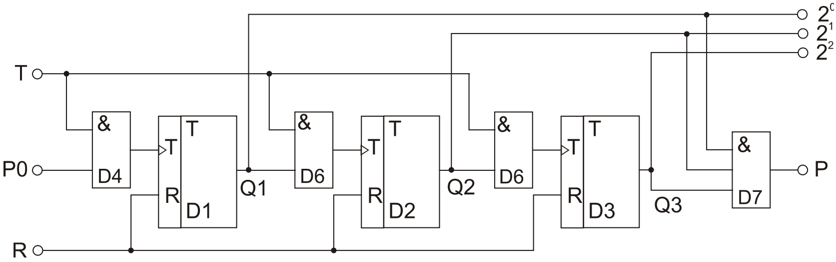
ზემოთ განხილული ამჯამავი მთვლელის გარდა არის გამოკლების პრინციპით მომუშავე მთვლელებიც, რომლებშიც გამოსასვლელი კოდი მცირდება ერთით ყოველი მთვლელი იმპულსის მისვლისას. ასეთი მთვლელი მიიღება ინვერსიული სიგნალების მიწოდებით ტაქტურ შესასვლელებზე. ამისათვის აუცილებელია ტრიგერთა ტაქტური შესასვლელები მივუერთოთ წინა ტრიგერთა ინვერსიულ $\bar{Q}$ გამოსასვლელებს. თუ ტრიგერის შემადგენლობაში შე-

ვიტანთ მულტიპლექსორს, რომელიც გადართავს ტრიგერის ტაქტურ შესასვლელებს წინა ტრიგერთა პირდეპირ ან ინვერსიულ გამოსასვლელებს, მივიღებთ რევერსიულ მთვლელს თვლის მიმართულების ცვლილებით.

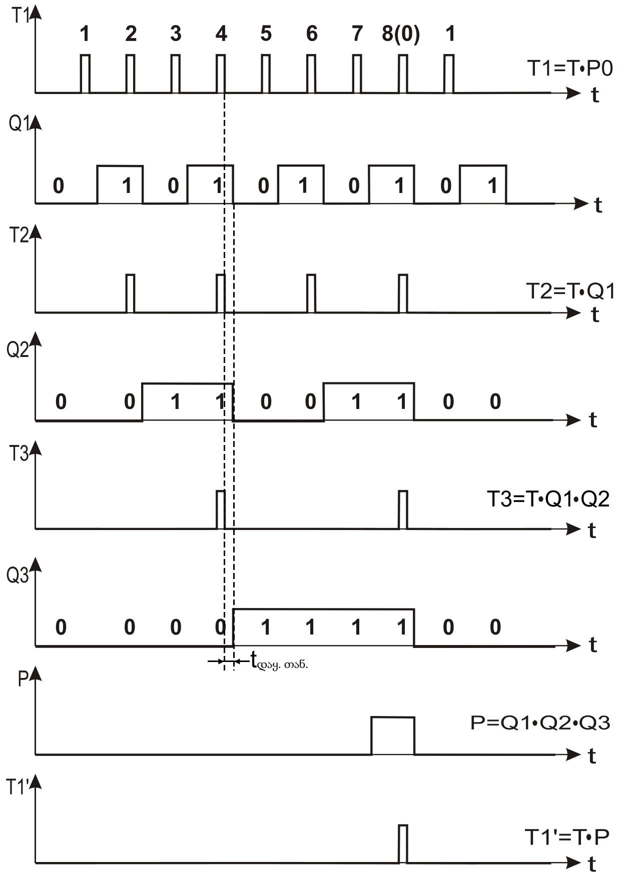
სინქრონული (პარალელური) ორობითი მთვლელი.

ამ მთვლელს უფრო მაღალი სწრაფქმედება გააჩნია იმის გამო, რომ ტაქტური იმპულსები ერთდროულად მიეწოდებიან მთვლელის ყველა ტრიგერის შესასვლელს. განვიხილოთ სამთანრიგა ორობითი სინქრონული მთვლელისმუშაობა, რომლის სქემა მოყვანილია ნახ. 3.19,ა-ზე, ხოლო დროითი დიაგრამები - ნახ. 3.19,ბ-ზე.

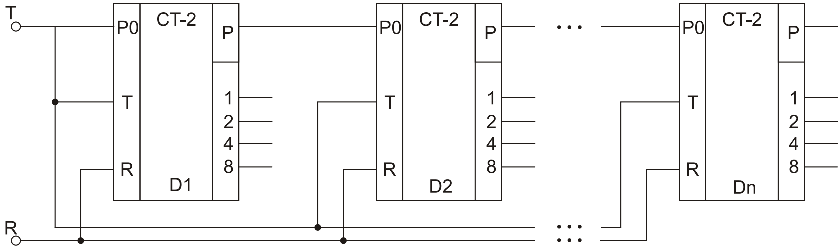
დიაგრამებზე მთვლელის სამი თანრიგის მთვლელი შესასვლელები და პირდაპირი გამოსასვლელები შესაბამისად აღნიშნულნი არიან $T1, T2, T3$ და $Q1, Q2, Q3$ - ებით. სამუშაოს დაწყების წინ R შესასვლელზე ერთეულოვანი სიგნალის მიწოდებით ნულოვანი მდგომარეობა დგება ყველა გამოსასვლელზე. ერთმანეთს შორის შეერთებულ T და $P0$ შესასვლელებზე (როგორც პირველ მიკრო სქემაზე - ნახ. 3.19,გ) მიეწოდება მთვლელი იმპულსები. ვინაიდან თავდაპირველად ყველა ტრიგერი იმყოფებოდა ნულოვან მდგომარეობაში, ამიტომ პირველი იმპულსი გავა მხოლოდ პირველი ტრიგერის შესასვლელზე და გადაიყვანს რა $D1$ ტრიგერს ერთის მდგომარეობაში, მოამზადებს $D5$ ელემენტს (სქემა „და“) მეორე იმპულსის გადასაცე-


T
P0
R
&
D4
D6
D7
T
R
D1
D2
D3
Q1
Q2
Q3
P
2^0
2^1
2^2

ა)


T1
1 2 3 4 5 6 7 8(0) 1
T1=T•P0
Q1
0 1 0 1 0 1 0 1 0 1
T2
T2=T•Q1
Q2
0 0 1 1 0 0 1 1 0 0
T3
T3=T•Q1•Q2
Q3
0 0 0 0 1 1 1 1 0 0
$t_{დაყ. თან.}$
P
P=Q1•Q2•Q3
T1'
T1'=T•P
t

ბ)



გ)

ნახ. 3.19.

მად $D2$ -ის სათვლელ შესასვლელზე. ყველა სხვა იმპულსის გავლა ადვილი წარმოსადგენია ლოგიკური ფუნქციების გათვალისწინებით, რომლებიც აგებულებია $D4$, $D5$ და $D6$ სქემებზე:

$$T1 = T \cdot P0 = T; \quad T2 = T \cdot Q1; \quad T3 = T \cdot Q1 \cdot Q2.$$

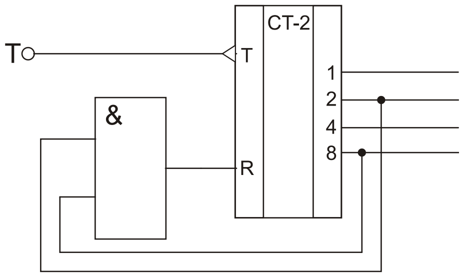
დიაგრამა გვიჩვენებს, რომ ტრიგერებში ერთნაირი დაყოვნებისას ინფორმაციის შეცვლა მთვლელის ყველა თანრიგში ხორციელდება ერთდროულად.

პარალელური მთვლელის თანრიგების შემდგომი ზრდის დროს ჩნდება აუცილებლობა შესასვლელთა დიდი რაოდენობის მქონე „და“ ელემენტებზე. ამიტომ ჩვეულებრივ ინტეგრალურ შესრულებაში არსებობენ ოთხთანრიგიანი მთვლელები, რომელთა პირობით-გრაფიკული აღნიშვნა მოყვანილია ნახ. 3.19,გ-ზე. მრავალთანრიგიანი მთვლელის დასამზადებლად აერთებენ რამდენიმე მიკროსქემას და აწვდიან გადატანის გამოსასვლელიდან P სიგნალის გადატანის ნებართვის დართვის $P0$ შესასვლელზე. დროითი დიაგრამიდან ჩანს, რომ გადატანის P სიგნალი საშუალებას იძლევა ტაქტური $T1'$ იმპულსის ფორმირებისათვის შემდეგი მთვლელის პირველი ტრიგერისათვის ყოველგვარი დამატებითი დაყოვნების გარეშე.

მთვლელი თვლის ნებისმიერი კოეფიციენტით.

ხშირად საჭიროა ისეთი მთვლელები, რომლებსაც გააჩნიათ 2^N მდგრადი მდგომარეობისაგან განსხვავებული მდგრადი მდგომარეობები. მაგალითად, ელექტრონულ საათებში არიან მიკროსქემები თვლის კოეფიციენტით 6 (წუთის მეათედები), 10 (წუთის ერთეულები), 7 (კვირის დღეები), 24 (საათები). $k_{თვ} \neq 2^N$ მთვლელის ასაგებად, საჭიროა გამოვიყენოთ მოწყობილობა აგებული N-ტრიგერისაგან, რომლისთვისაც სრულდება პირობა $2^N > k_{თვ}$. ცხადია, ასეთ მთვლელს გააჩნია ზედმეტი მდგრადი მდგომარეობები ($2^N - k_{თვ}$). ამ არასაჭირო მდგომარეობათა გამორიცხვა შეიძლება უკუ კავშირების გამოყენებით, რომელთა წრედებით მთვლელი გადაირთვება ნულის მდგომარეობაში მუშაობის იმ ტაქტზე, რომელშიც ის დაითვლის $k_{თვ}$-მდე.

მთვლელისათვის $k_{თვ} = 10$ -ით საჭიროა ოთხი ტრიგერი (ვინაიდან $2^3 < 10 < 2^4$). მთვლელს უნდა გააჩნდეს ათი (0, ... ,9) მდგრადი მდგომარეობა. იმ ტაქტში, რომელშიც ის უნდა გადასულიყო მეთერთმეტე მდგრად მდგომარეობაში (რიცხვი 10), საჭიროა ის დაყენებულ იქნას საწყის ნულოვან მდგომარეობაში. ასეთი მთვლელისათვის შეიძლება გამოვიყენოთ ნებისმიერი ოთხთანრიგა მთვლელი (ნახ.3.20) უკუ კავშირის წრედებით გამოსასვლელებიდან,



ნახ. 3.20.

რომლებიც შეესაბამებიან რიცხვს 10 (ანუ 2 და 8-დან), მთვლელის ნულზე დაყენების შესასვლელზე (შესასვლელი R).მეთერთმეტე მდგომარეობის დასაწყისში (რიცხვი 10) „და“ ეთობითს, ხოლო მის გამოსასვლელ კოდს - ოროზით-ათობით კოდს (ან კოდს 8421).

მთვლელი წინასწარი დაყენებით.

ეს მთვლელიშეიძლება დადგეს საწყის მდგომარეობაში, რომელიც ტოლია ნებისმიერი ციფრისა 0-დან $k_{თვ} - 1$ - მდე. ეს ოპერაცია ხორციელდება საჭირო რიცხვის კოდის მთვლელში პარალელური ჩაწერით. თვლა (შეკრება ან გამოკლება) უკვე დაიწყება არა ნულიდან, არამედ დაყენებული რიცხვიდან. მთვლელის მუშაობის ასეთი რეჟიმი აუცილებელია, მაგალითად კომპიუტერის მართვის მოწყობილობაში მოცემული საწყისი მისამართიდან ბრძანებათა მიმდევრობითი მისამართების შექმნისას.

სიხშირის გამყოფები.

როგორც სქემებიდან და დიაგრამებიდან ჩანს (ნახ. 3.17-3.20), მთვლელებს შეუძლიათ სიხშირის გამყოფის ფუნქციის შესრულება, ანუ ისეთი მოწყობილობებისა, რომლებიც $f_{შეს}$ სიხშირის შესასვლელ იმპულსურ თანმიმდევრობას გარდაქმნიან გამოსასვლელ ტრიგერზე $f_{გამ}$ სიხშირეში, რომელიც $k_{თვ}$ -ჯერ ნაკლებია შესასვლელზე. მთვლელის ასეთი გამოყენებისას არაა იმის აუცილებლობა, რომ ვიცოდეთ თუ რა რიცხვია მასში ჩაწერილი მოცემულ მომენტში. ამიტომ გამყოფებს შეიძლება არ ქონდეთ საშუალედო გამოსასვლელები. ეს კი მნიშვნელოვნად ამარტივებს მათ სქემას და კონსტრუქციას.

3.3.4. არითმეტიკულ-ლოგიკური კვანძები.

ამჯამავი.

ამჯამავის დანიშნულებაა ორი რიცხვის არიმმეტიკული შეკრება. ორი მრავალთანრიგიანი ორობითი რიცხვის აჯამვისას ყოველ i -ურ თანრიგში იმყოფება სამი რიცხვის ჯამი ორის მოდული (შესაკრებები A_i , B_i და გადატანა, რომელიც მიეწოდება უმცროსი P_i თანრიგიდან) და ფორმირდება გადატანის სიგნალი უფროს P_{i+1} თანრიგში. ერთთანრიგიანი ამჯამავის ჭეშმარიტების ცხრილი მოყვანილია ცრილ 3.5-ში.

ცხრილი 3.5

შესასვლელი			გამოსასვლელი	
შესაკრებები		გადატანა	ჯამი	გადატანა
A_i	B_i	P_i	S_i	P_{i+1}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

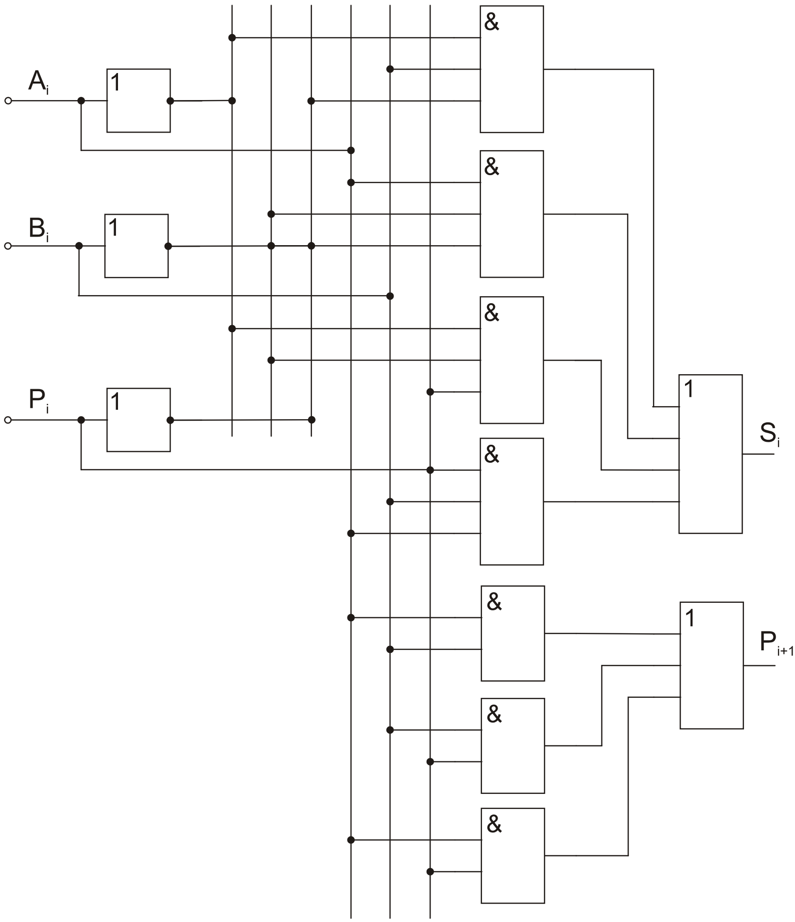
ცხრილის ანალიზის საფუძველზე შეგვიძლია ჩავწეროთ ლოგიკური ფუნქციები გამოსასვლელი სიდიდეებისათვის:

$$S = \bar{A} \cdot B \cdot \bar{P} \vee A \cdot \bar{B} \cdot \bar{P} \vee \bar{A} \cdot \bar{B} \cdot P \vee A \cdot B \cdot P;$$

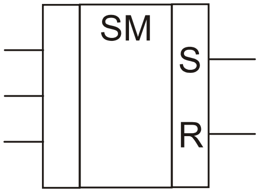
$$P_{i+1} = A \cdot B \cdot \bar{P} \vee \bar{A} \cdot B \cdot P \vee A \cdot \bar{B} \cdot P \vee A \cdot B \cdot P = A \cdot B \vee B \cdot P \vee A \cdot P\,.$$

ამ ფუნქციათა საფუძველზე შეგვიძლია ავაგოთ ამჯამავი, რომელიც შედგება ინვერტორების, „და“ ელემენტებისა და „ან“ ელემენ-

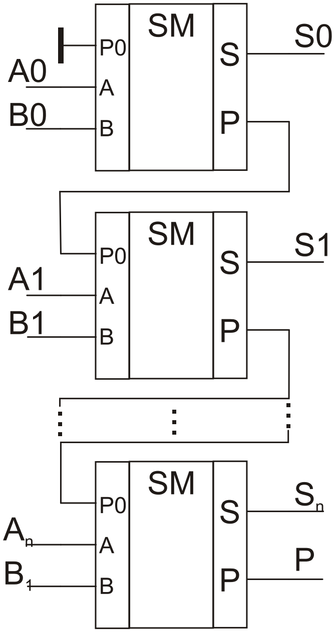
ტებისაგან (ნახ. 3.21,ა). ერთთანრიგიანი ამჯამავის სიმბოლური აღნიშვნა მოყვანილია ნახ. 3.21,ბ-ზე. ორი მრავალთანრიგიანი ორობითი რიცხვის აჯამვისათვის იყენებენ მრავალთანრიგიან ამჯამავებს, რომლებიც უმარტივეს სახეში წარმოადგენენ ერთთანრიგიანი ამჯამავების მიმდევრობით შეერთებას (ნახ. 3.21,გ).



ა)



ბ)



გ)

ნახ. 3.21.

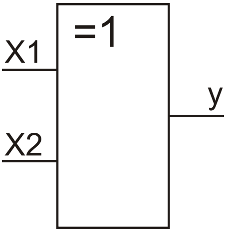
ასეთ ამჯამავში, თითოეული i -ური თანრიგის გამოსასვლელზე შექმნილი გადატანის სიგნალი, წარმოადგენს შემდეგი ერთთანრიგიანი ამჯამავის სამი შესაკრებიდან ერთს. გადატანის ყველა საშუალედო სიგნალი ამჯამავის გარე გამოსასვლელებს არ მიეწოდება.

აქ აჯამვის პროცესი მიმდინარეობს მიმდევრობით (თანრიგების მიხედვით, უმცროსიდან დაწყებული), და სწორი შედეგის მისაღებად i -ურ თანრიგში, ჩვენ თავდაპირველად უნდა ავჯამოთ რიცხვები $(i-1)$ თანრიგში და მისგან მივიღებთ გადატანის P_i მნიშვნელობა. ამიტომ გადატანის სიგნალის მიმდევრობითი ფორმირების მქონე ამჯამავში მისი სწრაფქმედება განისაზღვრება იმ შემთხვევისათვის, რომლის დროსაც A და B რიცხვების აჯამვისას გადატანის ერთიანი წარმოიქმნება მიმდევრობით თითოეულ თანრიგში და, აქედან გამომდინარე, აჯამვის დრო იქნება უდიდესი და ტოლი $T_{დაყ,თან} = n \cdot t_{დაყ,თან}$, სადაც $t_{დაყ,თან}$ - გადატანის ერთიანის გავრცელების დაყოვნებაა ერთ თანრიგში. სწრაფქმედების ასამაღლებლად ქმნიან ამჯამავებს პარალელური გადატანით, რომელთა აგეის პრინციპიც მსგავსია გამჭოლი გადატანის მთვლელების (სინქრონული მთვლელები) აგების პრინციპისა.

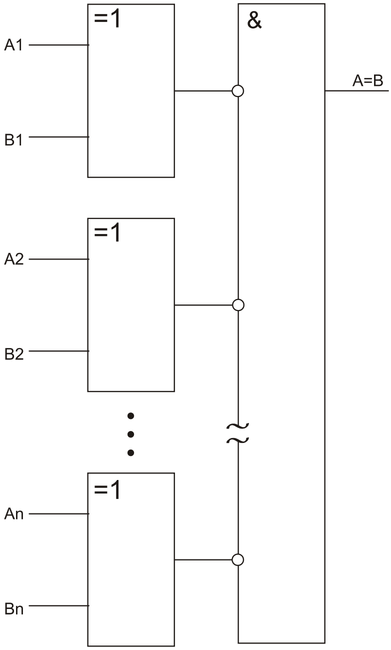
<u>შედარების კვანძი.</u>

შედარების კვანძის დანიშნულებაა ორი A და B მრავალთანრიგიანი ორობითი რიცხვების შედარება. ამ ამოცანის ამოხსნა შეიძლება დაყოფილ იქნას ორ ეტაპად. პირველზე - ადგენენ ორი რიცხვის თითოეული თანრიგის ტოლობას ($A_i = B_i = 1$ ან $A_i = B_i = 0$). ამ ოპერაციის შემსრულებელ ლოგიკურ ელემენტს წარმოადგენს ამჯამავი ორის მოდულით (ნახ. 3.22), რომლისთვისაც სამართლიანია ტოლობა $Y = \bar{A} \cdot B \vee A \cdot \bar{B}$, რომელიც მიუთითებს, რომ გამოსასვლელ Y სიგნალს აქვს დაბალი დონე მხოლოდ $A = B$ -ს დროს. მეორე ეტაპზე ადგენენ აქვს თუ არა ყველა ამჯამავებს ორის მოდულით გამოსასვლელზე სიგნალების დაბალი დონე. ამ ოპერაციას ასრულებს მრავალშესასვლელიანი „და“ ელემენტი დაბალი დონის შესასვლე-

ლი სიგნალებისათვის. ამ ელემენტის გამოსასვლელი სიგნალი ტო-
ლ



ნახ. 3.22.



ნახ. 3.23.

ამრიგად, ციფრულ კომპარატორს (ნახ. 3.23), რომელიც შედგება ორის მოდულით ამჯამავის (n ელემენტიანის) და ერთი n-შესასვლელიანი „და“ ელემენტისაგან (ინვერსიული სიგნალებისთვის), გამოსასვლელზე ექნება დონე $Q = 1$, როცა $A = B$ და $Q = 0$ -ს, როცა $A \neq B$. რიცხვი n უნდა შეესაბამებოდეს A და B რიცხვებს შორის უდიდესის თანრიგს. ინტეგრალური შესრულებით არიან ოპერაციული კვანძები, რომლებიც $A = B$ -ს ოპერაციის გარდა ასრულებენ შედარების სხვა ოპერაციებს $A < B$ და $A > B$.

ლ ი ტ ე რ ა ტ უ რ ა

1. გრ. მამარდაშვილი ელექტრონული ციფრული მანქანების არით-მეტიკული საფუძვლები. თბილისი „ განათლება“, 1982წ.
2. Карцев М. А. Арифметика цифровых машин. –М.: Наука, 1969.
3. Савельев А.Я. Арифметические и логические основы цифровых автоматов: Учебник.-М.: Высш. школа, 1980.
4. Акимов О. Е. Дискретная математика: логика, группы, графи. 2-е изд. дополн.- М.: Лаборатория базовых знаный, 2001.
5. Шевелев Ю. П. Дискретная математика. Ч. 1: Теория множеств. Булева алгебра. – Томск. Гос. ун-т систем управления и радио-электроники, 2003.
6. Шевелев Ю. П. Дискретная математика. Ч. 2: Теория конечных автоматов. Комбинаторика. Теория графов. – Томск. Гос. ун-т систем управления и радиоэлектроники, 2003.
7. გ. გოგიჩაიშვილი, გ. ჩაჩანიძე, ქ. ნანობაშვილი. ავტომატიზებული მართვის მოდელები. - თბილისი, გამომცემლობა-სტუ, 2005.
8. Гивоне Д., Россер Р. Микропроцессоры и микрокомпьютеры: Пер. с англ. – М.: Мир, 1983.
9. Галушкина Ю. И., Марьямов А. Н. Конспект лекций по дискретной математике. – М.: Айрис-пресс, 2007.
10. Лысиков Б.Г. Арифметические и логические основы цифровых автоматов: Учебник для вузов. По спец. “Электрон. вычисл. машины”.- 2-е изд.-Мн.: Высш. Школа, 1980.
11. Каган Б.М. Электронные вычислительные машины. - М.: Энерго-атомиздат, 1991.
12. Ямпольский В.С. Основы автоматики и элелтронно-вычислительной техники.-М.:Просвещение, 1991.

ს ა რ ჩ ე ვ ი

კომპიუტერის არითმეტიკული და ლოგიკური საფუძვლები

ავტორი:

თენგიზ ხინიკაძე - ტექნიკურ მეცნიერებათა კანდიდატი, ბათუმის შოთა რუსთაველის სახელმწიფო უნივერსიტეტის განათლებისა და მეცნიერებათა ფაკულტეტის კომპიუტერულ მეცნიერებათა დეპარტამენტის ასოცირებული პროფესორი