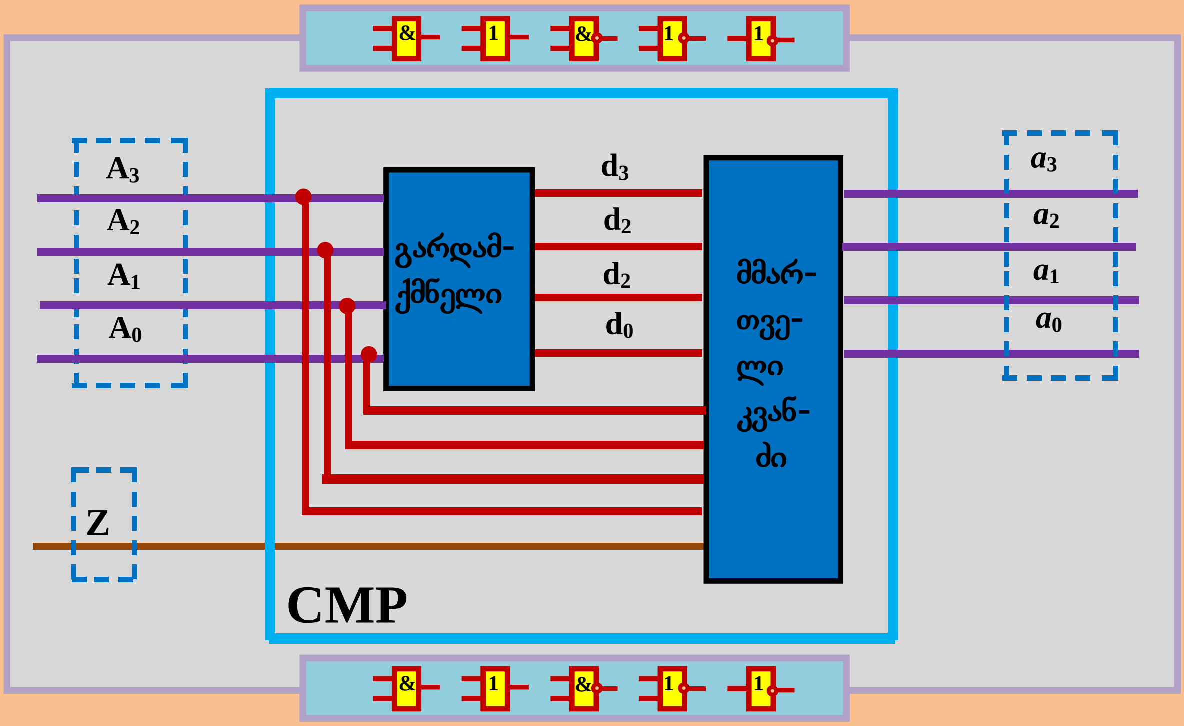
ალექსანდრე დუნდუა

კომპიუტერული სისტემებისა და საინფორმაციო ტექნოლოგიების თეორიული საფუძვლები



“ტექნიკური უნივერსიტეტი”

ნაშრომში განსაზღვრულია ინფორმაციის რაობა, სტრუქტურული ერთეულები და განხილულია მისი კლასიფიცირება; ნაჩვენებია თვლის სისტემების, სათვლელი მოწყობილობებისა და კომპიუტერების წარმოშობის წანამძღვრები და მათი განვითარების გზები; განხილულია ავტომატიზებული საინფორმაციო სისტემებისა და სამუშაო ადგილების რაობა, მათი ორგანიზებისა და კლასიფიცირების საკითხები; გადმოცემულია კომპიუტერში მონაცემების წარმოდგენის მეთოდები და ფორმულირებულია თვლის სხვადასხვა სისტემების საშუალებით გამოსახულ რიცხვებზე არითმეტიკული ოპერაციების შესრულების ხერხები; დიდი ადგილი აქვს დათმობილი ლოგიკის ალგებრის რაობისა და მეთოდების გადმოცემას; განხილულია პროცესორული მოწყობილობების ფუნქციონირების საკითხები და მოყვანილია კომპიუტერული სისტემის აპარატურული უზრუნველყოფის ცალკეული მოწყობილობების ასაგებად ლოგიკის ალგებრის გამოყენების მაგალითები.

დამხმარე სახელმძღვანელო განკუთვნილია ტექნიკური უმაღლესი სასწავლო დაწესებულების სტუდენტებისათვის, რომლებიც სწავლობენ დისციპლინას “”კომპიუტერული სისტემები და გამოყენებითი ტექნოლოგიები” ან “ინფორმატიკა”. სახელმძღვანელო დიდ დახმარებას გაუწევს როგორც ამ სფეროში მომუშავე პედაგოგებსა და სპეციალისტებს, ისე ამ საკითხით დაინტერესებულ ფართო საზოგადოებას.

რეცენზენტები: სრული პროფესორი კ. კამკამიძე,
სრული პროფესორი მ. გოცაძე

ISBN 978-9941-20-461-6
http://www.gtu.ge/publishinghouse/

Verba volant, scripta manent

(ვალერი საპოჟნიკოვი,
Валерий В.Сапожников)

დიდი პატივისცემითა და სიყვარულით ჩემს მასწავლებლებს ,"მიმოსვლის გზათა პეტერბურგის სახელმწიფო უნივერსიტეტის" პროფესორებს – ვალერი ვლადიმერის ძე საპოჟნიკოვსა და ვლადიმერ ვლადიმერის ძე საპოჟნიკოვს, მათთან გატარებული ძალზე ნაყოფიერი და საინტერესო წლებისათვის !

(ვლადიმერ საპოჟნიკოვი,
Владимир В.Сапожников)

С большим уважением и любовью моим учителям, профессорам «Петербургского государственного университета путей сообщения», Валерию Владимировичу Сапожникову и Владимиру Владимировичу Сапожникову, годы работы под руководством которых были наиболее плодотворными и интересными в моей жизни!

წ ი ნ ა ს ი ტ ყ ვ ა ო ბ ა

"*ზედაპირული ცოდნა სიამოვნებისა და ღირსების ნაცვლად გარცხვენს, ან უბრალოდ, სასაცილო მდგომარეობაში გაყენებს.*"
ფ. ჩესტერფილდი (1694-1773)

ადამიანი, როგორც გონიერი არსება (ლათ - **Homo Sapiens***), უძველესი დროიდანვე ცდილობდა საკუთარი როგორც ფიზიკური, ასევე გონებრივი საქმიანობის შემსუბუქებასა და მისი მწარმოებლურობის ამაღლებას.*

პირველი ამოცანის გადასაწყვეტად მან თავდაპირველად დაიწყო სხვადასხვა სახის პრიმიტიული ***სამუშაო იარაღების*** *(წამახვილებული ქვისა და ძვლის იარაღების, ბადეებ-ის, ბერკეტებისა და ა.შ.) დამზადება, ხოლო მეორე ამოცანის გადასაწყვეტად – ასევე პრიმიტიული* ***სათვლელი საშუალებების*** *(ძველ საბერძნეთში - "აბაკების", ჩინეთში - "სუან-პანების", იაპონიაში – "სერობიანების", ეგეოსის ზღვაში არსებულ კუნძულ სალამინში - "სალამინური დაფებისა" [37] და ა.შ.) გამოგონება.*

ზემოთ აღნიშნული ორივე მიმართულებით საქმიანობა ადამიანს დღემდე არ შეუწყვეტია და მოღვაწეობის ორივე სფეროში იგი პერმანენტულად აღწევს მნიშვნელოვან წარმატებებს. ფიზიკურ შრომასთან დაკავშირებული პრობლემების გადაწყვეტის პროცესში საუკუნეების განმავლობაში მიღწეული წარმატებების რეალიზების შედეგია ჩვენ გარშემო დღეს არსებული ***სუპერთანამედროვე საწარმოები,*** *ხოლო გონებრივი შრომის პრობლემების გადაწყვეტის პროცესში მიღწეული წარმატებების შედეგად,* ***XX*** *საუკუნის შუა პე-*

რიოდში დამუშავებული იქნა ***კომპიუტერული სისტემები.*** *პირველი მათგანი თუ მაღალ-მწარმოებლური* ***მატერიალური ტექნოლოგიების*** *გამოყენების საშუალებით* ***მატერიალური პროდუქტების*** *დამზადების საშუალებას იძლევა, მეორე მათგანმა მატერიალურ პროდუქტებზე არანაკლები მნიშვნელობის მქონე* ***საინფორმაციო პროდუქტების*** *დასამზადებლად საჭირო* ***საინფორმაციო ტექნოლოგიებს*** *ჩაუყარა საფუძველი და უმოკლეს დროში ისინი არნახულ სიმაღლეებზე აიყვანა.*

ზემოთ აღწერილი რელობა წარმოადგენს ფილოსოფიური მიზეზ-შედეგობრივი კატეგორიის კიდევ ერთ გამოვლინებას, რომლის ძალითაც ამქვეყნად უმიზეზოდ არაფერი არ ხდება და რომელიც ცნობილმა ინგლისელმა კლასიკოსმა ***უილიამ შექსპირმა (1564- 1616)*** *თავის უკვდავ "მეფე ლირში" მხატვრულად ასე გამოხატა:* ***"არარაისგან არ იქნება არარაიცა"*** *(მთარგმნელები* ***ივანე მაჩაბელი*** *და* ***ილია ჭავჭავაძე****).*

კომპიუტერულ სისტემებსა და საინფორმაციო ტექნოლოგიებს შეისწავლის საბუნებისმეტყველო მეცნიერება, რომელიც ამერიკაში ცნობილია ***"კომპიუტერული მეცნიერების"*** *სახელწოდებით* ***("Computer Science")****, ხოლო ჩვენთან და ევროპაში –* ***"ინფორმატიკის"*** *სახელწოდებით. მასში განიხილება კომპიუტერული სისტემების:*

- *თეორიული საფუძვლები;*
- *ტექნიკური საშუალებები (****HARDWARE****);*
- *პროგრამული საშუალებები (****SOFTWARE****);*
- *ალგორითმული საშუალებები (****BRAINWARE****);*
- *ქსელებად გაერთიანებისა და მათში დასამუშავებელი ინფორმაციის დაცვის საკითხები*

ჩვენი მიზანია ქართველი სტუდენტებისათვის ეტაპობრივად მოვამზადოთ ჩამოთვლილი ნაწილების შესაბამისი დამხმარე სახელმძღვანელოები. მოცემული წიგნი წარმოადგენს ამ მიმართულებით გადადგმულ პირველ ნაბიჯს, რომელიც ეძღვნება კომპიუტერული სისტემებისა და საინფორმაციო ტექნოლოგიების ***თეორიული საფუძვლების*** *საკვანძო საკითხების განხილვას.*

ავსტრიელი ცნობილი ფიზიკოსის ***ლუდვიგ ბოლცმანის*** *(****1844- 1906*** ***წ.წ.****) თანახმად* ***"კარგ თეორიაზე უფრო პრაქტიკული არაფერი არის"****, მაგრამ მისი პრაქტიკული ხიბლი ცივი ფორმულების ყრუ კედლითაა დაფარული. როგორც მეფისტოფელი გვიმტკიცებს,* ***"ულიმღამოა, ძვირფასო მეგობარო, ყველა თეორია, ხოლო მწვანეა ოქროვანი ხე ცხოვრებისა" (Grau, teurer freund, ist alle theorie und grün des lebens goldner baum.*** *იგოეთე, ფაუსტი,* ***I*** *მოქმედება,* ***IV*** *სცენა). ამ ულიმღამობის ნაწილობრივ მაინც გასაფანტავად ვცდილობდი თითოეული განსახილველი საკითხი ისეთი ფაქტებით შემევსო, რომლებიც მას საინტერესოს გახდიდა და, ამასთანავე, მკითხველს თვალსაწიერის გაფართოებაში დაეხმარებოდა. მოხარული ვიქნები ჩემი მცდელობა ზოგიერთი მკითხველის მოწონებას მაინც თუ დაიმსახურებს.*

მეცნიერულად დამტკიცებულია, რომ ადამიანის მიერ ***გრძნობის ორგანოებით აღქმულ მთელ ინფორმაციაში 80% თვალის მიერ აღქმულ ინფორმაციას უკავია.*** *ამიტომ წიგნზე მუშაობისას დიდი დრო დავუთმე ისეთი სპეციალური გრაფიკული ფორმების დამუშავებას, რომელიც, ჩემი თვალსაზრისით, მკითხველს გადმოცემული მასალის აღქმას გაუადვილებს. მოხარული ვიქნები მიღებული შედეგი თუ მკითხველის მოწონებას დაიმსახურებს.*

"ისტორია - ცხოვრების მასწავლებელია" *გვმოძღვრავდა ძველი რომაელი მოაზროვნე* ***ციცერონი (ჩვენს წელთაღრიცხვამდე 106-43 წ.წ)****, ხოლო ესპანელი გენიოსის* ***სერვანტესის (1547-1616 წ.წ)*** *აზრით* ***"ისტორია ჩვენი საქმიანობის საგანძური, წარსულის მოწმე, აწმყოში მაგალითის მომცემი და ცხოვრების მასწავლებელი, მომავალში კი – მოსალოდნელი საშიშროების შესახებ ჩვენი მაფრთხილებელია".*** *ამიტომ წიგნზე მუშაობისას ვცდილობდი განუხრელად დამეცვა* ***ისტორიზმის პრინციპი.***

მეცნიერებას ტიტანური შრომით ქმნიან და ანვითარებენ კონკრეტული ადამიანები, რომელთა სახეები ახალგაზრდებისათვის უცნობი არ უნდა იყოს; ამიტომაც ვეცადე ნაშრომი გამოჩენილ მეცნიერთა სურათებით გამემდიდრებინა.

დამხმარე სახელმძღვანელო მასში წარმოდგენილი საკითხების იმაზე უფრო ღრმად განხილვის საშუალებას იძლევა, ვიდრე ეს პროგრამით მოეთხოვება სტუდენტს. ვისარგებლე რა ამ შესაძლებლობით, **შევეცადე მეჩვენებინა არა მარტო ამა თუ იმ პრობლემის გადაწყვეტის შედეგი, არამედ თავად ამ გადაწყვეტის პროცესიც. იმედი მაქვს, რომ დაინტერესებულ პირს ეს დაეხმარება ანალიტიკური აზროვნების სრულყოფაში.**

I თავში განმარტებულია ინფორმაციის რაობა და გადმოცემულია გამოთვლითი ტექნიკის განვითარების მოკლე ისტორია; **II თავში** განხილულია თვლის სისტემებთან დაკავშირებული საკითხები, ნაჩვენებია რიცხვების საშუალებით ბგერითი, სიმბოლური და გრაფიკული ინფორმაციის წარმოდგენის გზები და გადმოცემულია ინფორმაციის რაოდენობრივი შეფასების პრობლემები; **III თავში** განხილულია ე.წ. "კომპიუტერული არითმეტიკის" საკითხები; **IV თავში** ჩამოყალიბებულია ინფორმატიკის ლოგიკური საფუძვლები, ხოლო ბოლოს, **V თავში** მოყვანილია ლოგიკის ალგებრის პრაქტიკაში გამოყენების მაგალითები. ზემოთ განხილული ზოგიერთი სხვა საკითხი, რამდენადმე სცილდება წმინდა თეორიულ სფეროს, მაგრამ მიგვაჩნია, რომ ასეთი "გადახრა" სასარგებლო და აუცილებელია თეორიულსა და პრაქტიკულ საკითხებს შორის არსებული ურღვევი კავშირის ილუსტრირებისა და მათი ერთი მთლიანობის სახით წარმოდგენისათვის. **ელემენტალურ ლოგიკურ ფუნქციათა ფუნქციონალურად სრული სისტემები** მნიშვნელოვანია არა მარტო წმინდა თეორიული თვალსაზრისით, არამედ მათი უდიდესი პრაქტიკული ღირებულების გამოც, რადგან სწორედ ასეთ სისტემებში შემავალი ფუნქციების მარეალიზებული **ლოგიკური ელემენტებითაა** შესაძლებელი ნებისმიერი კომპიუტერული სისტემის აპარატურული უზრუნველყოფის რეალიზება; სხვა სიტყვებით რომ ვთქვათ, ამ ელემენტების (ანუ **ვენტილების**) ერთობლიობით წარმოიქმნება კომპიუტერული უზრუნველყოფის ასაგებად საჭირო "სამშენებლო მასალის" სრული ნომენკლატურა. ასევე, **ლოგიკის ალგებრა**, რომელიც დიდი ხნის განმავლობაში პრაქტიკულ ღირებულებას მოკლებულ წმინდა თეორიულ მოძღვრებად ითვლებოდა, თურმე კომპიუტერული უზრუნველყოფის მოწყობილობების ისეთი მათემატიკური მოდელია, რომელიც ამ მოწყობილობების შექმნამდე დიდი ხნით ადრე იქნა დამუშავებული. მისი გამოყენებითაა შესაძლებელი აღნიშნული მოწყობილობების ანალიზისა და სინთეზის ამოცანების ფორმალიზებული გადაწყვეტა. განხილული და მათი მსგავსი მრავალი მაგალითიდან გამომდინარეობს, რომ **თეორია იქმნება პრაქტიკისათვის და პრაქტიკა თეორიის გამატერიალიზების პრო-ცესია.**

დიდი მცდელობის მიუხედავად, მოცემული წიგნი, ალბათ მაინც არ იქნება თავისუფალი ცალკეული უზუსტობებისა თუ შეცდომებისაგან, რისთვისაც წინასწარ ვიხდი ბოდიშს; მისი გაუმჯობესებისაკენ მიმართულ თითოეულ შენიშვნასა თუ რჩევას დიდი სიამოვნებით მივიღებ.

I თ ა ვ ი

ინფორმატიზაციის ზოგადი საკითხები

1.1. ინფორმაციის რაობა

"ვინც ფლობს ინფორმაციას, იგი ფლობს სამყაროს"

***ნათან-მაიერ როტშილდი** (1777—1836)*

1 **კომპიუტერი** (ინგ. **Computer** – "გამომთვლელი"), მრავალმნიშვნელოვანი ტერმინია, რომელსაც ყველაზე ხშირად **ინფორმაციის** დამამუშავებელი პროგრამულად მართვადი ელექტრონული მოწყობილობის აღსანიშნავად ვიყენებთ.

ზემოთ მოყვანილ განსაზღვრებაში გამოყენებული ტერმინი **"ინფორმაცია"** მიეკუთვნება ერთ-ერთ უმნიშვნელოვანეს ტერმინთა ჯგუფს, რადგან ადამიანის მთელი ცხოვრება ამა თუ იმ სახით დაკავშირებულია როგორც გარე, ისე საკუთარი შინაგანი სამყაროდან მიღებული ინფორმაციის აღქმას, დაგროვებასა და გადამუშავებასთან. როგორც სამეცნიერო კატეგორია, "ინფორმაცია" ერთმანეთისაგან განსხვავებული უამრავი დისციპლინის (ფიზიკის, ბიოლოგიის, ფილოსოფიის, მათემატიკისა და ა.შ.) შესწავლის საგანია; აღნიშნულის მიუხედავად მისი ზუსტი მეცნიერული **განსაზღვრება** დღემდე არ არსებობს და **განსაზღვრების** ნაცვლად გამოიყენება **ცნება** ინფორმაციის შესახებ. განსაზღვრებისაგან ცნება იმით განსახვავდება, რომ მეცნიერებისა და ტექნიკის სხვადასხვა სფეროში ცნების ფორმულირებას ისე ახდენენ, რომ მისი შინაარსი მაქსიმალურად შეესაბამებოდეს კონკრეტული დისციპლინის შესწავლის საგანსა და გადასაწყვეტ ამოცანებს. **უზოგადესი ფილოსოფიური მეთოდი** იძლევა ინფორმაციის ცნების ერთადერთ ფორმულირებას, რომლის თანახმადაც იგი რეალური სამყაროს **ასახვას** წარმოადგენს. **გამოყენებითი სახის კერძო მეთოდები** ინფორმაციის ცნების სხვადასხვა ფორმულირებებს გვთავაზობს. კერძოდ, ასეთი მეთოდების დროს ინფორმაცია შეიძლება განხილული იყოს, როგორც:

- რაიმე საქმის მდგომარეობის მაუწყებელი შეტყობინება; მონაცემები რაიმეს შესახებ; გადასაცემი მოდელი;
- ნებისმიერ პროცესში არსებული მრავალფეროვნების გადაცემა, ასახვა; ასახული მრავალფეროვნება;
- შეტყობინების მიღების შედეგად განუსაზღვრელობის შემცირება;
- საქონელი, რომელიც წარმოადგენს განსაზღვრული მიზნის მიღწევისათვის საჭირო ცოდნის ყიდვა-გაყიდვის ობიექტს და ა.შ.

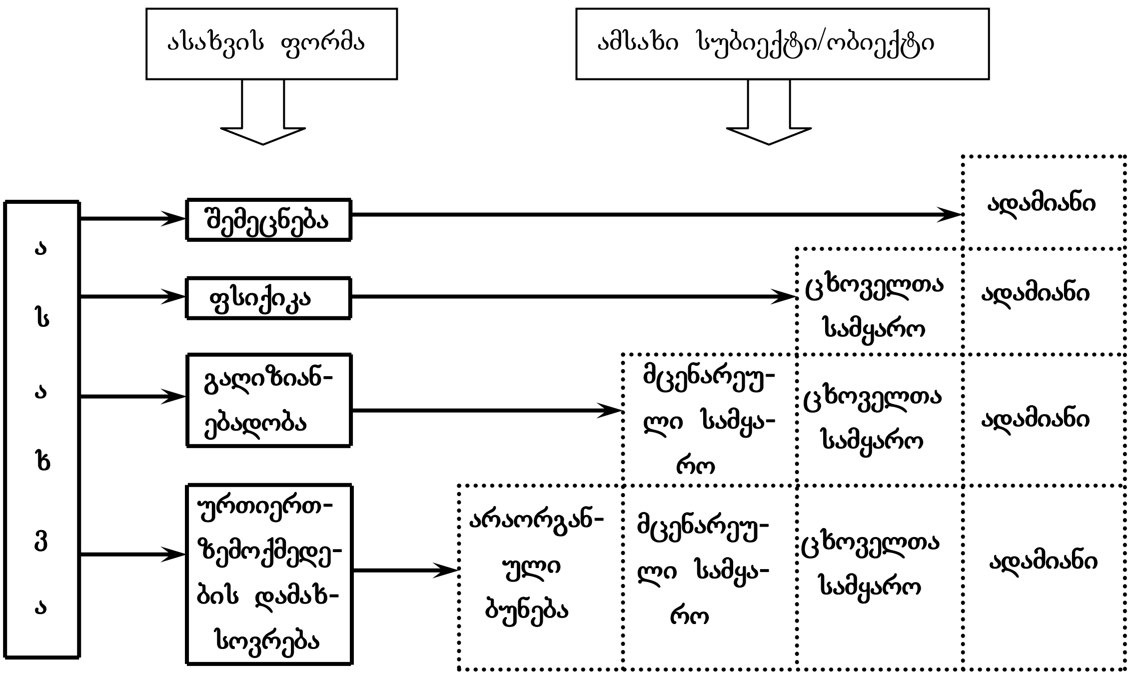
2 თავდაპირველად მიაჩნდათ, რომ სიტყვა **"ინფორმაცია"** (ინგლ. **Information** - განმარტება, გადმოცემა) მხოლოდ ადამიანის გონებისათვის დამახასიათებელ რაობას წარმოადგენდა და მას განმარტავდნენ, როგორც **"ადამიანების მიერ ზეპირად, წერილობით ან რაიმე სხვა ფორმით გადაცემულ ცოდნას, ცნობას, შეტყობინებას, უწყებას"**. შემდეგ ასეთი მიდგომა თანდათან გაფართოვდა, განზოგადდა და ინფორმაცია მიჩნეულ იქნა ზოგადად მატერიისათვის დამახასიათებელ ისეთივე საყოველთაო თვისებად, როგორიცაა მოძრაობა, სივრცე, დრო და ა.შ. ასეთი მიდგომის დროს ითვლება, რომ რეალურ ობიექტს აქვს საკუთარ მდგომარეობაში სხვა რეალური ობიექტის მდგომარეობის ადეკვატურად **ასახვის უნარი;** პირველ ობიექტს პირობითად ვუწოდოთ **ამსახი,** ხოლო მეორეს – **ასასახი ობიექტი**. ამსახი ობიექტის მდგომარეობაში ასასახი ობიექტის მდგომარეობის ასახვის ფაქტი სხვა არაფერია, თუ

არა მასში ასასახი ობიექტის შესახებ **ინფორმაციის** არსებობა. აღსანიშნავია, რომ **ამსახ და ასასახ ობიექტების მდგომარეობებს შორის ურთიერთშესაბამისობის დამყარებისთანავე ამსახ ობიექტში ჩნდება ინფორმაცია ასასახი ობიექტის მდგომარეობის შესახებ.** მაგალითად, როგორც კი დამყარდება ურთიერთშესაბამისობა ვოლტმეტრის ისრის მდებარეობასა და ვოლტმეტრზე მოდებული ძაბვის სიდიდეს შორის, ვოლტმეტრში გაჩნდება ინფორმაცია მის მიერ გასაზომი ძაბვის სიდიდის შესახებ.

ზემოთ აღნიშნულის თანახმად, **ინფორმაცია მატერიალური სამყაროსათვის დამახასიათებელი ასახვის პროცესის პროდუქტია**, ამიტომ მისი ნათლად გაგებისათვის ზოგადად განვიხილოთ ასახვის რაობა და მისი შესაძლო ფორმები.

ასახვა ეწოდება ინფორმაციის მიღების, შენახვისა და გადაცემის უნარს. მისი სახეებია შემეცნება, ფსიქიკა, გაღიზიანებადობა და ურთიერთზემოქმედების დამახსოვრება (**ნახ.1.1**). ცალ-ცალკე განვიხილოთ თითოეული მათგანი.

- **შემეცნება** ეწოდება ადამიანის გონებაში ობიექტური რეალობის ადეკვატურ ასახვას და იგი მხოლოდ ადამიანის გონებისათვის დამახასიათებელი ასახვის უმაღლესი ფორმაა;
- **ფსიქიკა** (ბერძ. **psychikos** – "სულიერი") უმაღლესად ორგანიზებული მატერიის მიერ ობიექტური რეალობის ასახვას ეწოდება, რომელიც როგორც ადამიანისათვის, ასევე ცხოველითვისაც დამახასიათებელ ასახვის ფორმას წარმოადგენს;



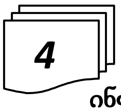
ნახ. 1.1. ცოცხალ და არაცოცხალ ბუნებაში არსებული ასახვის ფორმები

- **გაღიზიანებადობა** წარმოადგენს უჯრედშიდა წარმონაქმნების, ქსოვილებისა და ორგანოების უნარს **სტრუქტურისა და ფუნქციის შეცვლით** უპასუხოს გარემოს ყოველგვარ ცვლილებას. აღნიშნული სტრუქტურისა და ფუნქციის შეცვლით ხდება გარემოს ზემოქმედების ასახვა და, მაშასადამე, გარკვეული ინფორმაციის წარმოშობა. ასახვის განხილული ფორმა დამახასიათებელია ყველა ცოცხალი ორგანიზმისათვის, ე.ი. ადამიანისათვის, ცხოველისათვის, მცენარისა და უმარტივესი ორგანიზმებისათვის;

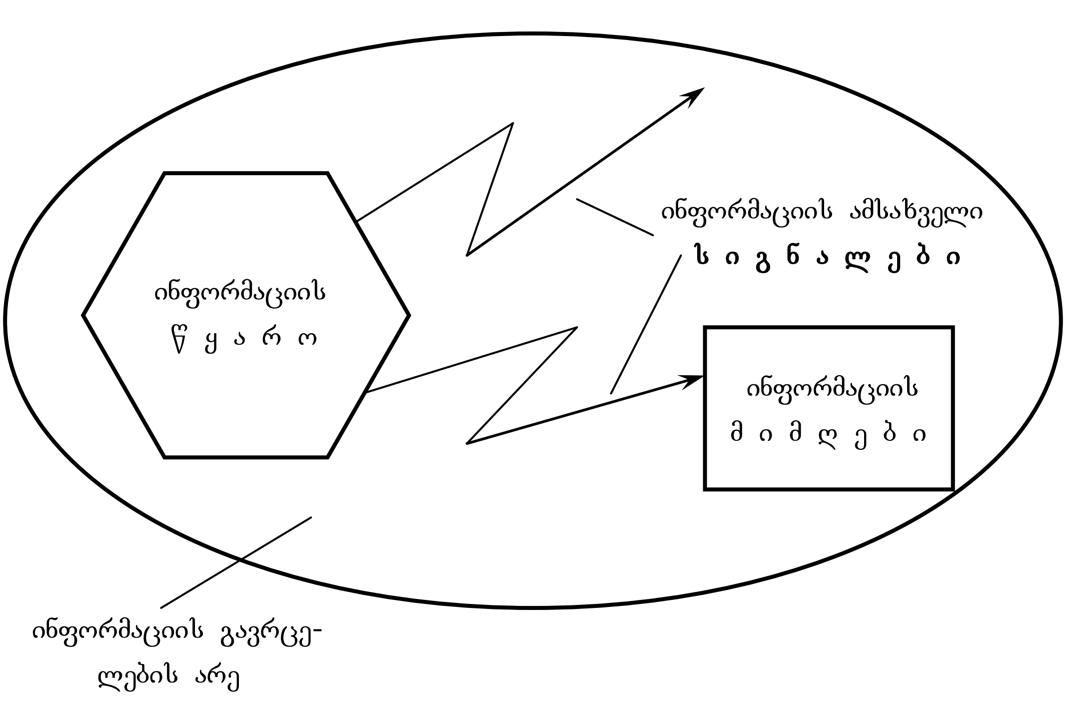
● **ურთიერთზემოქმედების დამახსოვრება** არაორგანული ბუნებისა და ელემენტალური ნაწილაკების, ე.ი. მთლიანად მატერიისათვის, დამახასიათებელი ასახვის უმარტივესი ფორმაა; ასე, მაგალითად, მინერალი ‘‘ასახავს’’ უძველეს დროს მომხდარ მოვლენებს და ამდენად მას აქვს ინფორმატიულობის თვისება, რომელსაც გეოლოგია იყენებს დასახული კონკრეტული ამოცანების გადასაწყვეტად. ინფორმატიულია გარკვეული წინადადებების შემცველი საქმიანი წერილიც, რამდენადაც იგი გარკვეული დაწესებულებებისა თუ უწყებების მიზნებს ასახავს.

ზემოთ აღნიშნულიდან გამომდინარე შეგვიძლია დავასკვნათ, რომ ინფორმაციის, როგორც ასახვის პროდუქტის, ფორმირების უნარი აქვს როგორც ადამიანს, ასევე ზოგადად ცოცხალ და არაცოცხალ ბუნებას. იგი არ წარმოადგენს არც მატერიასა და არც ენერგიას. ზემოთ მოყვანილი მაგალითებისათვის მინერალში ან საქმიან წერილში არსებული ინფორმაცია შეიძლება გაქრეს მინერალისა და საქმიანი წერილის გაქრობისას, მაგალითად, მათი დაწვის შემთხვევაში.

ინფორმაციის თავისებურებაა ის, რომ იგი მხოლოდ **ორგანიზებული სტრუქტურის** მქონე ობიექტების ანუ სისტემების ურთიერთზემოქმედების შემთხვევაში შეიძლება წარმოიშვას; ამასთანავე, აუცილებელი არ არის, რომ აღნიშნული სისტემების ელემენტებს უთუოდ ადამიანები წარმოადგენდნენ: ინფორმაციის გაცვლა შესაძლებელია მოხდეს ცოცხალ და არაცოცხალ ბუნებას, ადამიანებსა და მოწყობილობებს შორისაც. ასე მაგალითად, მინერალში თავმოყრილი ინფორმაცია ადამიანთან ურთიერზემოქმედებისას მჟღავნდება, ხოლო მცენარეები, აღიქვამს რა სინათლეს, დღისით შლის, ხოლო ღამით ხურავს თავის ყვავილებს, მზესუმზირას კი მზისკენ მისი ყვავილების ორიენტირების უნარიც გააჩნია.



ინფორმაცია, როგორც ასახვის პროდუქტი, ჩვეულებრივ გულისხმობს ორი ობიექტის არსებობას, რომელთაგანაც ერთ-ერთ (ასასახ) ობიექტს ეწოდება **ინფორმაციის წყარო**, ხოლო მეორე (ამსახ) ობიექტს – **ინფორმაციის მიმღები** (*ნახ. 1.2*).



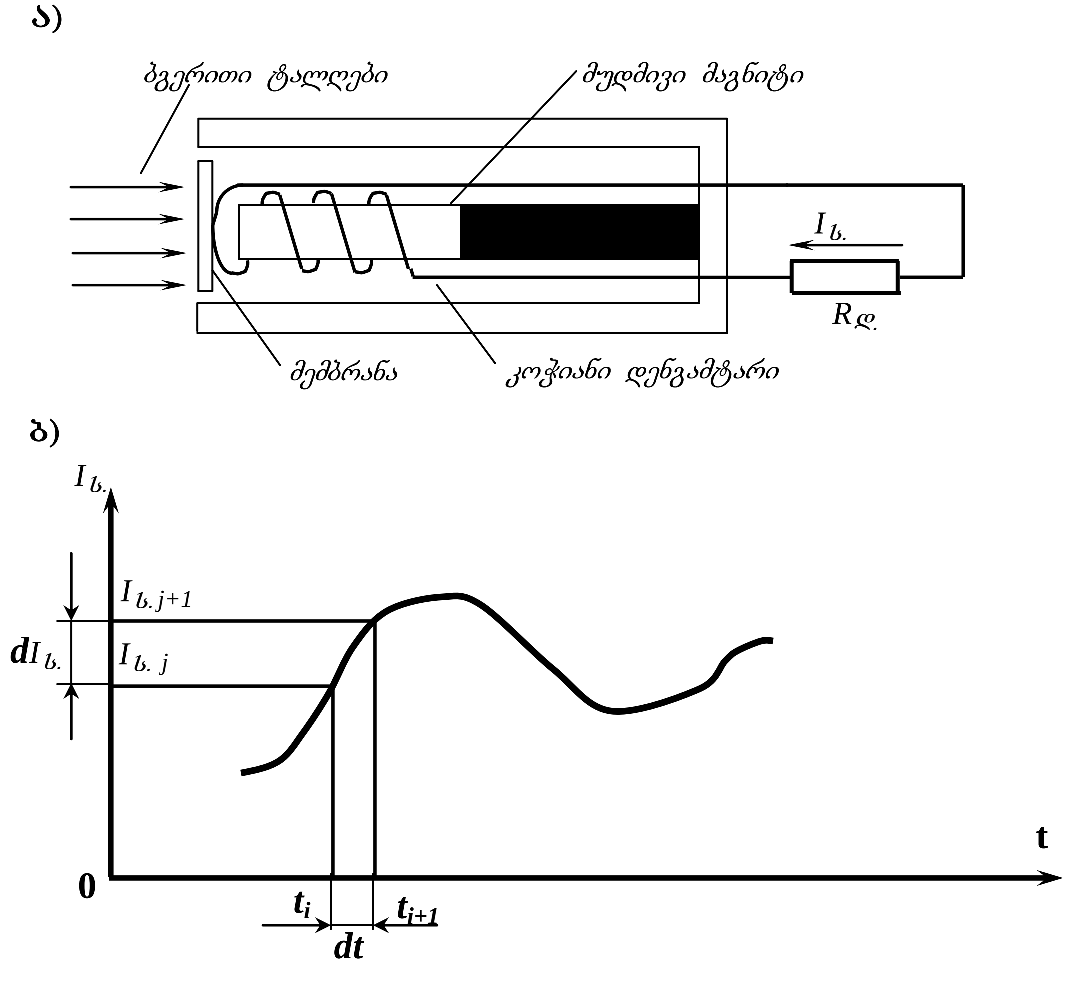
ნახ. 1.2. ინფორმაციის გაცვლის სტრუქტურული კომპონენტები

წყაროდან მიმღებამდე ინფორმაცია გარკვეულ არეში გავრცელების უნარის მქონე მატერიალურ-ენერგეტიკული (მაგალითად, ელექტრული, ბგერითი, შუქური და ა.შ.) სიგნალების სახით გადაიცემა. **სიგნალი** (ლათ. **Signium** – ‘‘ნიშანი’’) ეწოდება დაკვირვების **ობიექტის მდგომარეობის შესახებ** გარკვეული შეტყობინების (ინფორმაციის) გადამტან ფიზიკურ პროცესს (მოვლენას).

ინფორმაცია შეიძლება გადაიცეს უწყვეტად ან დისკრეტულად. ამაზე დამოკიდებულებით განასხვავებენ **უწყვეტ** და **დისკრეტულ ინფორმაციას**. **პირველ შემთხვევაში** ინფორმაციის გადასაცემად გამოიყენება ისეთი სახის ცვლადი სიგნალი (ძაბვა, დენი და ა.შ.), რომელიც დროში **დაკვირვების ობიექტის** მდგომარეობის ცვლილების **ანალოგურად** იცვლება. ასეთ სიგნალს **ანალოგური სიგნალი** ეწოდება. **მეორე შემთხვევაში** ინფორმაცია გადაიცემა ცალკეული სიგნალების მიმდევრობის სახით და ასეთ სიგნალს **დისკრეტული სიგნალი** ეწოდება.

ანალოგური სიგნალების სახით წარმოდგენილ ინფორმაციის გადამამუშავებელ კომპიუტერს **ანალოგური კომპიუტერი**, ხოლო დისკრეტული სიგნალების სახით წარმოდგენილი ინფორმაციის გადამამუშავებელ კომპიუტერებს – **დისკრეტული ანუ ციფრული კომპიუტერი** ეწოდება. სახელწოდება ‘‘ციფრული’’ იმიტომ გამოიყენება, რომ ზემოთ აღნიშნული მიმდევრობის შემადგენელი ცალკეული სიგნალები გარკვეულ აბსტრაქტულ ციფრებთან იდენტიფიცირდება.

აღსანიშნავია, რომ გარდა ანალოგური და ციფრული კომპიუტერებისა, დღეისათვის გამოჩნდნენ **იმპულსური, ოპტიკური და კვანტური კომპიუტერებიც**, მაგრამ ისინი ვიწროსპეციალიზებულ ან საექსპერიმენტო კომპიუტერების კლასს მიეკუთვნება და მათი განხილვა სცილდება მოცემულ დისციპლინაში განსახილველი საკითხების ფარგლებს.



ნახ. 1.3. მიკროფონით ანალოგური სიგნალის წარმოქმნის მაგალითი

5 ანალოგური სიგნალის ცნების თვალნათლად გაცნობიერებისათვის განვიხილოთ მიკროფონის ყველაზე მეტად გავრცელებული ელექტროდინამიკური მიკროფონის ფუნქციონირების მაგალითი (**ნახ.1.3**). იგი წარმოადგენს მუდმივი მაგნიტით შექმნილ ძლიერ მაგნიტურ მემბრანასთან მიერთებულ კოჭიან დენგამტარს (**იხ. ნახ. 1.3,ა**). ბგერითი რხევებით წარმოქმნილი წნევა ზემოქმედებს თხელ მემბრანაზე; ეს უკანასკნელი იწყებს რხევას, რომლის სიდიდე ბგერის როგორც ამპლიტუდის, ასევე სიხშირის პროპორციულია;

მემბრანის ზემოქმედებით რხევას იწყებს კოჭიანი დენგამტარით წარმოქმნილი ჩაკეტილი კონტური, რაც კონტურის გადამკვეთი მაგნიტური ნაკადის ცვლილებას იწვევს; ეს უკანასკნელი კონტურში წარმოშობს **ელექტრომაგნიტური ინდუქციის ძალას**, რომლის სიდიდე ბგერითი რხევების როგორც ამპლიტუდის, ასევე სიხშირის პროპორციულია.

ელექტრომაგნიტური ინდუქციის ძალა კონტურში აღძრავს $I_{\text{ს}}$ დენს, რომელსაც ექნება ბგერითი რხევების ამპლიტუდისა და სიხშირის ცვლილების **ანალოგურად ცვლადი სიდიდე**; ამიტომ მას **ბგერის ანალოგურ სიგნალს** უწოდებენ. t დროზე $I_{\text{ს}}$ დენის დამოკიდებულების $I_{\text{ს}} = f(t)$ ფუნქციას აქვს **1.3,ბ** ნახაზზე ნაჩვენები დამოკიდებულების ანალოგური სახე. როგორც აღნიშნული ნახაზიდან ჩანს, t არგუმენტის უსასრულოდ მცირე dt ნაზრდს ფუნქციის მნიშვნელობის უსასრულოდ მცირე $dI_{\text{ს}}$ ნაზრდი შეესაბამება, რაც $I_{\text{ს}} = f(t)$ ფუნქციის უწყვეტობის ნიშანია; ე.ი **ანალოგური სიგნალი** წარმოადგენს **უწყვეტ**, ანუ არანახტომისებური ზრდის უნარის მქონე **სიგნალს.**

როგორც ქვემოთ ვაჩვენებთ, არგუმენტის **დისკრეტიზაცია** ანალოგური სიგნალის **დისკრეტულ** ანუ **წყვეტილ** სიგნალად გარდაქმნის საშუალებას გვაძლევს, ხოლო შემდგომ **დაკვანტვის** მეთოდის გამოყენებით მას შეიძლება მივცეთ რიცხვითი სახე.

6 ზემოთ მოყვანილი მასალის საფუძველზე ინფორმაციის ცნება შეიძლება იმგვარად განვსაზღვროთ, რომ მისი შინაარსი მაქსიმალურად შეესაბამებოდეს კომპიუტერული სისტემების შემსწავლელი დისციპლინის შესწავლის საგანსა და მის მიერ გადასაწყვეტ ამოცანებს. კერძოდ:

ინფორმაცია წარმოადგენს რეალური სამყაროს ობიექტური ასახვის სპეციფიკურ ფორმას, რომლისთვისაც დამახასიათებელია შემდეგი სამი თავისებურება:

1) ასახვა ხდება სიგნალების ერთობლიობის წარმოშობის გზით;

2) სიგნალები ვლინდება "ინფორმაციის მიმღებთან" მათი ურთიერთმოქმედების პროცესში;

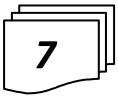
3) ინფორმაციის მიმღები საშუალებას იძლევა მოხდეს გარემოდან სიგნალების გამოყოფა, მათი რეგისტრირება და ამა თუ იმ კრიტერიუმების შესაბამისად აღნიშნული სიგნალების იდენტიფიცირება.

მოცემული განსაზღვრებიდან გამოდის, რომ:

- ინფორმაცია ობიექტურია, ვინაიდან იგი მატერიისთვის დამახასიათებელ ასახვის თვისებას წარმოადგენს;
- ინფორმაცია მხოლოდ ობიექტების ურთიერთზემოქმედებით წარმოშობილი სიგნალების სახით არსებობს;
- ინფორმაციის "მიმღების" "აწყობაზე" დამოკიდებულებით ერთი და იგივე ინფორმაცია სხვადასხვა ადრესატის მიერ სხვადასხვანაირად შეიძლება იქნეს ინტერპრეტირებული.

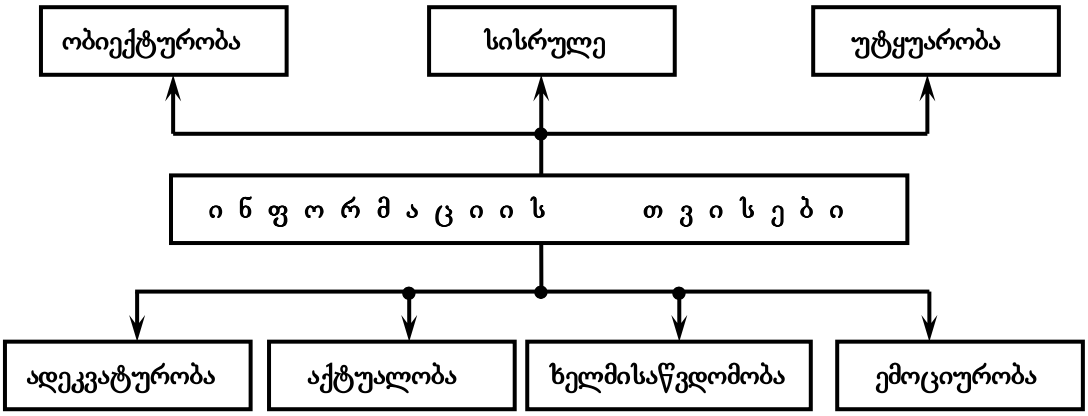
ადამიანი სიგნალებს აღიქვამს გრძნობის ორგანოებით და მათ იდენტიფიცირებას ახდენს ტვინის საშუალებით. ამიტომ, მაგალითად, სხვადასხვა სმენადობის მქონე ადამიანი ერთი და იგივე ბგერით სიგნალს სხვადასხვანაირად აღიქვამს, ხოლო ყრუ საერთოდ ვერ აღიქვამს მას. ტექნიკაში ინფორმაციის მიმღებები სიგნალებს სხვადასხვა საზომი და მარრეგისტრირებელი აპარატურის საშუალებით აფიქსირებს; ამიტომ რაც უფრო მაღალი

მგრძნობიერებით ახდენს მიმღები სიგნალების რეგისტრირებას და რაც უფრო სრულყოფილი ალგორითმებით ახდენს მათ დამუშავებას, მით უფრო დიდი მოცულობის ინფორმაციის მიღების საშუალებას გვაძლევს იგი.



ინფორმაციის ხარისხი მომხმარებლის საჭიროებასთან ინფორმაციის შესაბამისობის განმსაზღვრელი პარამეტრია. ვინაიდან მომხმარებლის საჭიროება შექმნილ ვითარებაზე დამოკიდებულებით იცვლება, ამიტომ ინფორმაციის ხარისხი წარმოადგენს ფარდობით პარამეტრს. შეიძლება გამოვყოთ მისი მახასიათებელი შემდეგი თვისებები (**ნახ. 1.4**):

- **ობიექტურობა** გვიჩვენებს ინფორმაცია თუ რამდენადაა დამოუკიდებელი როგორც არსებულ შეხედულებებზე, ასევე მისი მიღებისა და დამუშავების მეთოდებზე; რაც უფრო ნაკლები სუბიექტურობის ელემენტი შეაქვს ინფორმაციაში არსებულ შეხედულებებს, აგრეთვე მისი მიღებისა და დამუშავების მეთოდებს, მით უფრო მეტია მისი ობიექტურობა;
- **სისრულე** არის ინფორმაციაში არსებული მახასიათებელთა რაოდენობისა და საკმარისობის ფუნქცია; კერძოდ, **სრულად** ითვლება ინფორმაცია, რომელიც შეიცავს სწორი გადაწყვეტილების მისაღებად საჭირო მინიმალური რაოდენობის მახასიათებლებს. ინფორმაციაში ნაკლები ან ჭარბი მახასიათებლების არსებობა ამცირებს მისი გამოყენების საფუძველზე მიღებულ გადაწყვეტილებათა ეფექტურობას.



ნახ. 1.4. ინფორმაციის თვისებები

- **უტყუარობა** განსაზღვრავს ინფორმაციის სწორად აღქმადობას. ობიექტური ინფორმაცია ყოველთვის უტყუარია, მაგრამ უტყუარი ინფორმაცია შეიძლება იყოს როგორც ობიექტური, ასევე სუბიექტური;
- **ადეკვატურობა** საქმის რეალურ ობიექტურ მდგომარეობასთან ინფორმაციის შესაბამისობის ხარისხის მაჩვენებელია;
- **ხელმისაწვდომობა** განსაზღვრავს თუ რამდენადაა შესაძლებელი ინფორმაციის მიღება;
- **აქტუალობა** დროის მიმდინარე მომენტთან ინფორმაციის შესაბამისობის ხარისხის მაჩვენებელია;
- **ემოციურობა** ადამიანებისათვის სხვადასხვა ემოციის გამოწვევის თვისებაა. ინფორმაციის ამ თვისებას მედიაინფორმაციის მწარმოებლები იყენებენ. რაც უფრო ძლიერია ინფორმაციის მიერ გამოწვეული ემოცია, მით უფრო მაღალია ასეთი ინფორმაციისათვის ყურადღების მიქცევისა და მისი დამახსოვრების ალბათობა.

გარე სამყაროდან ინფორმაციის მიღება, მისი ანალიზი და გენერირება წარმოადგენს ადამიანის ერთ-ერთ ძირითად ფუნქციას, რომლითაც იგი განსხვავდება დანარჩენი ცოცხალი სამყაროსაგან.

1.2. საკვანძო ცნებები, განსაზღვრებები და დებულებები მოკლე ისტორიული ექსკურსის ფონზე

"ცნების განსაზღვრა უმნიშვნელოვანეს როლს თამაშობს თეორიულ და პრაქტიკულ საქმიანობაში. საგნის მნიშვნელობის შესახებ ცოდნის მოკლე სახით გამოხატვა სინამდვილის შემეცნების არსებითი მომენტია."
ვ.ი. კირილოვი, ა.ა.სტარჩენკო "ლოგიკა" [27]

"ისტორია – ცხოვრების მასწავლებელია"; "ისტორიის არცოდნა - მუდმივად ბავშვად დარჩენას ნიშნავს"
ციცერონი (***106 -43*** ჩვენ წ/აღრიცხვამდე)

ინფორმაციას საზოგადოებაში აქვს გარკვეული ფუნქციები, რომელთაგანაც მთავარია: **შემეცნებითი ფუნქცია**, რომლის მიზანია ახალი ინფორმაციის მიღება; **საკომუნიკაციო ფუნქცია**, რომლის მიზანია ადამიანებს შორის ურთიერთობის უზრუნველყოფა და **მმართველობითი ფუნქცია,** რომლის მიზანია უზრუნველყოს ინფორმაციის მიმღები მართული სისტემის მიზანმიმართული ქცევა.

ზემოთ ჩამოთვლილი ფუნქციებიდან ყურადღება გავამახვილოთ **მმართველობით ფუნქციზე;** კერძოდ, განვიხილოთ ინფორმაციის საფუძველზე ავტომატური და/ან ავტომატიზებული სისტემის მიერ **მმართველი ზემოქმედების წარმოქმნის** პროცესი. აღნიშნული ზემოქმედების წარმოსაქმნელად ინფორმაციაზე გარკვეული მოქმედების ჩატარებაა საჭირო, რაც წარმოადგენს **საინფორმაციო პროცესისა** და **საინფორმაციო სისტემის** განსაზღვრების ფორმირების საფუძველს.

საინფორმაციო პროცესი ეწოდება ინფორმაციაზე შესასრულებელი მოქმედებების ერთობლიობას

სისტემა (ბერძ. ***systëma*** – მთელი, ნაწილებისაგან შედგენილი; შეერთება) ეწოდება ერთმანეთთან და გარე საგნებთან ურთიერთზემომქმედ სხვადასხვაგვარ ელემენტების ან ნაწილების მოწესრიგებულ ერთობლიობას, რომლებიც გაერთიანებულია და ფუნქციონირებს საერთო მიზნის მისაღწევად.

საინფორმაციო სისტემა ეწოდება დასახული მიზნის მიღწევის მიზნით ინფორმაციის შესანახად, დასამუშავებლად და გადასაცემად გამოყენებული საშუალებების, მეთოდებისა და პერსონალის ურთიერთდამოკიდებულ ერთობას.

ვინაიდან ინფორმაციის მატერიალური გადამტანები სიგნალებია, ამიტომ ზემოთ ჩამოთვლილი ეტაპები ფაქტობრივად სიგნალების **მიმოქცევისა** და **გარდაქმნის** ეტაპებია (**ნახ.1.5**). მოკლედ განვიხილოთ თითოეული მათგანი.

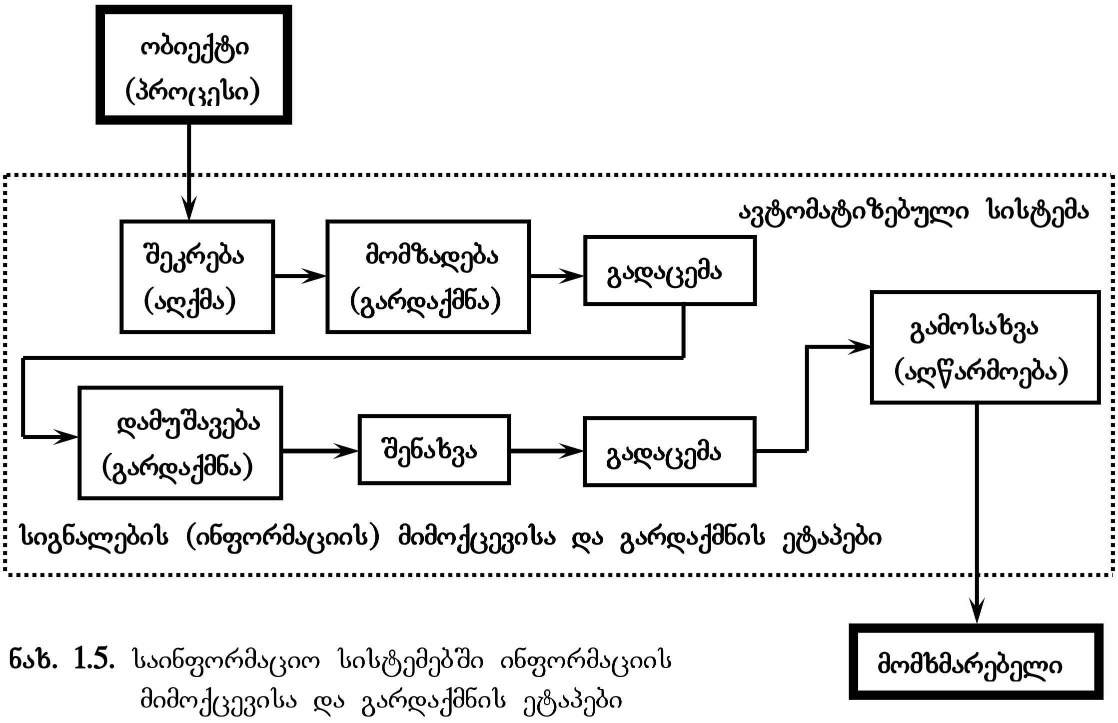
1) **ინფორმაციის შეკრების (აღქმის) ეტაპზე** ხორციელდება რაიმე ობიექტის (პროცესის) შესახებ ინფორმაციის მიზანმიმართული ამოღება და გაანალიზება; ამის შედეგად ფორმირდება ობიექტის სახე და ხდება მისი ამოცნობა-შეფასება. ამ ეტაპზე მთავა-

რია ხელშეშლებისაგან (ე.წ. ხმაურისაგან) სასარგებლო ინფორმაციის გამოყოფა, რაც ხშირად ძალიან ძნელია. აღქმის (შეკრების) უმარტივესი სახეა ობიექტში გარკვეული მდგომარეობის არსებობისა და არარსებობის ფაქტის დადგენა; პირველი ფაქტი პირობით აღინიშნება სიტყვით "დიახ" ანუ ციფრით "1", ხოლო მეორე ფაქტი კი სიტყვით "არა", ანუ ციფრით "0". აღქმის უფრო რთული სახეა ობიექტის მდგომარეობათა პარამეტრების გაზომვა.

2) **მომზადების (გარდაქმნის) ეტაპზე** ხდება ინფორმაციის პირველადი გარდაქმნა. ამ ეტაპზე სრულდება ისეთი ოპერაციები, როგორებიცაა ნორმალიზაცია, ანალოგურ-ციფრული გარდაქმნა, დაშიფვრა და ა.შ. ზოგჯერ ამ ეტაპს **აღქმის დამხმარე ეტაპად განიხილავენ.**

ინფორმაციის ზემოთ აღნიშნული აღქმისა და მომზადების ეტაპების ჩატარების შედეგად ფორმირდება გადაცემის, შენახვისა და დამუშავებისათვის მოსახერხებელი ფორმის **სიგნალი.**

3) **გადაცემის ეტაპზე** ინფორმაცია ერთი ადგილიდან (გამგზავნიდან) მეორე ადგილზე (მიმღებ-ადრესატთან) გადაიგზავნება. გადასაცემად გამოიყენება სხვადასხვა ფიზიკური ბუნების (ყველაზე ხშირად, ელექტრული, ელექტრომაგნიტური და ოპტიკური) არხები. გადაცემის პროცესში სიგნალები განიცდის სხვადასხვა სახის ხელშეშლების ზეგავლენას. აღნიშნული არხების ბოლოში სიგნალის ამოღებას აქვს მისი მეორადი აღქმის სახე.



ნახ. 1.5. საინფორმაციო სისტემებში ინფორმაციის მიმოქცევისა და გარდაქმნის ეტაპები

4) მეორადად აღქმული ინფორმაციის **დამუშავების (გარდაქმნის)** ეტაპზე დგინდება ინფორმაციის ზოგადი და არსებითი ურთიერთდამოკიდებულებები, რომლებიც აუცილებელია სისტემის ფუნქციონირებისათვის. ამ ეტაპზე (ისევე, როგორც სხვა ეტაპებზე) ინფორმაციას გარდაქმნის უშუალოდ საინფორმაციო ტექნიკა, ან ადამიანი.

ზოგადად **ინფორმაციის დამუშავება** ნიშნავს ინფორმაციის გარდაქმნას, რომლის დროსაც ვიყენებთ ლოგიკისა და მათემატიკის კანონებს, "საღ აზრზე" დაფუძნებულ ინტუიციას, განზოგადოებულ ცდას, მიღებულ შეხედულებებსა და ქცევის ნორმებს. ამუშავების შედეგად მიიღება აგრეთვე ინფორმაცია, რომელიც სხვა ფორმით იქნება წარმოდგენილი (მაგალითად, რაღაც ნიშნების მიხედვით იქნება მოწესრიგებული), ან რომელშიც მოთავსებული იქნება პასუხები დასმულ კითხვებზე. დამუშავების პროცესის ფორმალიზების შემთხვევაში იგი შეიძლება ტექნიკურმა საშუალებებმაც შეასრულოს. ამ სფეროში მნი-

შენელოვანი ცვლილებების მოხდენა დაკავშირებულია ინფორმაციის უნივერსალური გარდამქმნელის – **კომპიუტერის** გამოჩენასთან; კერძოდ, კომპიუტერის შექმნის შემდეგ გაჩნდა ცნებები: მონაცემები და მონაცემების დამუშავება.

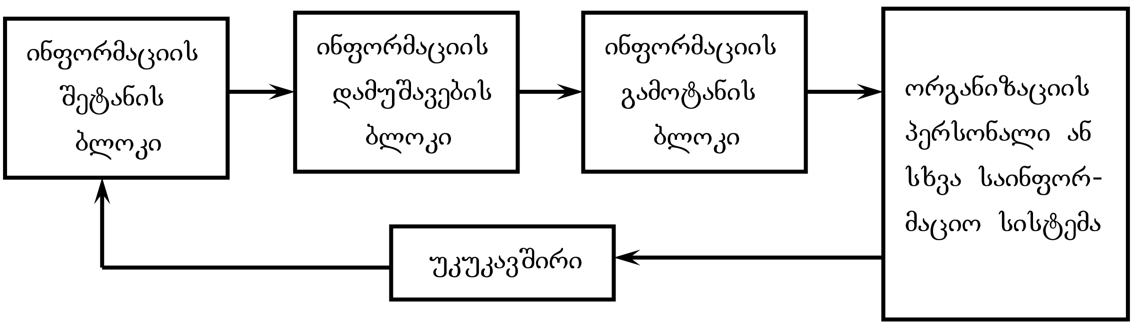
მონაცემები ეწოდება ფორმალიზებული (კოდირებული) სახით წარმოდგენილ ფაქტებსა და ცნობებს, რომლებიც ამა თუ იმ სახის მზიდშია (მეხსიერებაშია) ჩაწერილი და რომელთა დამუშავება ტექნიკური საშუალებებით (უპირველეს ყოვლისა, კომპიუტერებით) არის შესაძლებელი.

მონაცემების დამუშავება ეწოდება მონაცემებზე სხვადასხვა, უპირველეს ყოვლისა, არითმეტიკული და ლოგიკური, ოპერაციების ჩატარებას, რომლის შედეგადაც მიიღება ობიექტურად აუცილებელი ახალი მონაცემები.

5) შენახვის ეტაპზე შემდგომი გამოყენების მნიშნით ინფორმაცია ჩაიწერება დამხსომებელ მოწყობილობაში. ინფორმაციის შესანახად ძირითადად გამოიყენება ნახევრადგამტარული, მაგნიტური და ოპტიკური მზიდები.

6) ინფორმაციის გამოსახვის (აღწარმოების) ეტაპი წინ უნდა უსწრებდეს ისეთ ეტაპებს, რომლებშიც მონაწილეობს ადამიანი. ამ ეტაპის მიზანია ადამიანს ინფორმაცია ისეთი მოწყობილობების დახმარებით მიაწოდოს, რომლებსაც მის გრძნობის ორგანოებზე ზემოქმედება შეუძლია.

3 საინფორმაციო სისტემის სტრუქტურა და მასში მიმდინარე საინფორმაციო პროცესები შეიძლება სქემური სახით წარმოვადგინოთ (**ნახ.1.6**). მასში გამოყენებული ინფორმაციის შეტანის, დამუშავებისა და გამოტანის ბლოკები წარმოქმნის საინფორმაციო სისტემის აპარატურულ და პროგრამულ საშუალებებს. უკუკავშირი შეიძლება განხორციელდეს როგორც ადამიანების, ასევე პროგრამულ-აპარატურული საშუალებებით. უკუკავშირის არხით მიღებული ინფორმაცია შესასვლელი ინფორმაციის კორექტირებისთვისაა განკუთვნილი, ხოლო გამოსასვლელი ინფორმაცია გადაწყვეტილების მისაღებად გამოიყენება.



ნახ. 1.6. საინფორმაციო სისტემის განზოგადებული სტრუქტურა

საინფორმაციო სისტემა შეიძლება ქვესისტემების ერთობლიობის სახითაც წარმოვადგინოთ. **ქვესისტემა** რაიმე ნიშნის მიხედვით სისტემისაგან გამოყოფილი გარკვეული ნაწილია.

საინფორმაციო სისტემების კლასიფიცირება შესაძლებელია მრავალი სხვადასხვა ნიშნის მიხედვით მოვახდინოთ; მაგალითად, **გამოყენების სფეროს მიხედვით** საინფორმაციო სისტემები იყოფა ადმინისტრაციულ, სასწავლო, სამედიცინო, სამხედრო და ა.შ. სისტემებად; **ტერიტორიული დაყოფის მიხედვით** განასხვავებენ რაიონის, ქალაქის, დაბის, ავტონომიური რესპუბლიკისა და ა.შ. სისტემებს; **კონკრეტული საინფორმაციო პროცესების ორგანიზაციის შესაძლებლობების მიხედვით** განასხვავებენ საინფორმაციო-საცნობარო, სა-

ინფორმაციო-საძიებო სისტემებს, მონაცემების დამუშავებისა და გადაცემის სისტემებს, კავშირგაბმულობის სისტემებსა და ა.შ.

საინფორმაციო სისტემაში საინფორმაციო პროცესის პროცედურები შეიძლება შესრულდეს ხელით ან სხვადასხვა ტექნიკური საშუალებების (კომპიუტერების, სატელეკომუნიკაციო საშუალებების, პერიფერიული და საორგანიზაციო-ტექნიკური საშუალებების) გამოყენებით, ე.ი. მოხდეს საინფორმაციო სისტემების ავტომატიზება. კომპიუტერები და შესაბამისი პროგრამული უზრუნველყოფა რადიკალურად ცვლის ინფორმაციის დამუშავების მეთოდებსა და ტექნოლოგიას. ამიტომ არაავტომატიზებული და ავტომატიზებული საინფორმაციო სისტემები ერთმანეთისაგან თვისობრივად განსხვავდება.

არაავტომატიზებული საინფორმაციო სისტემა ეწოდება საინფორმაციო სისტემას, რომელშიც ინფორმაციის დამუშავების ყველა ოპერაციას მმართველობითი მუშაკები ინფორმაციის დამუშავების ტექნიკური საშუალებების გამოუყენებლად ასრულებენ.

ავტომატიზებული საინფორმაციო სისტემა ეწოდება საინფორმაციო სისტემას, რომელშიც საინფორმაციო პროცესის რუტინული ოპერაციების მნიშვნელოვან ნაწილს სპეციალური მეთოდების გამოყენებით დამუშავებული ალგორითმების მარეალიზებელი ტექნიკური საშუალებები ასრულებს ადამიანის ჩაურევლად ან ამ უკანასკნელის მინიმალური ჩარევით.

4

ავტომატიზებული საინფორმაციო სისტემების უმრავლესობა **ლოკალურ სისტემებს** წარმოადგენს; ისინი დაწესებულებების, წარმოებების დონეზე ფუნქციონირებს. დღეს მთელი ძალისხმევა ასეთი სისტემების ჯერ **კორპორატიულ**, ხოლო შემდეგ – **რეგიონალურ** და **გლობალურ სისტემებად** გაერთიანებისაკენ არის მიმართული.

ერთ ქსელში სხვადასხვა სახის ინფორმაციის გაერთიანება საშუალებას გვაძლევს ინფორმაცია წარმატებით გამოვიყენოთ ადმინისტრაციულ მმართველობასთან, დაგეგმვასთან, სამეცნიერო კვლევებთან, კონსტრუქტორულ დამუშავებებთან, წარმოების ტექნოლოგიასთან, მომარაგებასთან და აღრიცხვასთან დაკავშირებული ფართო სპექტრის ამოცანების გადასაწყვეტად.

უფრო მაღალი დონის სისტემები როგორც ფუნქციონურად ასევე მათი ტექნიკური რეალიზაციის მიხედვით წარმოადგენს **ტერიტორიულად განწერტებულ იერარქიულ სისტემებს.**

ტერიტორიულად განწერტებული სისტემების ურთიერთზემოქმედების უზრუნველსაყოფად საჭიროა გამოვიყენოთ კავშირგაბმულობის გრძელი და საიმედო არხები, ხოლო დასამუშავებელი ინფორმაციის მოცულობის გაზრდა მოითხოვს მაღალმწარმოებული კომპიუტერების გამოყენებას.

ეკონომიკური თვალსაზრისით გამართლებულია ავტომატიზაციისათვის აუცილებელი ზემოთ ჩამოთვლილი ძვირი ტექნიკური საშუალებები (კომპიუტერები, კავშირის არხები) და დამუშავებული ინფორმაცია (მონაცემების ბაზა) კოლექტიურად იქნეს გამოყენებული, რისთვისაც შეიქმნა **კოლექტიური მოხმარების საინფორმაციო-გამომთვლელი ქსელები**.

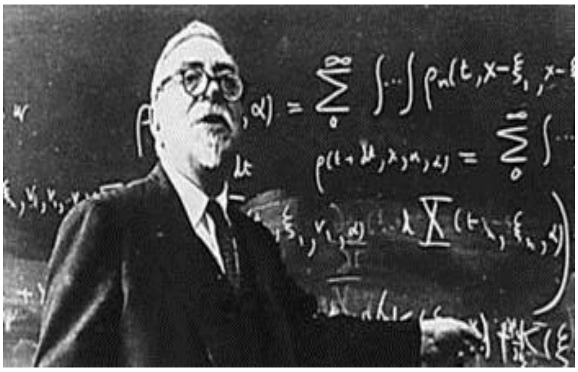
მისაწოდებელი ინფორმაცია თუ რაიმე ობიექტიდან (პროცესიდან) ამოიღება, ხოლო ფორმირებული **გამოსასვლელი ინფორმაცია** იმავე ობიექტის მდგომარეობის მიზანმიმართული ცვლილებისათვის იქნება გამოყენებული, მაშინ ასეთ ავტომატიზებულ საინფორმაციო სისტემას **მართვის ავტომატიზებული სისტემა (მას)** ეწოდება. ამ უკანასკნელში ძირითად მმართველი ზემოქმედებებს ირჩევს ანუ გადაწყვეტილებებს იღებს ადამიანი.

4

მართვა და **ინფორმაცია** სხვადასხვა (ტექნიკურ, ბიოლოგიურ ან სოციალურ) სისტემებში მართვის ზოგადი პრინციპების შემსწავლელი მეცნიერების – **კიბერნეტიკის** (ლათ. **kybernētikē** - მართვის ხელოვნება) ძირითადი ცნებებია.

ცნება **"კიბერნეტიკა"** პირველად ***XIX*** საუკუნის პირველ ნახევარში შემოიღო ფრანგმა ფიზიკოსმა **ანდრე მარი ამპერმა**, რომელმაც კიბერნეტიკა უწოდა ადამიანებისა და საზოგადოების მართვის საკითხების შემსწავლელ მეცნიერებას.

ანდრე მარი ამპერი
(1774 – 1836)



ნორბერტ ვინერი
(1894– 1964)

თანამედროვე გაგებით **კიბერნეტიკა** ეწოდება მეცნიერებას, რომელიც ნებისმიერი ფიზიკური ბუნების ორგანიზებულ სისტემაში არსებულ კავშირებსა და მართვას (თვითმართვას) ერთიანი პოზიციიდან შეისწავლის. მისი ფუძემდებელია ამერიკელი მათემატიკოსი **ნორბერტ ვინერი**.

მართვის სისტემა წარმოადგენს გასაკონტროლებელი ობიექტებისა და მათზე ზემოქმედი საშუალებების შესახებ მონაცემების შესაკრებად განკუთვნილი საშუალებების სისტემატიზებულ (მკაცრად განსაზღვრულ) ნაკრებს, რომელიც გარკვეული მიზნების მისაღწევადაა განკუთვნილი.

მართვის ნებისმიერი სისტემა შეიცავს მართვის ობიექტსა და სუბიექტს.

მართვის ობიექტს წარმოადგენს გარკვეული მიზნის მიღწევისაკენ მიმართული სამუშაოთა კომპლექსის შემსრულებელი და ამისათვის აუცილებელი მატერიალური, ფინანსური და სხვა სახის რესურსების მფლობელი კოლექტივი.

მართვის სუბიექტი ახდენს ეკონომიკური ობიექტის ფუნქციონირების მიზნების ფორმირებას და აკონტროლებს ამ მიზნების მისაღწევად საჭირო ქმედებების შესრულებას.

მართვის სისტემებს, რომლებშიც ადამიანები მართვის ობიექტების სახით იღებენ მონაწილეობას, **მენეჯმენტის** (ინგლ. ***management*** - "მართვა", "ხელმძღვანელობა", "ადმინისტრაცია", "დირექცია") **სისტემას** უწოდებენ.

მართვა ეწოდება მართვის ობიექტზე განხორციელებულ მიზანმიმართულ ზემოქმედებას, რომელიც ცვლის ან ინარჩუნებს მართვის ობიექტის მდგომარეობას. მისი ძირითადი ფუნქციებია დაგეგმვა, აღრიცხვა, ანალიზი, კონტროლი და რეგულირება.

მართვის ფუნქციების შესრულება ევალება **მართვის აპარატს**, რომელიც მოიცავს ცალკეული ფუნქციების შემსრულებელ სამსახურებსა და განყოფილებებს, კერძოდ, საგეგმო და საფინანსო განყოფილებებს, ბუღალტერიას, მომარაგება-გასაღების განყოფილებას და ა.შ. მართვის კერძო ფუნქციების შემსრულებელი ურთიერთდაკავშირებული ორგანოების ერთობლიობა წარმოქმნის **მართვის სისტემის სტრუქტურას.**

გამოყოფენ ეკონომიკური ობიექტის მართვის სტრატეგიულ, ტაქტიკურ (ფუნქციონალურ) და ოპერატიულ (მიმდინარე) დონეებს.

მართვის სტრატეგიულ დონეზე მუშავდება გრძელვადიანი მიზნების მისაღწევად საჭირო გადაწყვეტა. აქ განისაზღვრება მიზანი, აგრეთვე სრულდება საპროგნოზო დაგეგმვა. მართვის დანარჩენი ფუნქციები აქ არ განიხილება.

მართვის ტაქტიკურ (ფუნქციონალურ) დონეზე მუშავდება საშუალოვადიანი, მიმდინარე და ოპერატიულ-კალენდალური გეგმები და კონტროლდება მათი შესრულება. საკონტროლო ფუნქციების მნიშვნელოვანი ნაწილი რეალიზდება საბუღალტრო და სტატიკური აღრიცხვის დახმარებით, რომელიც იყენებს პირველადი აღრიცხვის მონაცემებს და აჯგუფებს მათ საჭიროების შესაბამისად. ამ დონეზე დიდი ადგილი ეთმობა ანალიზისა და მისგან გამომდინარე რეგულირების ფუნქციების შესრულებას.

მართვის ოპერატიულ (მიმდინარე) დონეზე რეალიზდება ოპერატიული აღრიცხვის ფუნქციები, რომლებიც მართვის ობიექტში მომხდარი ყველა ცვლილების შესახებ პირველადი ინფორმაციის შეკრების საშუალებას გვაძლევს. სწორედ ეს ინფორმაცია გადაეცემა მომდევნო დონეს და უფრო სრულყოფილი აღრიცხვისა და ანალიზისათვის გამოიყენება.

მართვა ეფუძნება ინფორმაციას. მართვის პროცესში წარმოიშობა მართვის სუბიექტსა და ობიექტს, აგრეთვე მათსა და გარე არეს შორის ცირკულირებადი **საინფორმაციო ნაკადები.**

ეკონომიკური ობიექტის მდგომარეობის შესახებ ინფორმაციისა და გარედან შემოსული ინფორმაციის საფუძველზე მართვის სუბიექტი განსაზღვრავს ეკონომიკური ობიექტის მიზანს და გამოიმუშავებს მართვის ობიექტზე განსახორციელებელ ზემოქმედებას (***პირდაპირი კავშირი***).

ეკონომიკურ ობიექტში გარკვეული ცვლილებები ხდება მისი ფუნქციონირების პროცესში. მართვის სუბიექტი იღებს ინფორმაციას ამ ცვლილებების შესახებ (***უკუკავშირი***) და გარე ზემოქმედებების (სადირექტივო ინფორმაციის, კონტრაგენტებისაგან მოსული ინფორმაციების და ა.შ.) გათვალისწინებით გამოიმუშავებს ახალ მმართველობით გადაწყვეტებს და ხელახლა ზემოქმედებს მართვის ობიექტზე.

5

შრომა ოდითგანვე წარმოადგენდა ადამიანის საქმიანობის ერთ-ერთ ძირითად სახეს, რომელიც მისი არსებობისათვის იყო აუცილებელი. იგი მისგან გონებრივი და ფიზიკური ძალების მაქსიმალურ მობილიზებას მოითხოვდა. ვინაიდან ადამიანი ფლობდა აღნიშნული ძალების შეზღუდულ რესურსებს, ამიტომ მისი ძალისხმევა ყოველთვის იყო მიმართული აღნიშნული რესურსების ეფექტური გამოყენებისაკენ; თავდაპირველად იგი ფიზიკური ძალების მაქსიმალურად ეფექტური გამოყენების პრობლემის გადაწყვეტით იყო დაკავებული; ამ მიზნით იგი ქმნიდა ისეთ სხვადასხვა სახის სამარჯვებსა და იარაღებს, რომლებიც მას ფიზიკური ძალის მაქსიმალურად დაზოგვის საშუალებას აძლევდა.

ამ მიმართულებით რევოლუციური მნიშვნელობა ჰქონდა ინგლისელი გამომგონებლის **ჯეიმზ უატის** (1736 – 1819) მიერ ორთქლის მანქანის გამოგონებას, რომელმაც ადამიანს საშუალება მისცა ხელით შრომა შეეცვალა მანქანური შრომით.

მანქანურ შრომაზე მასობრივად გადასვლამ **XIX** საუკუნის პირველ ნახევარში შრომის მწარმოებლურობის ნახტომისებური ამაღლება გამოიწვია და ამ ფაქტს **სამრეწველო რევოლუცია** ეწოდა.

სამრეწველო რევოლიციის შედეგად ხელის შრომაზე დაფუძნებული მანუფაქტურული წარმოების მსხვილ მანქანურ წარმოებად გარდაქმნის პროცესს **ინდუსტრიალიზაცია** (ლათ. **industria** – ''საქმიანობა'') ეწოდება. **ინდუსტრიალიზაცია** გულისხმობს მსხვილი მანქანური წარმოებისა და ქვეყნის მატერიალურ-ტექნიკური ბაზის შექმნას. მსოფლიოს

მოწინავე ქვეყნებში ინდუსტრიალიზაციის პროცესი ძირითადად გასული საუკუნის პირველ ნახევარში დასრულდა, რის შედეგადაც ჩამოყალიბდა **ინდუსტრიული საზოგადოება.**

ლ.ვ. კანტოროვიჩი
(1912– 1986)

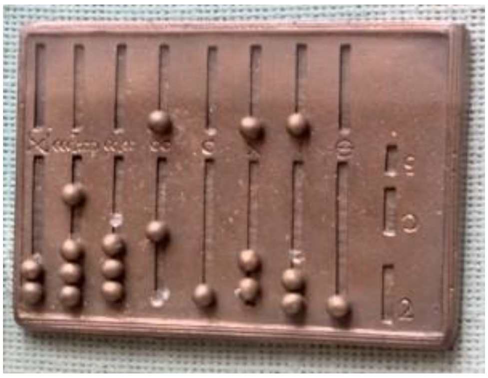
ტ. ჩ. კუპმანსი
(1810 – 1985)

6 ინდუსტრიალიზაციამ მინიმუმამდე შეამცირა შრომის პროცესში ადამიანის მიერ დასახარჯი ფიზიკური ძალის წილი, მაგრამ სათავე დაუდო გონებრივი ძალის წილის სწრაფ ზრდას. ასე მაგალითად, მეოცე საუკუნის დასაწყისამდე შრომის პროცესის ნორმალურად ორგანიზებისათვის საჭირო გამოთვლითი სამუშაოების შესასრულებლად ჩვეულებრივი საანგარიშოები, არითმომეტრები ან უმარტივესი კლავიშურ-პერფორაციული მანქანები თუ იყო საკმარისი, აღნიშნული პერიოდის შემდეგ ცხადი გახდა, რომ ისინი ამ სამუშაოებს უკვე ოპტიმალურად ვერ ასრულებდა. ამის კლასიკური მაგალითია საბჭოთა აკადემიკოს **ლეონიდ ვიტალის ძე კანტოროვიჩის** მიერ *1939* წელს გამოქვეყნებული ნაშრომი "**წარმოების ორგანიზაციისა და დაგეგმვის მათემატიკური მეთოდები**". მასში მოცემული მეთოდები წარმოების ოპტიმალური მართვის საშუალებას იძლეოდა, მაგრამ ისინი ისეთი დიდი მოცულობის მათემატიკური გამოთვლების ჩატარებას მოითხოვდა, რომ დიდი ხნის განმავლობაში მათი პრაქტიკაში გამოყენება შეუძლებელი იყო. აღნიშნული მეთოდების გამოყენებით ამერიკელმა მეცნიერმა **ტაილინგ ჩარლზ კუპმანსმა (Tjalling Chartes Koopmans)** დაამუშავა **წრფივი დაგეგმარების** სახელწოდების **კომპიუტერული მეთოდი**, რომელმაც რევოლუციური როლი შესარულა წარმოების შრომის ნაყოფიერების ამაღლებაში და **1975 წელს** ორივე მეცნიერს **ნობელის პრემია მიენიჭა "რესურსების ოპტიმალურად განაწილების თეორიაში შეტანილი ღვაწლისათვის".**

7 ზემოთ მოყვანილი მაგალითიდან ჩანს, თუ როგორი რევოლუციური როლი ითამაშა კომპიუტერმა კაცობრიობის განვითარების ისტორიაში. **ჯეიმზ უატის** ორთქლის მანქანამ თუ წარმოების ინდუსტრიალიზაციას დაუდო სათავე, კომპიუტერის გამოჩენით დაიწყო ინდუსტრიული საზოგადოებიდან საინფორმაციო საზოგადოებაზე გადასვლის პროცესი. აღნიშნულ პროცესს, **ინფორმატიზაციის პროცესი** ეწოდა. ზოგადად:

ინფორმატიზაცია წარმოადგენს ტერიტორიულად განაწილებული საინფორმაციო რესურსების გამაერთიანებელ პოლიტიკასა და პროცესს, რომლებიც სატელეკომუნიკაციო ინფრასტრუქტურის აგებისა და განვითარებისკენაა მიმართული.

კომპიუტერი მიეკუთვნება უნივერსალური გამომთვლელი მანქანების ჯგუფს, რომლებიც, ცარიელ ადგილზე არ შექმნილა. თვლის უნარის მქონე მოწყობილობების შექმნას დიდი ხნის ისტორია აქვს. ჩვენთვის ცნობილ პირველ მთვლელ მოწყობილობას წარმოადგენს **აბაკი** (ძვ. ბერძ. **αβαξ, ἀβάκιον**, ლათ. **abacus** – "დაფა"), ანუ **"სათვლელი დაფა"** რომელიც ჩვენს წელთაღრიცხვამდე **V** საუკუნეში ძველ საბერძნეთსა და რომში გამოიყენებოდა არითმეტიკული ოპერაციების შესასრულებლად. შემდეგ აღნიშნული დაფა დავიწყებული იქნა და დაახლოებით **X** საუკუნეში იგი აღადგინა და თვლის ათობითი სისტემის გამოყენების საფუძველზე გააუმჯობესა **ჰერბერტ ავრილელმა** (შემდგომში რომის პაპი **სილვესტრ II**)**;** სწორედ მოსკოვის ბიბლიოთეკაში აღმოჩენილი ავრილიელის ხელნაწერის გაცნობის სურვილი დაასახელა ცნობილი მწერლის **მ.ბულგაკოვის** ***(1891 – 1940)*** რომანის **"ოსტატისა და მარგალიტას"** მთავარმა მოქმედმა პირმა **ვოლანდმა** მოსკოვში მისი ჩამოსვლის მიზეზად. აღნიშნული დაფის გამოყენებით **ჰერბერტი** მრავალნიშნა რიცხვებზე არითმეტიკულ მოქმედებებს უსწრაფესად ასრულებდა, რის გამოც მას თანამედროვეები ჯადოქრობის უნარის მქონედ მიიჩნევდნენ.



რომისეული რეკონსტრუირებული
ა ბ ა კ ი

ჰერბერტ ავრილელი
(≈ 946 – 1003)

აბაკის შემდგომი მოდიფიცირება **XV** საუკუნისა და **XVI** საუკუნის დასაწყისში მოახდინა **ლეონარდო და ვინჩმა. 1642** წელს **19** წლის ფრანგმა მათემატიკოსმა და ფიზიკოსმა **ბლეზ პასკალმა** შექმნა **"პასკალინას"** სახელწოდებით ცნობილი ამჯამავი მანქანა**,** რომელსაც შეკრებისა და გამოკლების ოპერაციების შესრულება შეეძლო. პასკალის მანქანა წარმოადგენდა ერთმანეთთან დაკავშირებული მრავალი კბილის მქონე მექანიკურ მოწყობილობას. შესაკრები რიცხვები მანქანაში ამკრეფი თვლების სათანადოდ შებრუნების გზით შეიტანებოდა. თვლების რაოდენობა შესატანი ათობითი რიცხვის თანრიგების რაოდენობას უდრიდა; თითოეულ თვალს შეესაბამებოდა გარკვეული თანრიგი და მასთან დატანილი იყო **0**-დან **9**-მდე ციფრებით დანომრილი დანაყოფი. რიცხვის შესატანად თვალი საჭირო იყო სათანადო ციფრამდე შეგვეტრიალებინა.

შეკრების მათემატიკური ოპერაციის ავტომატური შემსრულებელი პირველი მოწყობილობა მექანიკური საათის ბაზაზე იყო შექმნილი. ასეთი მექანიკური მანქანა პირველად **1623 წელს** აღწერა ტიუბინგეის უნივერსიტეტის პროფესორმა **ვ.შიკარდმა**; დამზადებული იყო ერთადერთი ასეთი მანქანა, რომელიც **6**-ნიშნა რიცხვებზე არითმეტიკული ოპერაციების შესასრულებლად იყო განკუთვნილი. **შიკარდის მანქანა** შედგებოდა აჯამვის, გამრავლებისა და რიცხვების ჩამწერ სამი დამოუკიდებელი მოწყობილობისაგან; შეკრებისას – შესაკრები რიცხვები, ხოლო გამოკლებისას – საკლები და მაკლები, თანამიმდევრობით

შეიტანებოდა უშუალოდ ოპერაციის შესრულების წინ. გამრავლების ოპერაციის შესასრულებლად გამოიყენებოდა ცხრილური (ცხავური) გამრავლების მეთოდი. შიკარდის მანქანაში გამოყენებული პრინციპული სქემა კლასიკურ სქემად იქნა მიჩნეული; იგი (ან მისი მოდიფიკაცია) მომდევნო მთვლელი მანქანების უმრავლესობაში მანამ გამოიყენებოდა, სანამ მექანიკური დეტალები ელექტრომექანიკური დეტალებით არ იქნა შეცვლილი.

ლეონარდო და ვინჩი
(1452 – 1519)

ლეონარდო და ვინჩის აბაკი

1694 წელს გერმანელმა მათემატიკოსმა და ფილოსოფმა **გოტფრიდ ვილჰელმ ლეიბნიცმა** გამოიგონა ოთხივე არითმეტიკული ოპერაციის შემსრულებელი **არითმომეტრი** (ბერძ. **arithmos** – ‘‘რიცხვი’’, ‘‘თვლა’’ და **metro** – ‘‘ზომა’’, ‘‘გამზომი’’). მისი მთავარი ნაწილი იყო ე.წ. საფეხურებრივი ლილვაკი – სხვადასხვა სიგრძის მქონე კბილებიანი ცილინდრი, რომელიც მთვლელ თვალზე ზემოქმედებდა. ლილვაკის გასწვრივი გადაადგილებით შეიძლებოდა საჭირო რაოდენობის კბილებთან მისი მოდება, რისი მეშვეობითაც ხდებოდა საჭირო ციფრის დაყენება. ლაიბნიცის არითმეტიკული მანქანა არსებითად წარმოადგენდა მსოფლიოში პირველ არითმომეტრს, რომელიც საშუალებას იძლეოდა **8**-თანრიგიანი სამრავლისა და **9**-თანრიგიანი მამრავლის გადამრავლებით ფორმირებულიყო **16**-თანრიგიანი ნამრავლი. პასკალის მანქანისაგან განსხვავებით მასში გამოყენებული იყო პრინციპულად ახალი გამომთვლელი მოწყობილობა, რომელიც მნიშვნელოვნად ამცირებდა გამრავლებისა და გაყოფის შესრულებისათვის საჭირო დროს. აღნიშნულ არითმომეტრს ჰქონდა მთელი რიგი ნაკლოვანებებიც, რომლებიც შემდგომში გამოსწორდა. ყველაზე დიდი პოპულარობა მოიპოვა **პეტერბურგში (რუსეთი)** მოღვაწე **შვედი** ინჟინრის **ვ.გ. ოდნერის** მიერ **1877 წელს** კონსტრუირებულმა არითმომეტრმა.

8

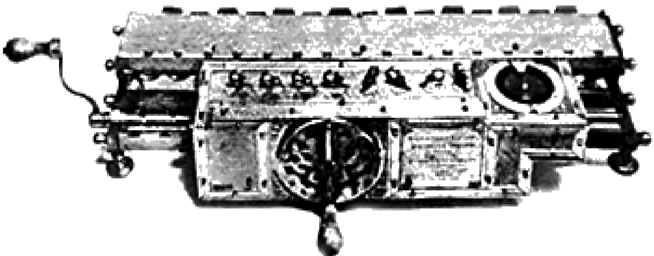
ზემოთ განხილულ მოწყობილობებში არითმეტიკული გამოთვლების შესასრულებლად **პროგრამული მართვა** არ გამოიყენებოდა. აღნიშნული მართვის გამოყენებით მექანიკური გამომთვლელი მანქანის შექმნის იდეა პირველად წამოაყენა და ამ იდეის რეალიზებას მიუძღვნა მთელი სიცოცხლე ინგლისელმა მათემატიკოსმა და გამომგონებელმა **ჩარლზ ბებიჯმა.** ასეთ მანქანას მან **ანალიზური მანქანა** უწოდა და იგი თანამედროვე კომპიუტერის მექანიკურ წინაპრადაა მიჩნეული. ელექტრომექანიკური საელემენტო ბაზის შეზღუდულმა ფუნქციონალურმა შესაძლებლობებმა **ჩარლზ ბებიჯს** საშუალება არ მისცა სრულად განეხორციელებინა თავისი ჩანაფიქრი, მაგრამ ამან ვერ დაჩრდილა მის მიერ გაწეული ტიტანური შრომა და იგი სამართლიანად ითვლება პროგრამული გამომთვლელი მანქანების შექმნის ფუძემდებლად. მისი იდეით იყო შეპყრობილი ცნობილი პოეტის **ჯორჯ ბაირონის** ქალიშვილი **ადა ლავლეისი,** რომელმაც ზემოთ აღნიშნული ანალიზური მანქანისათვის დაწერა პროგრამა და ეს იყო

მსოფლიოში შექმნილი **პირველი პროგრამა;** ასე რომ, **ადა ლავლეისი** ითვლება მსოფლიოში **პირველ დამპროგრამებელად.**

ბლეზ პასკალი
(1623– 1662)

გოტფრიდ ვილჰელმ ლაიბნიცი
(1646– 1716)

"პასკალინა"



ლეიბნიცის არითმომეტრი

ვილჰელმ შიკარდი
(1592– 1635)

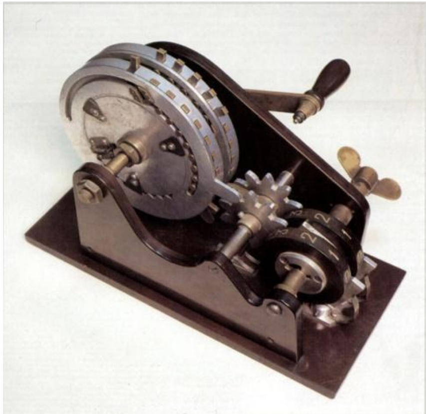
შიკარდის მანქანა

9

გამოთვლითი მანქანის ფუნქციონირებისათვის პროგრამული მართვის ბებიჯისეული იდეა პირველად იქნა რეალიზებული ჰარვარდის უნივერსიტეტის ახალგაზრდა მათემატიკოსის **ჰოვარდ ეიკენის** მიერ. მისი ხელმძღვანელობით მომუშავე ინჟინერთა ჯგუფმა **IBM**-თან დადებული კონტრაქტის შესაბამისად **1941** წელს ააწყო დაპროგრამებადი გამომთვლელი მანქანა. მისი შემოკლებული სახელი იყო **Mark I**, რაც გაშლილად იკითხება როგორც **"ავტომატური მიმდევრობებით მართვადი გამომთვლელი" ("Automatic Sequence Controlled Cflculator").** პირველი ტესტირების წარმატებით გავლის შემდეგ იგი **1946** წელს ჰარვარდის უნივერსიტეტში

გადაიტანეს და იქ იქნა ფორმალურად ამუშავებული. მოცემული კომპიუტერი შეიცავდა **765000** დეტალს (ელექტრომაგნიტურ რელეებს, გადამრთველებს და ა.შ.), იწონიდა დაახლოებით **4,5** ტონას, მისი სიგრძე იყო თითქმის **17** მეტრი, სიმაღლე – **2,5** მეტრი, ხოლო მასში გამოყენებული შემაერთებელი სადენების სიგრძე თითქმის **800** კილომეტრს აღწევდა. ძირითადი გამომთვლელი მოდულები სინქრონიზირდებოდა **15** მეტრიანი ლილვით, რომლის ბრუნვასაც უზრუნველყოფდა **5** ცხენის ძალის (**4** კილოვატის) მქონე ელექტრული ძრავა. ფაქტობრივად **Mark I გაუმჯობესებულ არითმომეტრს** წარმოადგენდა, რომელიც ჩვეულებრივი სახელურიანი ხელსაწყოების გამომყენებელი დაახლოებით **20** ოპერატორის შრომას ცვლიდა, მაგრამ ვინაიდან მას დაპროგრამების უნარი ჰქონდა, ამიტომ ხშირად მას რეალურად მოქმედ **კომპიუტერსაც უწოდებენ**.

ვ.ტ. ოდნერი
(1845– 1905)



ოდნერის არითმომეტრი

10

მკაცრად თუ ვიმსჯელებთ, პირველ ავტომატურ დაპროგრამებად უნივერსალურ ციფრულ კომპიუტერს წარმოადგენს გერმანელი მეცნიერის **კონრად ცუზეს** მიერ **1939-1941** წლებში დამუშავებული მანქანა **Z3**. იგი იმართებოდა ნახმარი კინოფირისაგან დამზადებული **პერფობარათით** და მასში შეტანა-გამოტანის ოპერაციები ოთხი ღილაკისაგან შემდგარი ციფრული კლავიატურით სრულდებოდა; აღნიშნული კომპიუტერი შეიცავდა დაახლოებით **2600** რელეს, რომელთაგანაც **1400** რელეს საშუალებით წარმოიქმნებოდა მეხსიერების მოდული, **600** რელესაგან – არითმეტიკული მოდული და დანარჩენი რელეებისაგან – მართვის სქემები. პრაქტიკულად დამზადებული იქნა ერთადერთი ასეთი კომპიუტერი და იგი **1944** წელს განხორციელებული საჰაერო დაბომბვის დროს იქნა განადგურებული.

11

მსოფლიოში პირველი დიდი უნივერსალური ელექტრონული ციფრული კომპიუტერი **აშშ**-ის არმიის შეკვეთით ბალისტიკური კვლევების ცენტრში ამერიკელი მეცნიერების **ჯ. მოჩლისა და ჯ. ეკერტის** ხელმძღვანელობით იქნა აგებული. მისი სახელწოდებაა **ENIAC** (**E**lectronic **N**umberical **I**ntergrator and **C**omputer) და იგი **1946** წლის **14** თებერვალს იქნა ამუშავებული.

კომპიუტერი შეიცავდა **17468** ელექტრონულ მილაკს, **7200** კრისტალურ დიოდს, **4100** მაგნიტურ ელემენტს და იკავებდა **300** $მ^2$-ის ტოლ ფართს.

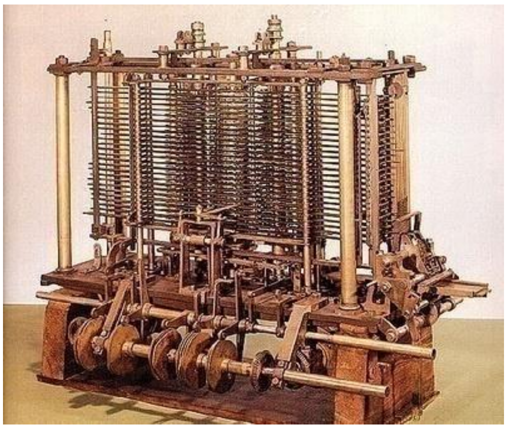
12

ინდუსტრიალიზაციის დაწყებას თუ სათავე დაუდო **ჯეიმზ უატის** ორთქლის ძრავის გამოჩენამ, **კომპიუტერის** გამოჩენა ინფორმატიზაციის დაწყების მიზეზი გახდა.

ინდუსტრიალიზაციის მიზანი იყო ადამიანის ფიზიკური საქმიანობის მაქსიმალური შემსუბუქება, **ინფორმატიზაციამ** კი ადამიანის გონებრივი საქმიანობის მაქსიმალურად გაიოლებას დაუდო სათავე. პირველი მათგანი კომფორტულს ხდიდა ადამიანის ფიზიკურ საქმიანობას და ზრდიდა ამ უკანასკნელის მარგი ქმედების კოეფიციენტს; მეორე მათგანი

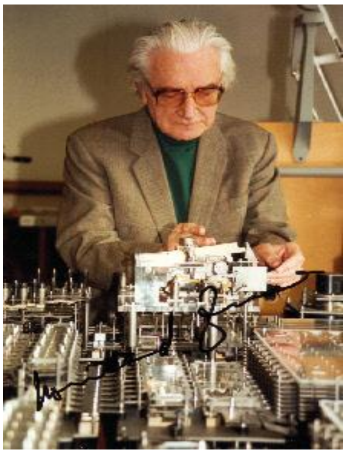
ადამიანის გონებრივი საქმიანობის კომფორტულობისა და მარგი ქმედების კოეფიციენტის ამაღლებაზეა ორიენტირებული.

ჩარლზ ბებიჯი
(1791– 1871)



ბებიჯის "ანალიზური მანქანა"

ადა ლავლეისი
(1815– 1852)



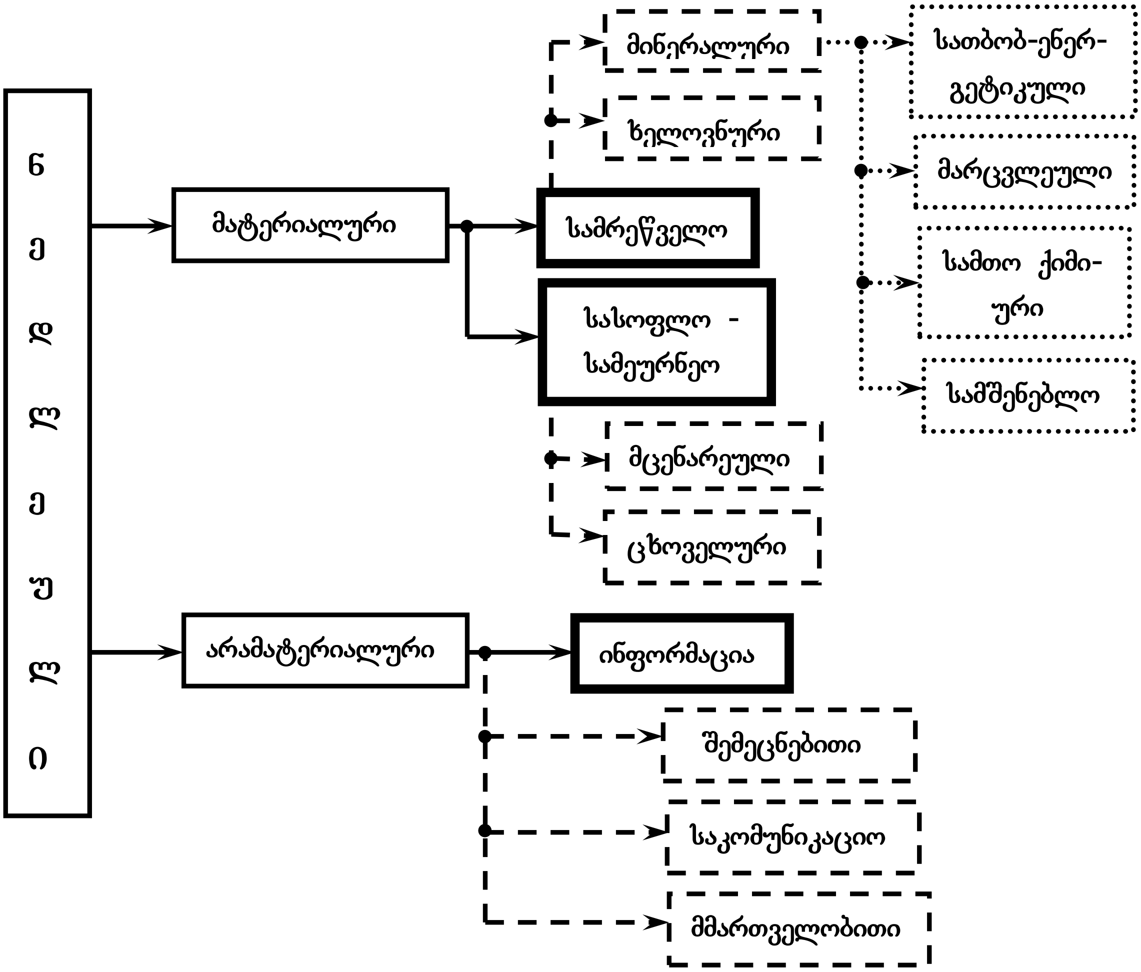
კონრად ცუზე
(1910– 1995)

ორთქლის ძრავის მეშვეობით შექმნილი მანქანებით ადამიანი გადაამუშავებს მატერიალურ ნედლეულს და გარკვეულ პროდუქციას იღებს. **კომპიუტერი** ადამიანს მის-თვის ხელმისაწვდომი ინფორმაციის გადამუშავებით ახალი ინფორმაციის მიღების საშუ-ალებას აძლევს. ანალოგიის პრინციპს თუ გამოვიყენებთ, საწყის, ანუ გადასამუშავებელ ინფორმაციას შეიძლება **ნედლეული,** ხოლო საბოლოო, ანუ გადამუშავების შედეგად მიღე-ბულ ინფორმაციას – **პროდუქტი** ვუწოდოთ. პირველ შემთხვევაში პროდუქტი თვი-სობრივად განსხვავდება ნედლეულისაგან, ხოლო მეორე შემთხვევაში – შინაარსობრივად.

შრომის საგანს, რომელმაც შრომის ზემოქმედებით გარკვეული ცვლილება განიცადა და შემდეგ გადამუშავებას საჭიროებს, **ნედლეული** ეწოდება. ინფორმატიზაციამდე მხო-ლოდ ორი სახის, კერძოდ სამრეწველო და სასოფლო-სამეურნეო, ნედლეულს განიხილავ-დნენ და ისინი (**ნახ.1.7**) მატერიალური ფორმის ნედლეულებს მიეკუთვნებოდნენ.

კომპიუტერის საშუალებით ინფორმაციის დამუშავების დაწყების შემდეგ ნედლეულის ზემოთ მოყვანილი განმარტების პირობებს ინფორმაციაც აკმაყოფილებს. მართლაც, კომპი-უტერისათვის მიწოდებამდე აუცილებელია ინფორმაციამ გარკვეული ცვლილება განი-ცადოს, კერძოდ, მიიღოს ისეთი ფორმა, რომელიც აუცილებელია კომპიუტერის მიერ

მისი აღქმისათვის. ასეთი სახის ინფორმაცია შემდგომში კომპიუტერის საშუალებით უნდა გადამუშავდეს, რათა მიიღოს პროდუქტის, ანუ საბოლოო ინფორმაციის სახე.



ნახ. 1.7. ბუნებაში არსებული ნედლეულის კლასიფიკაცია

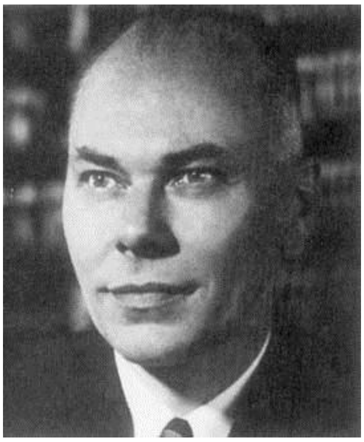
ზემოთ აღნიშნულიდან გამომდინარე, ინფორმატიზაციის შედეგად წარმოიშვა ნედლეულის კიდევ ერთი ნაირსახეობა – ინფორმაცია, რომელიც არამატერიალური ფორმის ნედლეულს წარმოადგენს (იხ. ნახ. **1.7**).

13

ხელით წარმოებაზე დაფუძნებულ წვრილ ხელით წარმოებას **ხელოსნობა** ეწოდება. გოტინგემის (**გერმანია**) უნივერსიტეტის ფილოსოფიის პროფესორმა **იოჰან ბეკმანმა 1772 წელს**, ე.ი. ინდუსტრიალიზაციის დაწყებამდე დიდი ხნით ადრე, ხელოსნობის პრობლემების შემსწავლელი მეცნიერების სახელწოდებისათვის შემოიღო ტერმინი **ტექნოლოგია** (ძვ. ბერძ. **τεχνη** - “ოსტატობა”, **λογοζ** - “აზრი”, “მეთოდიკა”). შემდგომში აღნიშული ტერმინის შინაარსი გაფართოვდა და ზოგადად წარმოების პრობლემების შემსწავლელი მეცნიერების აღმნიშვნელ ტერმინად გარდაიქმნა.

ხელით წარმოებიდან სამანქანო წარმოებაზე გადასვლის, ე.ი. ინდუსტრიალიზაციის შემდეგ **ტექნოლოგია** ეწოდა ნედლეულის, მასალის, ნახევფარფაბრიკატების ან ნაკეთობების მიღება-დამუშავების საშუალებებისა და წესების ერთობლიობას, რომელიც ხორციელდება მრეწველობის სხვადასხვა დარგში, მშენებლობაში და ა.შ. იგი წარმოადგენს სამეცნიერო დისციპლინას, რომელიც ამუშავებს და სრულყოფს ასეთ საშუალებებსა და წესებს. ამ პერიოდში ჩამოყალიბდა ტექნოლოგიების ისეთი სახეობები, როგორებიცაა მანქა-

ნათმშენებლობის ტექნოლოგია, მშენებლობის ტექნოლოგია, ლითონთა ტექნოლოგია, ქსოვილთა ტექნოლოგია, ქიმიური ტექნოლოგია და ა.შ.



ჰოვარდ ეიკენი
(1900– 1973)

კომპიუტერი **Mark I**

ჯ. მოჩლი (1907–1980)
ჯ. ეკერტი (1919-1995)

კომპიუტერი ENIAC თანამედროვე კომპიუტერის ფონზე

იოჰან ბეკმანი
(1739– 1811)

ინდუსტრიული საზოგადოებისთვის დამახასიათებელი ტექნოლოგიის ძირითადი კომპონენტებია: მატერიალური ნედლეულისა და მასალების მომზადება, მატერიალური პროდუქტის მომზადება, მომხმარებლებისათვის ნაწარმოები პროდუქტების მიწოდება. ასეთი სახის ტექნოლოგიას **მატერიალური ტექნოლოგია** ვუწოდოთ.

ინფორმატიზაციის შედეგად ტექნოლოგიათა ზემოთ ჩამოთვლილი კლასი შეივსო **საინფორმაციო ტექნოლოგიის** ცნებით. საინფორმაციო ტექნოლოლოგიაში საწყის მასალად (ნედლეულად) ინფორმაცია გამოიყენება; ამ დროს საბოლოო პროდუქტიც ინფორმაციაა, მაგრამ იგი წარმოადგენს ობიექტის, პროცესის ან მოვლენის შესახებ თვისობრივად ახალ ინფორმაციას. საინფორმაციო ტექნოლოგიის კომპონენტებია: მონაცემების (პირველადი

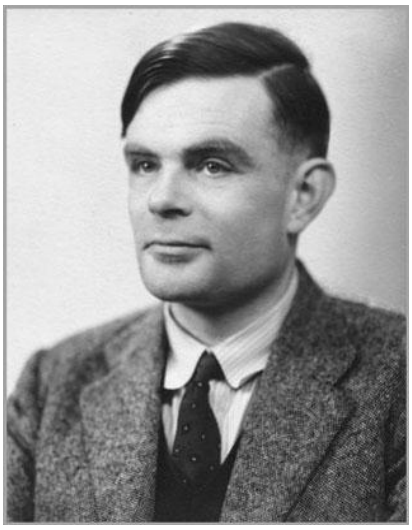
ინფორმაციის) შეკრება, მონაცემების დამუშავება, შედეგობრივი ინფორმაციის მიღება და მომხმარებლებისათვის მისი გადაცემა. ზოგადად:

საინფორმაციო ტექნოლოგია ეწოდება დისციპლინების ფართო კლასს და ადამიანის საქმიანობის სფეროებს, რომლებიც დაკავებულია **მონაცემების** შექმნის, შენახვის, მართვისა და დამუშავების პრობლემების გადაწყვეტით, რომლის დროსაც შეიძლება გამოთვლითი ტექნიკაც იქნეს გამოყენებული.

უკანასკნელ პერიოდში ტერმინ ‘‘საინფორმაციო ტექნოლოგიის’’ ნაცვლად ხშირად იყენებენ ტერმინს **კომპიუტერული ტექნოლოგია.**

კომპიუტერული ტექნოლოგია წარმოადგენს თეორიული და პრაქტიკული ცოდნის ერთობლიობას, რომელსაც გამოთვლითი ტექნიკის, დაპროგრამების, საინფორმაციო სისტემებისა და ტექნნოლოგიების სფეროში დასაქმებული სპეციალისტები იყენებენ თავიანთ საქმიანობაში.

ხშირად კომპიუტერულ ტექნოლოგიებს **კომპიუტერულ მეცნიერებებსაც (Computer Sciense)** უწოდებენ და როგორც სამეცნიერო დისციპლინა იგი **XX** საუკნის **30**-იანი წლების შუა პერიოდში ალგორითმების თეორიისა და მათემატიკური ლოგიკის შერწყმით წარმოიშვა. მას სათავე ელექტრონული გამომთვლელი მანქანების (კომპიუტერების) გამოგონებამ დაუდო.



ალან ტიურინგი
(1912– 1954)

კომპიუტერული მეცნიერების სფეროში გამოქვეყნებულ პირველ სამეცნიერო ნაშრომად ითვლება ინგლისელი მათემატიკოსის, ლოგიკოსისა და კრიპტოგრაფის, ბრიტანეთის იმპერიის ორდენის კავალერის **ალან ტიურინგის** მიერ **1936** წელს გამოქვეყნებული ცნობილი სამეცნიერო სტატია “**On Computable Numbers, with an Application to the Entscheidungsproblem",** რომელშიც შემოთავაზებულ ჰიპოთეკურ გამომთვლელ მანქანას შემდგომში სამეცნიერო ლიტერატურაში **ტიურინგის მანქანის** სახელწოდებით მოიხსენიებენ.

14

XX საუკუნის შუა წლებში ტერმინების ‘‘**საინფორმაციო ტექნოლოგიისა”** ’’(**Information technology**) და **“კომპიუტერული მეცნიერების”** (*computer science*) ნაცვლად შემოღებული იქნა ახალი ტერმინი **ინფორმატიკა**. იგი ნაწარმოები იქნა სიტყვების **ინფორმაცია** (**information**) და **ავტომატიკა** (**automatiqe**) ურთიერთშერწყმით (***ინფორმატიკა*** = ***ინფორ***მაცია + ავტო-***მატიკა***). ინფორმატიკისადმი მიძღვნილ ჩემთვის ხელმისაწვდომ თითქმის ყველა სახელმძღვანელოში აღნიშნული ტერმინის წარმომშობ ქვეყანად საფრანგეთი, ხოლო წარმოშობის თარიღად XX საუკუნის **60**-იანი წლების შუა პერიოდია მიჩნეული. აღნიშნული მოსაზრების საწინააღმდეგოდ ინტერნეტში მუშაობისას აღმოვაჩინე, რომ **გერმანელმა** მეცნიერმა **კარლ სტეინბუხმა** ჯერ კიდევ **1957** წელს გამოაქვეყნა მონოგრაფია “**Informatik: automatische informations ver arbeitung**” (‘‘ინფორმტიკა: ინფორმაციის ავტომატური დამუშავება’’) და ეს წარმოადგენს პირველ ნაშრომს, რომელშიც გამოჩნდა ტერმინი “**ინფორმატიკა**’’. აღნიშნულიდან გამომდინარე თავს ნებას ვაძლევ არ დავეთანხმო მიღებულ შეხედულებას და მოცემული ტერმინის ავტორად გერმანელი მეცნიერი **კარლ სტეინბუხი** მივიჩნიო.

ინფორმატიკის პრობლემებისათვის მიძღვნილი საერთაშორისო კონგრესის (1978) მიერ შემოთავაზებული იქნა ინფორმატიკის განსაზღვრება. "**ინფორმატიკა**" წარმოადგენს ინფორმაციის დამამუშავებელი სისტემების დაპროექტების, აგების, გამოყენებისა და მატერიალურ-ტექნიკური მომსახურების საკითხების მომცველ ცნებას, რომელშიც განიხილება როგორც მანქანები, მოწყობილობები, მათემატიკური უზრუნველყოფა და ორგანიზაციული საკითხები, ასევე სამრეწველო, კომერციული, ადმინისტრაციული და სოციალური ზემოქმედების მთელი კომპლექსი.

კარლ სტეინბუხი
(Karl Steinbuch)
(1917 – 2005)

ცნობილმა რუსმა აკადემიკოსებმა **ა. პ. ერშოვმა და ბ.ნ. ნაუმოვმა** ინფორმატიკა შემდეგნაირად განსაზღვრეს:

ინფორმატიკა წარმოადგენს საბუნებისმეტყველო მეცნიერებას, რომელიც შეისწავლის ინფორმაციის ზოგად თვისებებს, აგრეთვე მისი დამუშავების (შეკრების, შენახვის, გარდაქმნის, გადანაცვლებისა და გაცემის) პროცესებს, მეთოდებსა და საშუალებებს.

15

მატერიალური ტექნოლოგიით გათვალისწინებული პრობლემების გადაწყვეტით ადამიანთა შეზღუდული რაოდენობის ჯგუფები იყო დაკავებული; რაც შეეხება **საინფორმაციო ტექნოლოგიას**, იგი მეთოდურად იკავებს წამყვან ადგილს საზოგადოების საქმიანობის უკლებლივ ყველა, მათ შორის **საპასუხისმგებლო გადაწყვეტილების** მიღებასთან დაკავშირებულ სფეროებშიც; აღნიშნულის გამო სასიცოცხლო მნიშვნელობა აქვს ზუსტად იქნეს დადგენილი საზოგადოების ცხოვრებაზე საინფორმაციო ტექნოლოგიების ზემოქმედებათა შესაძლო შედეგების სრული ჩამონათვალი. ეს უაქტიურესი პრობლემაა, რადგან უკვე ცნობილია ის ფაქტი, რომ აღნიშნულ ზემოქმედებას პოზიტიურის გარდა შეიძლება **ნეგატიური ხასიათიც** ჰქონდეს. ასე, მაგალითად:

- კომპიუტერები სულ უფრო და უფრო ფართოდ გამოიყენება ისეთ სისტემებში (მაგალითად, ატომურ რეაქტორებში, შეიარაღებაში და ა.შ.) მიმდინარე ტექნიკური პროცესების მართვისათვის, რომელთა პარამეტრების რეგლამენტირებულ ფარგლებს გარეთ გასვლამ შეიძლება არა მარტო მსხვილი ავარიები, არამედ გლობალური კატასტროფებიც გამოიწვიოს. ასეთი პროცესების მართვის სისტემებისათვის დამახასიათებელ სპეციფიკურ თავისებურებას წარმოადგენს ის გარემოება, რომ მათ დროის უმცირეს მონაკვეთებში დიდი მოცულობის ურთულესი ინფორმაციის დამუშავება უნდა უზრუნველყოს. აღნიშნულის გამო, **ჯერ ერთი,** განუწყვეტლივ იზრდება სისტემების სირთულე **და, მეორეც,** აღნიშნულ სისტემაში გამოყენებული მრავალი მილიონი სამანქანო ბრძანებისაგან შემდგარ პროგრამულ უზრუნველყოფაში შეცდომების არარსებობის გარანტია პრაქტიკულად შეუძლებელია. აღნიშნულ შეცდომებს შეიძლება დაემატოს:

– აპარატურის ამოვარდნები ან მტყუნებები,
– პერსონალის პროვოკაციული ან დივერსიული მოქმედებები,
– ელექტრონული ვირუსებით კომპიუტერების დასნებოვნება და ა.შ.

ადვილი წარმოსადგენია სამხედრო დანიშნულების სისტემებში ასეთი შემთხვევების შედეგები. სხვა სიტყვებით, თანამედროვე პირობებში არა მარტო ინფორმაციის დამუშა-

ვების სისტემები უნდა დავიცვათ გარემოს ზემოქმედებისაგან, არამედ გარემოცაა დასაცავი აღნიშნული სისტემის მიერ დამუშავებული ინფორმაციის ზემოქმედებისაგან. უზრუნველყოფილი უნდა იყოს არა მარტო სისტემაში შემავალი, დაგროვილი და გადამუშავებადი ინფორმაციის უსაფრთხოება, არამედ გარემოს საინფორმაციო უსაფრთხოებაც.

- საქმიანობის მრავალ სფეროში კომპიუტერების გამოყენება ადამიანის უფლებების დარღვევის პოტენციურ შესაძლებლობას მკვეთრად ზრდის, რადგან ახალი საინფორმაციო ტექნოლოგიების მასობრივი დანერგვა აფართოებს ადამიანებზე დოსიების შედგენის, მათი სატელეფონო საუბრების მოსმენის, ელექტრონული ფოსტის არასანქცირებული წაკითხვის, ანაბრების გაკონტროლების, კომპიუტერული თვალთვალის და ა.შ. შესაძლებლობას;
- იზრდება საავტორო უფლებისა და საკუთრების (განსაკუთრებით კომპიუტერული პროდუქტების საკუთრების) უფლების დარღვევის საფრთხე;
- ინფორმატიზაცია შეიძლება სოციალური დაძაბულების წყაროც გახდეს და ა.შ.

საინფორმაციო ტექნოლოგიების ზემოთ ჩამოთვლილი და სხვა შესაძლო ნეგატიური ასპექტები მხედველობაში უნდა მივიღოთ ინფორმატიზაციის ამოცანების გადაწყვეტის დროს. ეს პრობლემები აუცილებლად უნდა იყოს თანამედროვე ინფორმატიკის შესწავლის საგანი.

1.3. კლასიფიკაციის სახეები და ინფორმაციის კლასიფიცირება

სამყაროში არსებული ცოცხალი და არაცოცხალი ბუნების მრავალმილიარდიანი წარმომადგენლების მიერ ურთიერთასახვის პროცესში უსასრულო რაოდენობის სხვადასხვა სახისა და ფორმის ინფორმაციები წარმოიშვება. ისინი ჩვენს აღქმაზე ზემომქმედი სპეციფიკური გამაღიზიანებლებია, რომლებიც ახდენენ ჩვენი აღქმის გადატვირთვას. აღნიშნული გამაღიზიანებლების (ინფორმაციების) გააზრებისათვის ჩვენ ვახდენთ ზოგადი მახასიათებლების მიხედვით ცალკეულ ჯგუფებად მათ დაყოფას, ანუ კლასიფიცირებას.

კლასიფიკაცია (ლათ. *classis* – თანრიგი და *facere* - ვაკეთებ) ეწოდება საგნების (მოვლენების, ცნებების) ერთობლიობის დაყოფას კლასებად, ჯგუფებად, თანრიგებად რაიმე ნიშან-თვისების მიხედვით.

1.8 ნახაზის თანახმად განასხვავებენ კლასიფიკაციის იერარქიულ, ფასეტურ და დესკრიპტორულ სისტემებს.

<u>კლასიფიკაციის იერარქიული სისტემის</u> გამოყენებისას ითვლება, რომ ობიექტების საწყისი სიმრავლე (რომლის კლასიფიცირებაც ანუ დაყოფა უნდა მოხდეს) მოთავსებულია კლასიფიკაციის **ნულოვან დონეზე;** იგი იყოფა შერჩეული საკლასიფიკაციო ნიშნის მიხედვით კლასებად, რომლებიც თავსდება კლასიფიკაციის **პირველ დონეზე;** შემდეგ პირველ დონეზე არსებული თითოეული კლასი საკუთარი საკვალიფიკაციო ნიშნების მიხედვით იყოფა ქვეკლასებად. ამ დროს წარმოშობილი ყველა ქვეკლასი თავსდება კლასიფიკაციის **მეორე დონეზე** და ა.შ. კლასიფიკაციის იერარქიულ სისტემას აქვს ე.წ. **''ხისმაგვარი სტრუქტურა''**, რომელშიც ''ხის ძირი ანუ ტანია'' კლასიფიკაციის ნულოვანი დონე, ხოლო ხის ტანიდან გამომავალი ტოტები კი – კლასიფიკაციის პირველი, მეორე და ა.შ. დონეები.

სისტემის ღირსებებია აგების სიმარტივე და იერარქიული სტრუქტურის სხვადასხვა ტოტებზე დამოუკიდებელი საკვალიფიკაციო ნიშნის გამოყენების შესაძლებლობა,

ხოლო **ნაკლია –** აგების სტრუქტურის სიხისტე, რაც ართულებს სტრუქტურაში რაიმე ცვლილების შეტანას.

კ ლ ა ს ი ფ ი კ ა ც ი ა

1. კლასიფიკაციის **იერარ-ქიული** სისტემა

2. კლასიფიკაციის **ფასე-ტური** სისტემა

3.კლასიფიკაციის **დესკრი-პტორული (აღწერითი)** ს ი ს ტ ე მ ა

ნახ.1.8. კლასიფიკაციის სახეები

<u>კლასიფიკაციის ფასეტური სისტემა</u> წარმოადგენს ცალკეული ჯგუფების (ფასეტების) ნაკრებს, რომელთაგანაც თითოეული მათგანი შეიცავს ერთტიპურ ობიექ-ტებს, მაგალითად ‘‘ვერცხლის მონეტების ქისა’’, ’’ძვირფასეულობის ზარდახშა’’, ‘‘ხორა-გიანი ხურჯინი’’, ‘‘პროდუქტებიანი მაცივარი’’ და ა.შ. კლასიფიკაციის წინა სისტემისაგან განსხვავებით მოცემული სისტემა საშუალებას გვაძლევს კლასიფიკაციის ნიშნებად წოდე-ბული **ფასეტები** (ფრანგ. ***facette*** – ‘‘მიჯნა’’, ‘‘ზღვარი’’) ავირჩიოთ ერთმანეთისაგან და-მოუკიდებლად და კლასიფიცირებადი ობიექტების აზრობრივი შინაარსისაგან დამოუკი-დებლად.

სისტემის ღირსებებია სისტემის მარტივად მოდიფიცირებისა და კლასიფიცირების დიდი რაოდენობის ნიშნების გამოყენების შესაძლებლობა, ხოლო **ნაკლია** – დიდი რაოდე-ნობის ობიექტების არსებობის დროს მისი აგების სირთულე.

<u>კლასიფიკაციის დესკრიპტორული (აღწერითი) სისტემა</u> წარმო-ადგენს ზემოთ განხილული სისტემების გაერთიანებას და ითვალისწინებს ისეთი **დეს-კრიპტორების** (საკვანძო სიტყვებისა და სიტყვათაშეთანხმებების) ლექსიკონის შექმნას**,** რომელთა შორისაც შეიძლება დამყარდეს:

სინონიმური კავშირები, რომლებიც გვიჩვენებს საკვანძო სიტყვების, როგორც სინონი-მების, გარკვეულ ერთობლიობას, მაგალითად: სტუდენტი – მოწაფე – მსწავლელი (ისევე როგორც კლასიფიკაციის ფასეტური სისტემის დროს);

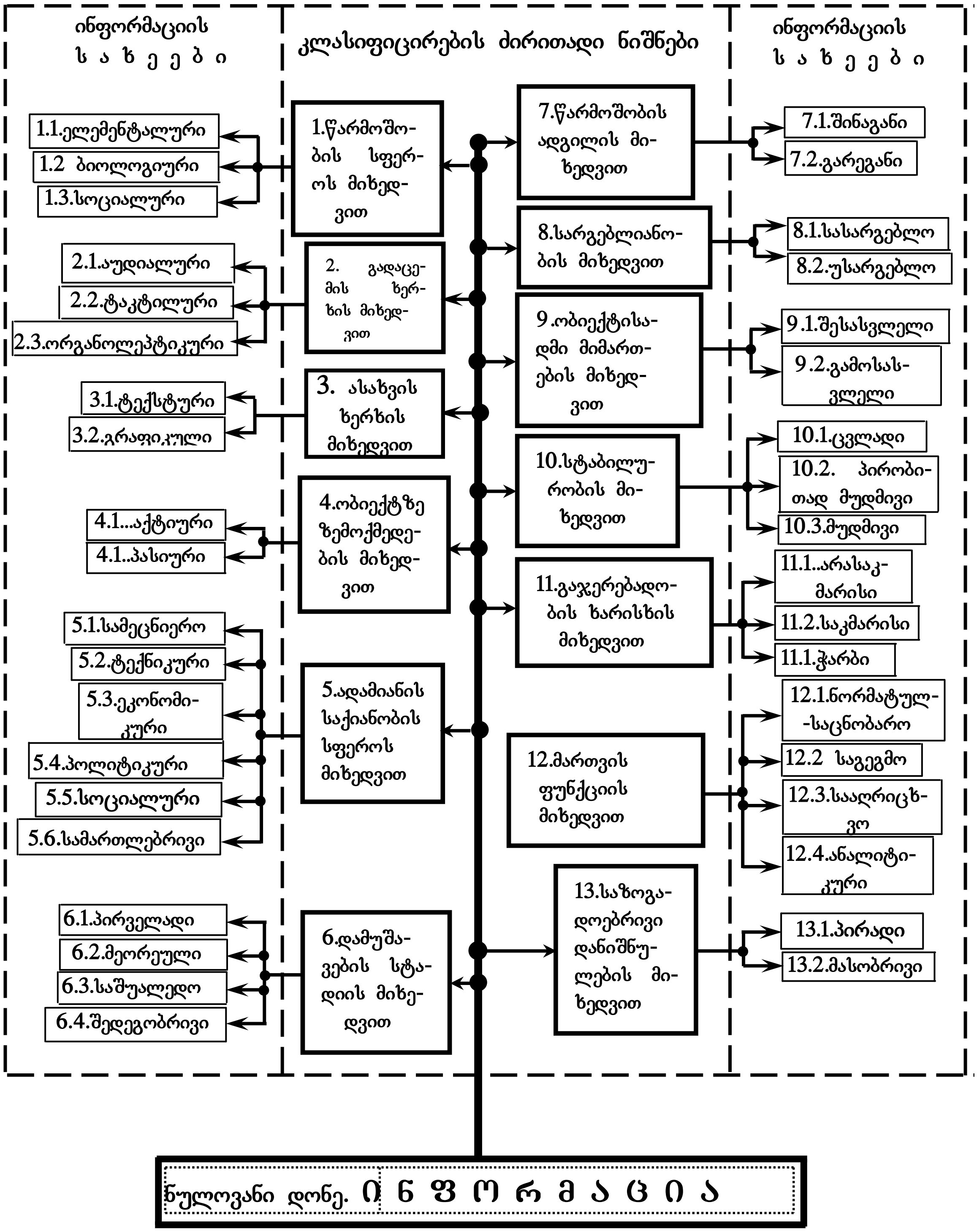
გვარსახეობრივი კავშირები, რომლის დროსაც ობიექტების გარკვეული კლასი შედის უფრო წარმომადგენლობით კლასში, მაგალითად: უნივერსიტეტი – ფაკულტეტი – კა-თედრა (ისევე როგორც კლასიფიკაციის იერარქიული სისტემის დროს);

ასოციატური კავშირები, რომლის დროსაც ერთიანდება საერთო თვისებების მქონე დესკრიპტორები, მაგალითად: სტუდენტი – გამოცდა – უნივერსიტეტი.

ინფორმაციათა კლასიფიცირებისათვის გამოიყენება **შერეული სისტემა**, რომლის დროსაც მთელი ინფორმაცია რამდენიმე სხვადასხვა ნიშნის მიხედვით კლასიფიცირდება.

2 დღეისათვის ინფორმაცია საზოგადოების ერთ-ერთი ძირითადი რესურსია და შემდეგი ძირითადი ნიშნებით კლასიფიცირდება (**ნახ.1.9**):

1. წარმოშობის სფეროს მიხედვით განასხვავებენ ელემენტალურ, ბიოლოგიურ და სოციალურ ინფორმაციებს. **ელემენტალური ინფორმაცია** ასახავს არაცოცხალ მოვლენებსა და პროცესებს, **ბიოლოგიური ინფორმაცია** – ცხოველური და მცენარიული სამყაროს პროცესებს, ხოლო **სოციალური ინფორმაცია** – ადამიანთა საზო-გადოების მოვლენებს.



ნახ. 1.9. ინფორმაციის კლასიფიკაციის სტრუქტურული სქემა

2. გადაცემის ხერხის მიხედვით განასხვავებენ აუდიალურ, ტაქტილურ და ორგანოლეპტიკურ ინფორმაციებს. **აუდიალური ინფორმაცია** არის სმენის ორგანოებით აღქმადი, **ტაქტილური ინფორმაცია** (ლათ. *tactilis* – "შეხება") – შეგრძნებებით (შეხებით) აღქმადი, ხოლო **ორგანოლეპტიკური ინფორმაცია** – გემოთი და სურნელებით აღქმადი ინფორმაცია;

3. ასახვის ხერხის მიხედვით განასხვავებენ **ტექსტურსა** და **გრაფიკულ ინფორმაციებს**;

4. ობიექტზე ზემოქმედების მიხედვით განასხვავებენ აქტიურსა და პასიურ ინფორმაციებს. **აქტიური ინფორმაცია** ეწოდება უშუალოდ ობიექტზე ზემოქმედ ინფორმაციას, ხოლო **პასიური ინფორმაცია** კი ისეთ ინფორმაციას, რომელიც ასეთ ზემოქმედებას არ ახდენს;

5. ადამიანის საქმიანობის სფეროს მიხედვით განასხვავებენ **სამეცნიერო, ტექნიკურ, ეკონომიკურ, პოლიტიკურ,სოციალურსა და სამართლებრივ ინფორმაციებს;**

6. დამუშავების სტადიის მიხედვით განასხვავებენ პირველად, მეორეულ, საშუალედო და შედეგობრივ ინფორმაციებს. **პირველადი ინფორმაცია** უშუალოდ ობიექტის მოქმედების პროცესში წარმოშობილი და საწყის სტადიაში რეგისტრირებული ინფორმაციაა; **მეორეული ინფორმაცია** მიიღება პირველადი ინფორმაციის დამუშავების შედეგად (იგი შეიძლება წარმოადგენდეს საშუალედო ან შედეგობრივ ინფორმაციას, ე.ი. ეს ორი ინფორმაცია მეორეული ინფორმაციის ნაირსახეობებია); **საშუალედო ინფორმაცია** შემდგომ გამოთვლებში საწყის მონაცემებად გამოსაყენებელი ინფორმაციაა; **შედეგობრივი ინფორმაცია** წარმოადგენს პირველადი და საშუალედო ინფორმაციების დამუშავების პროცესში წარმოშობილი ისეთი ინფორმაციაა, რომელიც უშუალოდ მმართველობითი გადაწყვეტილების გამომუშავებისათვის გამოიყენება;

7. წარმოშობის ადგილის მიხედვით განასხვავებენ შინაგან და გარეგან ინფორმაციებს; **შინაგანი ინფორმაცია** ობიექტის შიგნით, ხოლო **გარეგანი ინფორმაცია** ობიექტის ფარგლებს გარეთ წარმოშობილი ინფორმაციაა;

8. სარგებლიანობის მიხედვით განასხვავებენ **სასარგებლო და უსარგებლო ინფორმაციებს;**

9. ობიექტისადმი მიმართების მიხედვით განასხვავებენ შესასვლელ და გამოსასვლელ ინფორმაციებს. **შესასვლელი ინფორმაცია** ობიექტისათვის მიწოდებული ინფორმაციაა, ხოლო **გამოსასვლელი ინფორმაცია** წარმოადგენს ისეთ ინფორმაციას, რომელსაც მოცემული ობიექტი აწოდებს მეორე ობიექტს;

10. სტაბილურობის მიხედვით განასხვავებენ ცვლად, პირობითად მუდმივ და მუდმივ ინფორმაციებს. **ცვლადი (მიმდინარე) ინფორმაცია** ხშირად ცვლადი, **პირობითად მუდმივი ინფორმაცია –** იშვიათად ცვლადი, ხოლო **მუდმივი ინფორმაცია** – არასდროს ცვლადი ინფორმაციაა;

11. გაჯერებადობის ხარისხის მიხედვით განასხვავებენ **არასაკმარის, საკმარისსა და ჭარბ ინფორმაციებს;**

12. მართვის ფუნქციის მიხედვით განასხვავებენ ნორმატულ-საცნობარო, საგეგმო, სააღრიცხვო და ანალიტიკურ ინფორმაციებს. **ნორმატულ-საცნობარო ინფორმაცია** შეიცავს სხვადასხვა ნორმატულ და საცნობარო მონაცემებს; **საგეგმო ინფორმაცია** ასახავს ობიექტის სამომავლო პარამეტრებს; **სააღრიცხვო ინფორმაცია** ახასიათებს მართვის ობიექტის მოქმედებას დროის გარკვეულ წარსულ პერიოდში; **ანალიტიკური ინფორმაცია** წარმოადგენს ინფორმაციას, რომელიც წარმოიშვება დაწესებულების სამეწარმეო-სამეურნეო და ფინანსური საქმიანობის მაჩვენებლების შეფასების პროცესში;

13. საზოგადოებრივი დანიშნულების მიხედვით განასხვავებენ **მასობრივ და პირად ინფორმაციებს.** უნდა შევნიშნოთ, რომ მასობრივ ინფორმაციას წარმოადგენს ზემოთ ჩამოთვლილი ეკონომიკური, პოლიტიკური, სოციალური და ა.შ. ინფორმაციები.

ინფორმაციის განხილული კლასიფიკაცია სრულყოფილად ვერ ჩაითვლება, რადგან სამყაროში არსებულ ინფორმაციათა მრავალფეროვნება და მათი კლასიფიცირების მიზნების სწრაფად ცვლებადობა დამატებითი საკლასიფიკაციო ნიშნების შემოტანის წინაპირობებს ქმნის.

1.4. ინფორმაციის სტრუქტურული ერთეულები

ინფორმაციას გააჩნია რთული სტრუქტურული იერარქია, რომელიც განსაზღვრავს მის აგებულებას და გამოყოფს ამ აგებულების ელემენტებს – **სტრუქტურულ საინფორმაციო ერთეულებს.** უმარტივეს სტრუქტურულ საინფორმაციო ერთეულს წარმოადგენს ე.წ. **რეკვიზიტი**; რეკვიზიტები ხატოვნად შეიძლება შევადაროთ ''საინფორმაციო ატომებს'' რომელთა ურთიერთდაკავშირებით წარმოიშვება ისეთი შედგენილი სტრუქტურული საინფორმაციო ერთეულები, როგორებიცაა **მაჩვენებელი, დოკუმენტი, მასივი, ნაკადი** და **საინფორმაციო ბაზა.** მოკლედ განვიხილოთ ზემოთ ჩამოთვლილი უმარტივესი და შედგენილი სტრუქტურული საინფორმაციო ერთეულები.

რეკვიზიტები წარმოადგენს ლოგიკურად დაუყოფად ელემენტალურ სტრუქტურულ საინფორმაციო ერთეულებს, რომელიც ასახავს ობიექტის, პროცესის, მოვლენის განსაზღვრულ თვისებებს. ერთმანეთისაგან განასხვავებენ **რეკვიზიტ-ნიშანსა** და **რეკვიზიტ-ფუძეს.**

რეკვიზიტ-ნიშანი ახასიათებს ობიექტის ან პროცესის თვისობრივ მხარეს (ასეთებია, მაგალითად, მასალის დასახელება, გაზომვის ერთეული, მიმწოდებლის დასახელება და ა.შ.), ხოლო **რეკვიზიტ-ფუძე** ახასიათებს ობიექტის, პროცესის ან მოვლენის რაოდენობრივ მხარეს (მიღებული მასალის რაოდენობას).

რეკვიზიტ-ფუძეების დამუშავების დროს სრულდება **არითმეტიკული ოპერაციები,** ხოლო რეკვიზიტ-ნიშნების დამუშავების დროს – **ლოგიკური ოპერაციები** (დაჯგუფება, დახარისხება და ა.შ.).

ცნება **"რეკვიზიტის"** სინონიმებია: ***სიტყვა, ელემენტი, ატრიბუტი.***

ცალკე აღებულ რეკვიზიტ-ფუძეებსა და რეკვიზიტ-ნიშნებს არ აქვს ეკონომიკური აზრი, ამიტომ მათ შეხამებულად იყენებენ.

მაჩვენებელი ეწოდება ერთი რეკვიზიტ-ფუძესა და ერთი ან რამდენიმე რეკვიზიტ-ნიშნის შემცველ შედგენილ საინფორმაციო სტრუქტურულ საინფორმაციო ერთეულს.

მაჩვენებელი ერთი მხრივ **რთული ერთეულია,** რომელიც ახასიათებს ობიექტის რაოდენობრივ და თვისობრივ მხარეებს, ხოლო მეორე მხრივ **მინიმალური შედგენილი საინფორმაციო ერთეულია,** რომელსაც აქვს ინფორმატიულობა და ამიტომ შეუძლია წარმოქმნას დოკუმენტი.

დოკუმენტი ეწოდება ერთი ან რამდენიმე მაჩვენებლის ერთობლიობას.

მასივი წარმოადგენს სხვადასხვა ერთგვაროვან დოკუმენტებში არსებული ინფორმაციების ერთობლიობას.

ნაკადი ეწოდება მართვის ერთი ფუნქციასთან დაკავშირებულ მასივების ნაკრებს.

საინფორმაციო ბაზა წარმოადგენს მთლიანად მმართველობითი სამუშაოს მახასიათებელი ნაკადების ერთობლობას.

საინფორმაციო ბაზის რესურსების გამოსათვლელად აუცილებელია რაოდენობრივად იქნეს შეფასებული ინფორმაცია. ინფორმაციის რაოდენობრივი შეფასების საკითხებს გან-

ვიხილავთ მეორე თავში, ახლა კი შევნიშნავთ, რომ ამ მიზნით შემოტანილია **ბიტის, ბაიტის** და მათგან ნაწარმოები ტერმინები.

მეორე თავში დავინახავთ, რომ სპეციალური მეთოდიკების გამოყენებით ინფორმაცია შეიძლება შეფასდეს როგორც რაოდენობრივად, ასევე თვისობრივად. ასეთ შეფასებებს დიდი მნიშვნელობა აქვს ინფორმაციის შენახვის ორგანიზაციისა და მისი დამუშავების ტექნოლოგიის დონეებზე.

1.5. საზოგადოების ინფორმატიზაციის საკითხები

მიღებული განსაზღვრების თანახმად **ინფორმატიზაცია** წარმოადგენს ტერიტორიულად განაწილებული **საინფორმაციო რესურსების** გაერთიანების პოლიტიკასა და ამ პოლიტიკასთან დაკავშირებულ პროცესებს (**იხ. გვ.19**).

საინფორმაციო რესურსების გაერთიანებით დაკავებულია სოციუმი (საზოგადოება), რომელმაც ამ მიზნის მისაღწევად საჭიროა გამოიყენოს გარკვეული სახის საინფორმაციო ტექნოლოგიები. ამისათვის საზოგადოების ცალკეული წევრების საქმიანობის სხვადასხვა სფეროში აუცილებელია დაინერგოს სხვადასხვა სახის საინფორმაციო ტექნოლოგიები.

ადამიანის საქმიანობის სხვადასხვა სფეროში საინფორმაციო ტექნოლოგიების დანერგვას ეწოდება **საზოგადოების ინფორმატიზაცია**.

ინფორმატიზაციის ცნების განსაზღვრებაში საკვანძოა ტერმინი ''საინფორმაციო რესურსები''. განვსაზღვროთ იგი:

საინფორმაციო რესურსები ეწოდება:

1) უტყუარი ინფორმაციის ეფექტურად მიღებისათვის ორგანიზებული მონაცემების ერთობლიობას;

2) ცალკეულ დოკუმენტებსა და დოკუმენტების ცალკეულ მასივებს;

3) საინფორმაციო სისტემებში (არქივებში, ფონდებში, მონაცემების ბანკებსა და სხვა საინფორმაციო სისტემებში) დაცულ დოკუმენტებსა და დოკუმენტების მასივებს.

უკანასკნელ განსაზღვრებაში გამოყენებული ტერმინი **მონაცემები** ფორმალიზებული (კოდირებული) სახით წარმოდგენილი ფაქტები და ცნობებია (**იხ. გვ.14**), ე.ი. სპეციფიკური ფორმის ინფორმაციას წარმოადგენს.

მოყვანილი განსაზღვრებების ანალიზი გვიჩვენებს, რომ როგორც ''მონაცემები'', ასევე ''დოკუმენტები'' ინფორმაციების ნაირსახეობებია, რომელთა დამუშავებით არიან დაკავებული საზოგადოების ცალკეული წევრები.

ინფორმაციის დამუშავება ადამიანმა უძველესი დროიდან დაიწყო. მას საკუთარი არსებობის უზრუნველსაყოფად ოდითგანვე უხდებოდა ფიზიკური (მატერიალური) და გონებრივი (ინტელექტუალური) საქმიანობის ურთიერთშეხამება. მაგალითად, პირველყოფილი ადამიანის მიერ საკვები პროდუქტების მოპოვებას თუ ფიზიკური (მატერიალური) საქმიანობის თავისებურ ფორმად მივიჩნევთ, მოპოვებული სარჩოს ''სამართლიანად'' განაწილება და სათადარიგოდ შესანახი მარაგის რაოდენობის განსაზღვრა გონებრივი (ინტელექტუალური) საქმიანობის ჩანასახოვან ფორმად შეიძლება ჩაითვალოს. პირველ შემთხვევაში იგი თუ მატერიალურ რესურსებს ''ამუშავებდა'', მეორე შემთხვევაში - ინფორმაციის ''დამუშავებით'' იყო დაკავებული. ცხადია, რომ თავდაპირველად ადამიანი ძირითადად მატერიალური რესურსების დამუშავებით იყო დაკავებული და ინფორმაციის დამუშავებას, ანუ ''ინტელექტუალურ'' საქმიანობას უმნიშვნელო დროს უთმობდა.

ადამიანის ცხოვრებაში ''საინფორმაციო საქმიანობის'' წილის მოცულობა თანდათან და მდოვრედ (ადამიანის ''ინტელექტუალური'' განვითარების კვალობაზე) იზრდებოდა, მა-

გრამ მატერიალურ სფეროში მიღწეული ზოგიერთი კონკრეტული წარმატებები კარდინალურად ცვლიდა საინფორმაციო საქმიანობის ხასიათს და მთლიანად არსებული საზოგადოებრივი ურთიერთობების ნახტომისებური, ანუ რევოლუციური გარდაქმნის მიზეზი ხდებოდა.

საინფორმაციო რევოლუცია ეწოდება ინფორმაციის დამუშავების სფეროში კარდინალური ცვლილებების შედეგად საზოგადოებრივი ურთიერთობების ნახტომისებურად გარდაქმნას.

ცივილიზაციის განვითარების ისტორიაში დაფიქსირებულია ოთხი საინფორმაციო რევოლუცია. მოკლედ განვიხილოთ თითოეული მათგანი.

პირველი საინფორმაციო რევოლუცია დაკავშირებულია დამწერლობის წარმოშობასთან (ჩვენს წელთაღრიცხვამდე მესამე ათასწლეული). მისი მეშვეობით მომავალი თაობისათვის დაგროვილი ინფორმაციის გადაცემა გახდა შესაძლებელი. ამ რევოლუციის შემდგომ პერიოდში შეზღუდულად გამოიყენებოდა საინფორმაციო რესურსები, მონაცემების დამუშავება ხელით ხდებოდა და ამიტომ ცოდნამ წარმოებაზე არსებითი ზეგავლენა ვერ მოახდინა.

მეორე საინფორმაციო რევოლუციის (**XVII საუკუნე**) მიზეზი გახდა წიგნების ბეჭდვის დაწყება. მან მძლავრი ბიძგი მისცა კულტურული და საორგანიზაციო საქმიანობის განვითარებას. ამ რევოლუციის შემდეგ დაიწყო ცოდნის ტირაჟირება და მან წარმოებაზე გავლენის მოხდენას დაუდო სათავე.

მესამე საინფორმაციო რევოლუცია (**XIX საუკუნე**) ელექტროობის გამოგონებამ გამოიწვია, რომლის შედეგადაც გამოჩნდა ტელეგრაფი, ტელეფონი, რადიო და ტელევიზორი. ამ რევოლუციის შემდეგ ნებისმიერი მოცულობის ინფორმაციის ოპერატიულად გადაცემა გახადა შესაძლებელი.

მეოთხე საინფორმაციო რევოლუცია (**XX**საუკუნე) მიკროპროცესორის გამოგონებასთან არის დაკავშირებული. მიკროპროცესორების საფუძველზე კომპიუტერები (ელექტრონული გამომთვლელი მანქანები) იქნა დამუშავებული, რომლებიც შემდგომ კომპიუტერულ ქსელებად და მონაცემების გადამცემ სისტემებად (საინფორმაციო კომუნიკაციებად) გაერთიანდა. მოხდა საზოგადოების კომპიუტერიზაცია.

საზოგადოების კომპიუტერიზაცია ეწოდება ადამიანის საქმიანობის სხვადასხვა სფეროში ელექტრონული გამომთვლელი მანქანების (კომპიუტერების) დანერგვასა და განვითარებას.

საზოგადოების კომპიუტერიზაციამ საყოველთაო ინფორმატიზაცია გახადა შესაძლებელი. თანამედროვე კომპიუტერები ნებისმიერი ინფორმაციის დამუშავების შედეგების ოპერატიულად მიღებისა და დაგროვების შესაძლებლობას იძლევა.

საზოგადოების ინფორმატიზაციის დროს ძირითადი ყურადღება ექცევა ადამიანის ნებისმიერი სახის საქმიანობაში ჭეშმარიტი და ამომწურავი ცოდნის დროულად გამოყენების უზრუნველყოფისაკენ მიმართულ კომპლექსურ ღონისძიებებს.

ინფორმატიზაციის წარმატებულად რეალიზებისათვის სასურველია დავიცვათ თანამედროვე საზოგდოების მიერ დამუშავებული შემდეგი ზოგადი პრინციპები:

- მსოფლიოში მეცნიერებისა და ტექნიკის მიღწევების ფართოდ გამოყენება;
- ვაჭრობისა და სხვა მსგავსი დარგების ნაცვლად მეცნიერულად ტევადი დარგების (განსაკუთრებით საინფორმაციო სექტორის) პრიორიტეტული განვითარება;
- სახელმწიფო და კერძო ინფორმატიზაციაში მნიშვნელოვანი საფინანსო საშუალებების ჩადება.

2

ინფორმატიზაციის პროცესის შედეგად ყალიბდება საინფორმაციო საზოგადოება, რომელიც მანიპულირებს არა მარტო მატერიალური ობიექტებით, არამედ ცოდნითაც.

საინფორმაციო საზოგადოება ეწოდება თანამედროვე სოციუმს, რომელშიც ადამიანთა უმრავლესობა დაკავებულია ინფორმაციის წარმოებით, შენახვით, დამუშავებითა და გავრცელებით.

თანამედროვე სახელმწიფოში ბუნებრივი, ადამიანური და ტექნოლოგიური რესურსების გვერდით ყველაზე ძვირფას რესურსს წარმოადგენს ე.წ. **საინფორმაციო რესურსი.**

საინფორმაციო რესურსების დამუშავების საფუძველზე შესაძლებელია საინფორმაციო პროდუქტების მიღება.

საინფორმაციო პროდუქტი ეწოდება შემდგომი გავრცელებისათვის მწარმოებლის მიერ ფორმულირებული მონაცემების ერთობლიობას.

ისევე როგორც ტრადიციული სახის რესურსებისა და პროდუქტების გამოყენებისას, საინფორმაციო რესურსებისა და პროდუქტების გამოყენების დროს აუცილებელია ვიცოდეთ:

- სად ინახება საინფორმაციო რესურსები და პროდუქტები;
- რამდენად ხელმისაწვდომია ისინი;
- ვინა არის მათი მფლობელი;
- ვის ესაჭიროება ისინი;
- როგორია მათი ღირებულება.

ზემოთ ჩამოთვლილ კითხვებზე პასუხის მიღება შეიძლება **საინფორმაციო ბაზარზე,** რომელსაც **საინფორმაციო რესურსებისა და პროდუქტების ბაზარსაც** უწოდებენ.

საინფორმაციო ბაზარი წარმოადგენს ინტელექტუალური შრომის პროდუქტების ვაჭრობისათვის განკუთვნილ ეკონომიკურ, საორგანიზაციო და სამართლებრივ ურთიერთობათა სისტემას. ამ დროს პროდუქტებისა და მომსახურებების **მიმწოდებლებია** სათანადო რესურსების მფლობელი სხვადასხვა კომერციული ფირმები, ხოლო **მომხმარებლები** – საკუთარი ამოცანების გადამწყვეტი იურიდიული და ფიზიკური პირები.

საინფორმაციო ბაზარი ვერ იარსებებს სათანადო **ინფრასტრუქტურის** გარეშე, რომელსაც წარმოქმნის ამ ბაზრის მომსახურე სხვადასხვა სახელმწიფო და კერძო ორგანიზაციები.

საინფორმაციო ბაზარი პირობითად შემდეგ **ხუთ სექტორად** შეიძლება დაიყოს:

პირველი სექტორი განკუთვნილია **სპეციალური** (სამეცნიერო, ტექნიკური და პროფესიული) რესურსებისა და პროდუქტების გასაყიდად;

მეორე სექტორი - საქმიანი (სტატისტიკური, კომერციული და ფინანსური) რესურსებისა და პროდუქტების გასაყიდად;

მესამე სექტორი - სამომხმარებლო (ახალი ამბების, ლიტერატურის, გართობის) რესურსებისა და პროდუქტების გასაყიდად;

მეოთხე სექტორი - (საშუალო, უმაღლესი, გადასამზადებელი) **საგანმანათლებო** რესურსებისა და პროდუქტების გასაყიდად;

მეხუთე სექტორი - უზრუნველყოფისათვის განკუთვნილი საინფორმაციო საშუალებების (პროგრამული პროდუქტების, ტექნიკური საშუალებების, კონსულტირებისა და ა.შ.) გასაყიდად;

საინფორმაციო მომსახურების გასაწევად აუცილებელია კომპიუტერული ან არაკომპიუტერული სახის მონაცემთა ბაზის შექმნა. შეიძლება გამოვყოთ შემდეგი ექვსი სახის ასეთი მომსახურება:

- პირველადი წყაროს მიწოდება;

- საინფორმაციო ნაკეთობის გამოშვება;
- მონაცემთა ბაზაში ინფორმაციის მოძიება;
- მონაცემთა დაშორებულ ბაზაში დისტანციურად შეღწევა;
- ინფორმაციის დამუშავება (თარგმანების, მიმოხილვების დამზადება);
- პროგრამული უზრუნველყოფისა და საინფორმაციო ტექნოლოგიების დამუშავება.

თანამედროვე ადამიანი აღჭურვილი უნდა იყოს სათანადო ცოდნით და გააჩნდეს გარკვეული დონის საინფორმაციო კულტურა.

საინფორმაციო კულტურა გულისხმობს ინფორმაციასთან მიზანმიმართული მუშაობის უნარსა და ტექნიკური საშუალებების ბაზაზე დამუშავებული საინფორმაციო ტექნოლოგიების გამოყენებით აღნიშნული ინფორმაციის მიღების, დამუშავებისა და გადაცემის ცოდნას.

1.6. ავტომატიზებული საინფორმაციო სისტემის სტრუქტურული აგებულება

1 პირველ პარაგრაფში ფორმირებული განსაზღვრების თანახმად **ავტომატიზებული საინფორმაციო სისტემაში** მონაცემების დამუშავების სპეციალური მეთოდებისა და გამოთვლითი, საკომუნიკაციო და სხვა ტექნიკური საშუალებების კომპლექსის გამოყენების მეშვეობით, მართვის საინფორმაციო პროცესი ავტომატიზებულია გარკვეული **შედეგობრივი ინფორმაციის** მოსამზადებლად; ეს უკანასკნელი სჭირდება აღნიშნული სისტემის მომხმარებელ სპეციალისტს მისთვის დაკისრებული მართვის ფუნქციების შესასრულებლად და ამიტომ ავტომატიზებულმა საინფორმაციო სისტემამ უნდა უზრუნველყოს ამ ინფორმაციის მისთვის გადაცემა.

ზემოთ აღნიშნულიდან გამომდინარე, ავტომატიზებული საინფორმაციო სისტემა შეიცავს შემდეგ სამ ძირითად კომპონენტს (**ნახ.1.10**):

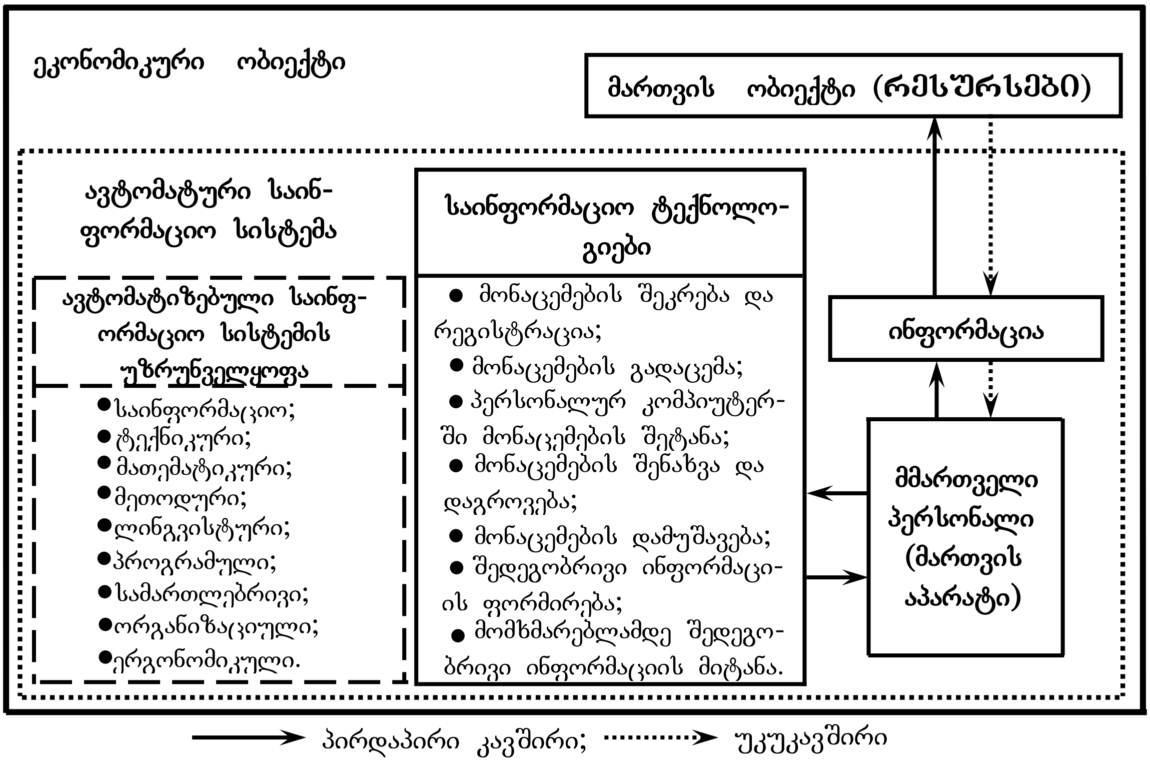
- **ინფორმაციას**, რომელიც წარმოადგენს მართვის სისტემის როგორც ნედლეულს (საგანს), ასევე პროდუქტს;
- **საინფორმაციო ტექნოლოგიას**, რომელიც მოიცავს ინფორმაციის დამუშავების საშუალებებსა და მეთოდებს;
- **პერსონალს**, რომელიც საინფორმაციო ტექნოლოგიის გამოყენებით ახდენს საინფორმაციო პროცესის რეალიზებას.

2 საინფორმაციო სისტემის შესასვლელზე არსებობს **პირველადი ინფორმაცია** მართვის ობიექტში შესასრულებელი ყველა ცვლილების შესახებ. იგი წარმოადგენს საინფორმაციო სისტემის **საგანს,** რომელიც ფიქსირდება ოპერატიული აღრიცხვის ფუნქციების შესრულების შედეგად. საინფორმაციო სისტემაში პირველადი ინფორმაცია (საინფორმაციო ტექნოლოგიის ნედლეული) გარდაიქმნება გადაწყვეტილების მისაღებად ვარგის ინფორმაციად, რომელსაც **შედეგობრივ ინფორმაციას** უწოდებენ. იგი წარმოადგენს საინფორმაციო სისტემის მიერ წარმოებულ **პროდუქტს.** შედეგობრივ ინფორმაციად პირველადი ინფორმაციის ფორმალურად გარდაქმნის პროცედურების ნაწილს ავტომატიზებულ საინფორმაციო სისტემაში წინასწარ დასახული ალგორითმების მიხედვით ტექნიკური საშუალებები ადამიანის უშუალოდ ჩარევის გარეშე ახდენს.

ზემოთ აღნიშნული არ ნიშნავს, რომ საინფორმაციო სისტემას შეუძლია მთლიანად ავტომატურ რეჟიმში იმუშაოს. მართვის სისტემის **მმართველი პერსონალი:**

- განსაზღვრავს როგორც პირველადი და შედეგობრივი ინფორმაციის შედგენილობასა და სტრუქტურას, ასევე პირველადი ინფორმაციის შეკრებისა და რეგისტრაციის წესს;
- აკონტროლებს პირველადი ინფორმაციის სისრულესა და უტყუარობას;

● განსაზღვრავს შედეგობრივ ინფორმაციად პირველადი ინფორმაციის გარდაქმნის შე-სრულების თანამიმდევრობას და აკონტროლებს გარდაქმნის პროცესის მიმდინარეობის სისწორეს.



ნახ. 1.10. ავტომატური საინფორმაციო სისტემის სტრუქტურა

გარდა ამისა, აღსანიშნავია, რომ დღემდე სუსტადაა ავტომატიზებული პირველადი ინ-ფორმაციის შეკრების პროცედურა და ამიტომ ტექნიკურ საშუალებებში იგი ხშირად მმართველ პერსონალს შეაქვს.

ინფორმაციის გარდამქმნელი ტექნიკური საშუალებების უმნიშვნელოვანესი ნაწილია კომპიუტერები, რომლებიც გარკვეული პროგრამების გამოყენებით მონაცემებს ავტომატუ-რად ამუშავებს.

თანამედროვე ავტომატიზებულ საინფორმაციო სისტემებში საინფორმაციო პროცესის პროცედურები **დეცენტრალიზებულია** და კომპიუტერთან მომხმარებლის დიალოგურ რე-ჟიმში მუშაობით სრულდება. ეს მომხმარებელს საშუალებას აძლევს აკონტროლოს მონა-ცემების გარდაქმნის პროცესი და ოპერატიულად მოახდინოს საჭირო კალაპოტით მისი მიმართვა. ამით განსხვავდები ისინი **დიდ ელექტრონულ გამომთვლელ მანქანებზე დაფუ-ძნებული ავტომატიზებული საინფორმაციო სისტემებისაგან,** რომლებშიც ინფორმაციის დამუშავების პროცესი **ცენტრალიზებულად** სრულდება და განცალკევებულია მმართველი პერსონალისაგან. ამ უკანასკნელს მხოლოდ საბოლოო შედეგები მიეწოდება. მას თუ აღ-ნიშნული შედეგები ამა თუ იმ მიზეზის (მაგალითად, საწყის მონაცემებში შეცდომების დაგვიანებით აღმოჩენის) გამო არ აწყობს, მაშინ იგი იძულებულია მიმართოს სათანადო სამსახურებს გაიმეორონ მისთვის საინტერესო ამოცანის გადაწყვეტის პროცესი.

ამგვარად, თანამედროვე ავტომატურ საინფორმაციო სისტემებში საინფორმაციო პრო-ცესის ავტომატიზებულად შესასრულებელი პროცედურები მართვის ფუნქციებთან არის ინტეგრირებული. საკუთარ ძირითად ფუნქციებთან ერთად ამ პროცედურებს უშუალოდ მმართველი პერსონალი ასრულებს. **უფრო მეტიც,** მმართველ სპეციალისტს, იყენებს რა პროფესიული კომპიუტერული მომზადების არმქონე მომხმარებელზე ორიენტირებული საინსტრუმენტო პროგრამულ საშუალებებს, ხშირად თავად შეუძლია მოახდინოს მონაცე-

მების მისთვის აუცილებელი დამუშავების პროცედურების ავტომატიზება, დასვას ამოცანა და შესარულოს დამპროგრამებლის როლი.

დასასრულს შევნიშნავთ, რომ თანამედროვე გაგებით ტერმინი ‘‘საინფორმაციო სისტემა’’ საინფორმაციო პროცესების ავტომატიზებას გულისხმობს. ამიტომ უკანასკნელ პერიოდში ცნებებს **‘‘საინფორმაციო სისტემა’’** და **‘‘ავტომატიზებული საინფორმაციო სისტემა’’** ტოლძალოვან ცნებებად მიიჩნევენ, თუმცა უნდა გვახსოვდეს, რომ საინფორმაციო სისტემებში შეიძლება ინფორმაციის არაავტომატიზებული დამუშავების ტექნოლოგიაც იყოს გამოყენებული.

3 ავტომატიზებულ საინფორმაციო სისტემებში ინფორმაციის გადამუშავების პროცესი თავისი არსით მატერიალურ წარმოებაში გამოყენებული ტექნოლოგიური პროცესის ანალოგიურია. ამ უკანასკნელის შესასრულებლად რადგან სხვადასხვა ტექნიკური საშუალებები (ჩარხები, მოწყობილობები, ხელსაწყოები და ა.შ.) არის გამოყენებული, ამიტომ აღნიშნული მოწყობილობების ანალოგური საშუალებებისა და მეთოდების გამოყენებაა საჭირო ინფორმაციის გადამუშავების პროცესის რეალიზებისთვისაც.

ინფორმაციის გარდაქმნის სხვადასხვა ეტაპებზე სხვადასხვა **საინფორმაციო ტექნოლოგიები** ანუ საინფორმაციო პროცედურების შესრულების სხვადასხვა საშუალებები და მეთოდები გამოიყენება. ამგვარად საინფორმაციო ტექნოლოგიისათვის დამახასიათებელია შემდეგი ორი მდგენელი:

● **ტექნოლოგიური პროცესი**, რომელიც განსაზღვრავს ინფორმაციის გარდაქმნის ეტაპებსა და პროცედურებს;

● **საშუალებებისა და მეთოდების ერთობლიობა,** რომლებიც საინფორმაციო პროცესის პროცედურების შესრულების დროს გამოიყენება.

საინფორმაციო პროცესის რეალებისათვის შეიძლება სხვადასხვა საშუალებები და მეთოდები იქნეს გამოყენებული. მათი სახით განისაზღვრება საინფორმაციო ტექნოლოგიის განვითარების დონე. თანამედროვე საინფორმაციო ტექნოლოგიებისათვის დამახასიათებელია შემდეგი თავისებურებებია:

● სპეციალისტები, როგორც საბოლოო მომხმარებლები, საინფორმაციო პროცესში აქტიურად უშუალოდ კომპიუტერებით აღჭურვილ სამუშაო ადგილებიდან იღებენ მონაწილეობას;

● კომპიუტერული ქსელების არეში შესაძლებელია **დოკუმენტების კოლექტიური დამუშავება;**

● გამოიყენება **ერთიანი საინფორმაციო ბაზა** და მომხმარებელთა ფართო წრის თითოეულ წარმომადგენელს მისთვის მინიჭებული უფლებამოსილების ფარგლებში შეუძლია აღნიშნულ ბაზაში შეღწევა;

● მუშაობის **ინტერაქტიული რეჟიმის არსებობა**, რომელიც ინფორმაციის დამუშავების პროცესში ჩარევისა და საჭიროების შემთხვევაში ამოცანების გადაწყვეტის პროცესის მიმდინარეობის შეცვლის შესაძლებლობის საშუალებას იძლევა;

● **მეგობრული ხასიათის სამომხმარებლო ინტერფეისის არსებობას**, კომფორტული მუშაობის უზრუნველსაყოფად **მომხმარებელს** სთავაზობენ სპეციალურად შედგენილ მენიუს, მისთვის ორგანიზებულია მოკარნახის სისტემა, მას შეუძლია დაუბრუნდეს უკვე შესრულებელ ქმედებებს და ახლებურად შეასრულოს ისინი და ა.შ.

● წარმოშობილი ამოცანების გადაწყვეტების დროს გამოყენებული **სხვადასხვა პროგრამული კომპლექსების** (ტექსტური რედაქტორების, ცხრილური პროცესორების, სპეციალიზებული ეკონომიკური პროგრამებისა და ა.შ.) **ინტეგრირება;**

● გამოყენების კონკრეტული პირობებისადმი **უნიფიცირებული სისტემების ადაპტაცია** (შეწყობა, შეგუება).

4

არსებული მრავალფეროვანი საინფორმაციო ტექნოლოგიებიდან გამოყოფენ შემდეგი ორი სახის საინფორმაციო ტექნოლოგიას:

1. **უზრუნველმყოფ საინფორმაციო ტექნოლოგიებს,** რომლებიც ნებისმიერი ამოცანის გადაწყვეტის დროს **ინსტრუმენტარიუმად** გამოიყენება. ისინი საკმაოდ მრავალფეროვანია და ორიენტირებულია **უმარტივესი** (მაგალითად, ტექსტური პროცესორებით დოკუმენტების დამუშავების) **ამოცანებიდან** დაწყებული, **ურთულესი** (მაგალითად, ექსპერტულ სისტემებში გადაწყვეტის მიღების) **ამოცანებით** დამთავრებული სხვადასხვა კლასის ამოცანებზე.

ინსტრუმენტარიუმი ეწოდება: **1.** გარკვეული სახის ინსტრუმენტების ნაკრებს, რაიმე სპეციალობაში გამოყენებული ინსტრუმენტების ერთობლიობას (მაგალითად, ქირურგიული ინსტრუმენტარიუმი) **2.** (გადატანით) რაიმე მიზნის მისაღწევად ან რაიმის განსახორციელებლად გამოყენებული საშუალებების ერთობლიობას (მაგალითად, მმართველობითი საბუღალ-ტრო აღრიცხვის ინსტრუმენტარიუმი).

2. **ფუნქციონალურ საინფორმაციო ტექნოლოგიებს,** რომლებიც კონკრეტულ საგნობრივ სფეროში ახდენს ამოცანების გადაწყვეტის ტექნოლოგიის რეალიზებას რამდენიმე სხვადასხვა უზრუნველმყოფი ტექნოლოგიის კომბინაციის გამოყენების მეშვეობით. უზრუნველმყოფი ტექნოლოგიების სირთულეებზე დამოკიდებულებით ისინი განკუთვნილია ან მხოლოდ ავტომატიზებული საინფორმაციო სისტემების დამპროექტებელი სპეციალისტებისათვის, ან მათი გამოყენება შეუძლიათ მმართველობითი აპარატის ფართო წრის წარმომადგენლებსაც (რომლებიც კომპიუტერული ტექნიკის სფეროში არაპრფესიონალები არიან). ასე მაგალითად, მონაცემების დამუშავების მრავალი სისტემა შეიცავს: **საშუალებებს,** რომლებიც დამპროგრამებლებს პროგრამათა საგნობრივად-ორიენტიერებული სისტემების შექმნის საშუალებას აძლევს და **ინსტრუმენტებს,** რომლის დახმარებითაც არაპროფესიონალ მომხმარებელს (მაგალითად, ეკონომისტს) მონაცემების საკუთარი ბაზის შექმნა შეუძლია.

რაიმე საგნობრივი სფეროს ავტომატიზების დროს სხვადასხვა პროგრამული საშუალებებით რეალიზებული რამდენიმე უზრუნველმყოფი საინფორმაციო ტექნოლოგიის ერთდროულად გამოყენება ხდება საჭირო. მაგალითად, საჭირო ხდება ბუღალტრული აღრიცხვის პროგრამით ფორმულირებული მონაცემების ექსპორტირება ეკონომიკური ანალიზის ჩასატარებლად საჭირო ცხრილურ პროცესორში ან ფინანსური ანალიზის სპეციალურ პროგრამაში. ამის გამო განსაკუთრებულ მნიშვნელობას იძენს გამოყენებული პროგრამული არეების ინტეგრირება (შეპირაპირება).

ავტომატიზებულ საინფორმაციო სისტემასა და საინფორმაციო ტექნოლოგიას შორის არსებობს შემდეგი ორი განსხვავება:

1. განსხვავებული მიზნები. ავტომატიზებული საინფორმაციო სისტემის ძირითადი მიზანია მიღებული იქნეს შედეგობრივი ინფორმაცია და იგი მიეწოდოს მმართველ მუშაკს, რათა ამ უკანასკნელმა შეძლოს გარკვეული გადაწყვეტილების მიღება. რაც შეეხება საინფორმაციო ტექნოლოგიას, მისი მიზანია სათანადო საშუალებებისა და საინფორმაციო პროცესის პროცედურათა რეალიზაციის მეთოდების გამოყენებით უზრუნველყოს ინფორმაციის დამუშავების ზუსტად განსაზღვრული მოქმედებების შესრულება.

2. განსხვავებული სტრუქტურული თანაფარდობები. ავტომატიზებული საინფორმაციო სისტემა წარმოადგენს არეს, რომელშიც რეალიზდება საინფორმაციო ტექნოლოგიები. რაც შეეხება საინფორმაციო ტექნოლოგიებს, ისინი შეიძლება არსებობდეს კონკრეტული საინფორმაციო სისტემის გარეც; მაგალითად, ასეთებია უზრუნველმყოფი საინფორმაციო ტექნოლოგიები.

5 ავტომატიზებული საინფორმაციო სისტემის ფუნქციონირებისათვის აუცილებელია საშუალებების, მეთოდებისა და ღონისძიებების მთელი კომპლექსის არსებობა, ამიტომ აღნიშნულ სისტემას აქვს შემდეგი **ცხრა** ქვესისტემა (**იხ. ნახ. 1.10**):

1. **საინფორმაციო უზრუნველყოფის ქვესისტემა;** იგი წარმოადგენს სისტემის საინფორმაციო ფონდის აგებისათვის, მისი ფუნქციონირებისა და გამოყენებისათვის საჭირო საშუალებებისა და მეთოდების ერთობლიობას;

ერთმანეთისაგან განასხვავებენ ავტომატიზებული საინფორმაციო სისტემის ‘‘საინფორმაციო ფონდისა’’ და ‘‘საინფორმაციო ბაზის’’ ცნებებს. **საინფორმაციო ფონდი** მოიცავს როგორც ქაღალდურ დოკუმენტებში, ასევე სამანქანო დამხსომებლებში (მზიდებში) დაფიქსირებულ ეკონომიკური ობიექტის მთელ ინფორმაციას, ხოლო **საინფორმაციო ბაზა – მხოლოდ** სამანქანო დამხსომებლებში დაფიქსირებულ ინფორმაციას.

საინფორმაციო უზრუნველყოფა იყოფა გარემანქანურ და შიდამანქანურ უზრუნველყოფებად.

<u>გარემანქანურ საინფორმაციო უზრუნველყოფას</u> წარმოქმნის ინფორმაციის კლასიფიკატორებისა და კოდიფიკატორების სისტემა, აგრეთვე დოკუმენტაციისა და დოკუმენტების მიმოქცევის მაორგანიზებელი სიტემები. ავტომატიზებული სისტემები იყენებს ზოგადკავშირულ, დარგობრივ და ლოკალურ **კლასიფიკატორებს**. პირველი ორი სახის კლასიფიკატორების დამუშავება ხდება ცენტრალიზებულად, ხოლო უკანასკნელი კლასიფიკატორი კონკრეტულ ეკონომიკურ ობიექტში პერსონალის უშუალო მონაწილეობით მუშავდება. აუცილებელია უზრუნველყოფილი იყოს ყველა სახის კლასიფიკატორის შეთავსებადობა.

დოკუმენტაციის სისტემა მოიცავს პირველად, გამოსასვლელ და ნორმატიულ-საცნობარო დოკუმენტებს.

პირველად დოკუმენტებს მიეკუთვნება დარგთაშორისი და დარგობრივი დოკუმენტები, აგრეთვე თავად ეკონომიკურ ობიექტში დასამუშავებული დოკუმენტები.

დარგთაშორისი დოკუმენტები შედის პირველადი სააღრიცხვო დოკუმენტაციის უნიფიცირებული ფორმების ალბომებში, რომლებიც დოკუმენტაციის უნიფიცირებული სისტემის ნაწილია.

გამოსასვლელი დოკუმენტები ასახავს შედეგობრივ ინფორმაციას, რომელიც ფორმირდება კომპიუტერით და წარმოიდგინება ნაბეჭდი დოკუმენტის სახით. ისინი იყოფა სააღრიცხვო რეგისტრების შესაბამის სტანდარტულ ანგარიშგებებად და სპეციალიზებულ მარეგლამენტირებელ ანგარიშგებებად. უკანასკნელებს მიეკუთვნება სახელმწიფო ანგარიშმგებლობა, რომელსაც გააჩნია უნიფიცირებული ტიპური ფორმები, დარგობრივი და რეგიონალური ანგარიშგებები.

ნორმატულ-საცნობარო დოკუმენტები შეიცავს პირობითად მუდმივ ინფორმაციას ნორმატივების, აგრეთვე მატერიალური და შრომითი დანახარჯების, ფასების, ტარიფებისა და ა.შ. შესახებ.

დოკუმენტების მიმოქცევის სისტემა განსაზღვრავს დოკუმენტების მიმოქცევის რაციონალურად ორგანიზებულ სქემას ამ დოკუმენტების წარმოშობის მომენტიდან მათი გამოყენებისა და არქივებში ჩაბარების მომენტამდე. კომპიუტერული სისტემების არსებობის პირობებში სასურველია უზრუნველყოფილი იყოს დოკუმენტების მიმოქცევის სრული ავტომატიზაცია.

<u>შიდამანქანური საინფორმაციო უზრუნველყოფა</u> მოიცავს ავტომატიზებული საინფორმაციო სისტემის საინფორმაციო ბაზას.

საინფორმაციო ბაზა ეწოდება ინფორმაციის სამანქანო დამხსომებელ მოწყობილობებში (ინფორმაციის მზიდებში) შენახვის განსაზღვრული წესების დაცვით შენახული მონაცემების ერთობლიობას.

ავტომატიზებულ საინფორმაციო სისტემებში საინფორმაციო ბაზა წარმოდგენილია პირობითად მუდმივი და მუდმივი მასივების სახით.

პირობითად მუდმივ მასივებს მიეკუთვება სპეციალური სახით ორგანიზებული და კომპიუტერების ხანგრძლივ მეხსიერებაში შენახული ყველა სახის კლასიფიკატორები. ასეთი მასივები შეიცავს შემდეგ ცნობარებს:

- საბუღალტრო აღრიცხვის, ანალიზური აღრიცხვის ობიექტების ანგარიშგებათა გეგმას;
- ნორმატიულ და სხვა დამხმარე ინფორმაციას;
- უნიფიცირებული პირველადი დოკუმენტების ძირითად ფორმებს;
- ბუღალტრული და სტატისტიკური ანგარიშმგებლობის ფორმებს.

მუდმივ მასივებს მიეკუთვება:

- სამეურნეო ოპერაციების შესახებ მიმდინარე მონაცემების მასივები;
- საანგარიშო პერიოდის დასაწყისსა და ბოლოში სინთეტურ და ანალიტიკურ ანგარიშებში არსებული მონაცემები ნაშთების შესახებ;
- პირველადი დოკუმენტების მასივები.

აღნიშნული მონაცემების ორგანიზებისა და შენახვის ხერხები გამოყენებული პროგრამული უზრუნველყოფის თავისებურებებით განისაზღვრება და შეიძლება განსხვავდებოდეს ერთმანეთისაგან.

2. **ტექნიკური უზრუნველყოფის ქვესისტემა;** იგი წარმოადგენს ინფორმაციის შეკრების, რეგისტრაციის, გადაცემისა და დამუშავების ტექნიკური საშუალებების, აგრეთვე ავტომატიზებული საინფორმაციო სისტემის საინფორმაციო ტექნოლოგიებით უზრუნველმყოფი საოფისე ტექნიკის საშუალებების კომპლექსს. თანამედროვე ავტომატიზებული სისტემები შეიძლება ორიენტირებული იყოს კომპიუტერების როგორც ავტონომიურ, ასევე ქსელურ გამოყენებაზე. უკანასკნელ შემთხვევაში ტექნიკური მოწყობილობების ქვესისტემა სპეციალურ ქსელურ მოწყობილობებსაც შეიცავს.

3. **მათემატიკური უზრუნველყოფის ქვესისტემა;** იგი წარმოადგენს ამოცანების გადასაწყვეტი ალგორითმების აღწერისათვის გამოყენებული მათემატიკური საშუალებების, აგრეთვე ავტომატიზებული საინფორმაციო სისტემის პროგრამულ უზრუნველყოფაში გამოყენებული ინფორმაციის წარმოდგენისა და ინტერპრეტაციის მოდელების ერთობლიობას.

ეკონომიკური ამოცანების უმრავლესობა მიეკუთვნება **სტრუქტურიზებულ ამოცანებს**, რომლებსაც გააჩნია გადაწყვეტის ზუსტი ალგორითმი და, მაშასადამე, შესაძლებელია მათემატიკური მოდელით მათი წარმოდგენა. ასეთი ამოცანების გადამწყვეტი ალგორითმები არ საჭიროებს რთული მათემატიკური აპარატის გამოყენებას და ისინი დამუშავებულ ნორმატულ აქტებსა და ინსტრუქციებში არსებული ეკონომიკური გამოთვლების ჩატარების წესების აღწერაზე არის დაფუძნებული.

თავად ამოცანების გადასაწყვეტი ალგორითმების ავტომატიზებული საინფორმაციო სისტემა მოიცავს საგნობრივი სფეროს ძირითადი თავისებურებების ამსახ, მაგრამ პროგრამულ უზრუნველყოფაში სხვადასხვაგვარად რეალიზებული მოდელების მთელ სისტემას. აღნიშნულ მოდელებს მიეკუთვნება:

- **დოკუმენტების მიმოქცევის ორგანიზაციის მოდელი.** იგი განსაზღვრავს საინფორმაციო ბაზაში არსებულ ჩანაწერების სისტემაში დოკუმენტების ფორმირების, შენახვის, დამუშავებისა და ტრანსფორმაციის წესს;
- საწყისი მონაცემების ავტომატიზებული დამუშავების **კონცეფტუალური მოდელი;**
- **მაჩვენებელთა სისტემის მოდელი,** რომელიც ასახავს მაჩვენებლების აგებისა და ინტერპრეტირების წესს;

● **ფუნქციონალური ამოცანების გადაწყვეტის მოდელს** – სტანდარტული, მრავალჯერადად განმეორებადი და ერთმაგი ოპერაციების რეალიზაციის ალგორითმებს;

ზემოთ ჩამოთვლილი მოდელების რეალიზაციის თავისებურებები განსაზღვრავს პროგრამული უზრუნველყოფის გამოყენების ტექნოლოგიის სპეციფიკას და მნიშვნელოვანწილად მის ფუნქციონალურ შესაძლებლობასაც.

მათემატიკური უზრუნველყოფა ასახულია დოკუმენტაციაში, რომელშიც აღწერილია ამოცანები, მოყვანილია მათი გადაწყვეტის მოდელები და ალგორითმები, აგრეთვე ტესტური და საკონტროლო მაგალითები.

მათემატიკური უზრუნველყოფის დასამუშავებლად იწვევენ მათემატიკური მეთოდების მცოდნე სპეციალისტებს, რომლებისთვისაც ნაცნობია საგნობრივი სფერო და რომლებსაც შეუძლიათ მართვის ამოცანების გადაწყვეტის წესის ფორმალიზებული აღწერა.

4. მეთოდური უზრუნველყოფის ქვესისტემა; იგი წარმოადგენს საკანონმდებლო, ნორმატიული, საბუღალტრო აღრიცხვის, ფინანსური და საინვესტიციო ანალიზის, აგრეთვე ცოდნის სხვა სფეროების აქტებისა და ინსტრუქციების ერთობლიობას, რომელიც საშუალებას გვაძლევს დავამუშაოთ ეკონომიკური ინფორმაცის გარდაქმნისა და გადაწყვეტილების მიღების იურიდიული მხარდაჭერის უზრუნველყოფის ფორმირებისათვის საჭირო ალგორითმები.

5. ლინგვისტური უზრუნველყოფის ქვესისტემა; იგი წარმოადგენს ავტომატიზებული საინფორმაციო სისტემების დამუშავებისა და ფუნქციონირების პროცესში გამოყენებული ხელოვნური ენების, ტერმინებისა და განსაზღვრებების სისტემას. მასში შედის:

● ინფორმაციის სტრუქტურული ერთეულების (რეკვიზიტების, მაჩვენებლების, დოკუმენტების, მასივების, ნაკადების) აღმწერი ენები;

● საინფორმაციო ბაზის მონაცემების მართვისათვის საჭირო ენები;

● ამოცანების გადასაწყვეტი ამოცანების აღსაწერი ენები;

● საინფორმაციო-საძიებო სისტემებში გამოყენებული ენები;

● სპეციალური დანიშნულების ენები.

6. პროგრამული უზრუნველყოფის ქვესისტემა; იგი წარმოადგენს მართვის ავტომატიზებულ სისტემაში მონაცემების დამუშავებისა და გადაცემის პროგრამების, აგრეთვე მათი გამოყენების დოკუმენტაციის კომპლექსს. მასში შედის სისტემური, დამხმარე და გამოყენებითი (სპეციალიზებული) პროგრამული უზრუნველყოფები. მოკლედ განვიხილოთ თითოეული მათგანი.

სისტემური პროგრამული უზრუნველყოფა ეწოდება გამოთვლითი ტექნიკის, ქსელური მოწყობილობების საშუალებებისა და სხვადასხვა პროგრამული უზრუნველყოფის ფუნქციონირების მმართველ ოპერაციულ სისტემებს.

დამხმარე პროგრამული უზრუნველყოფა წარმოადგენს ფუნქციონალური ამოცანების გადასაწყვეტი პროგრამების ფუნქციონირებისა და მომხმარებლებისათვის დამატებითი სერვისის უზრუნველყოფისათვის აუცილებელი პროგრამული საშუალებების ერთობლიობას. ასეთი პროგრამული უზრუნველყოფა მოიცავს:

● მონაცემების ბაზების მართვის სისტემებს;

● პროგრამების ინტერპრეტატორებს, რომლებიც დამუშავებულია დაპროგრამების მაინტერპრეტირებელი სისტემების მიერ;

● პროგრამების ფუნქციონირებისათვის აუცილებელ სხვადასხვა გარე ბიბლიოთეკებს;

● დაარქივებისა და არასანქცირებული შეღწევისაგან მონაცემების დაცვის საშუალებებს და ა.შ.

ასე, მაგალითად, ეკონომიკური დანიშნულების მრავალი პროგრამული პროდუქტი გამოიყენება მონაცემების შენახვისა და მონაცემების ბაზების მართვის სისტემებში შეღწევისათვის და მათ გარეშე ფუნქციონირება არ შეუძლია.

მრავალი პროგრამა ანგარიშების ფორმირებას **Microsoft Exel**-ის ფორმატში ახდენს, ამიტომ ამ ანგარიშების დასათვალიერებლად აუცილებელია კომპიუტერში არსებობდეს ან თავად ცხრილური **Exel** პროცესორი, ან ამ ფორმატის ფაილების დასათვალიერებელი პროგრამები.

ანგარიშების დასათვალიერებელი ან მონაცემების შეტანითვის საჭირო ზოგიერთი თანამედროვე პროგრამა მოითხოვს კომპიუტერში არსებობდეს web-გვერდების დასათვალიერებელი რაიმე პროგრამა და მისი არარსებობისას იგი ნორმალურად ვერ ფუნქციონირებს.

გამოყენებითი პროგრამული უზრუნველყოფა წარმოადგენს ფუნქციონალური ამოცანების გადაწყვეტის ალგორითმების უშუალოდ მარეალიზებელი სპეციალიზებული პროგრამების ერთობლიობას.

7. სამართლებრივი უზრუნველყოფის ქვესისტემა; იგი წარმოადგენს ავტომატიზებული საინფორმაციო სისტემის შექმნისა და ფუნქციონირების დროს სამართლებრივი ურთიერთობების მარეგლამენტირებელი სამართლებრივი ნორმების ერთობლიობას.

ავტომატიზებული საინფორმაციო სისტემის **დამუშავების ეტაპზე** სამართლებრივი უზრუნველყოფა მოიცავს ნორმატიულ აქტებს, რომლებიც დაკავშირებულია:

- სისტემის დამმუშავებელსა და შემკვეთს შორის სახელშეკრულებლო ურთიერთობებთან;
- სისტემის დამუშავების პროცესის გადახრების დარეგულირებასთან;
- სხვადასხვა რესურსებით დამუშავების პროცესის უზრუნველყოფასთან.

ავტომატიზებული საინფორმაციო სისტემის **ფუნქციონირების ეტაპზე** სამართლებრივი უზრუნველყოფა განსაზღვრავს:

- მართვის პროცესში სისტემის სტატუსს;

ავტომატიზებული საინფორმაციო სისტემის ცალკეული სტრუქტურების კომპეტენციებსა და მათი ფუნქციონირების ორგანიზაციას;

- ავტომატიზებული საინფორმაციო სისტემის ფუნქციონირების უსაფრთხოების სამართლებრივ უზრუნველყოფას.

8. ორგანიზაციული უზრუნველყოფის ქვესისტემა; იგი წარმოადგენს ავტომატიზებული საინფორმაციო სისტემის შექმნისა და ფუნქციონირების პროცესში ტექნიკურ საშუალებებთან, პროგრამულ უზრუნველყოფასთან და ერთმანეთს შორის აღნიშნული სისტემის თანამშრომლების ურთიერთზემოქმედების მარეგლამენტირებელი საშუალებებისა და მეთოდების ერთობლიობას.

ავტომატიზებული საინფორმაციო სისტემა წარმოადგენს ''ადამიანი-მანქანა'' სისტემას და ამიტომ ვერ იფუნქციონირებს მისი ექსპლუატაციის წესების მცოდნე პერსონალის არსებობის გარეშე. ამასთანავე, ერთმანეთისაგან უნდა გავმიჯნოთ სისტემის **მომხმარებელი** და **მომსახურე** პერსონალი.

მომხმარებელ პერსონალს წარმოადგენენ საკუთარი პროფესიული ამოცანების გადასაწყვეტად ავტომატიზებული საინფორმაციო სისტემის უშუალოდ გამომყენებელი მართვის აპარატის თანამშრომლები. **მომსახურე პერსონალს** მიეკუთვნება სისტემის ფუნქციონირების უზრუნველსაყოფად საჭირო ფუნქციების შემსრულებელი თანამშრომლები; ისინი უზრუნველყოფენ ტექნიკური საშუალებებისა და პროგრამული უზრუნველყოფის მუშა მდგომარეობაში ყოფნას, საინფორმაციო ბაზის დაცულობას.

მსხვილ ეკონომიკურ ობიექტებზე ავტომატიზებული მართვის სისტემის მომსახურე პერსონალი ჩვეულებრივ შედის დაწესებულების მართვის ავტომატიზებული სისტემის მომსახურე სპეციალიზებული ქვედანაყოფის შემადგენლობაში. როგორც წესი, ესენი წარ-

მოადგენენ საკომუნიკაციო, გამოთვლითი, აგრეთვე მმართველობითი და საინჟინრო ტექნიკური სამუშაოების მექანიზაციისა და ავტომატიზაციისათვის გამოყენებულ ტექნიკურ საშუალებათა კომპლექსის (რომელსაც შემოკლებით **ორგტექნიკა** ანუ **საორგანიზაციო ტექნიკა** ეწოდება) მომსახურების პროფესიული ჩვევების მქონე ტექნიკურ სპეციალისტებს. ხშირად ასეთ ქვედანაყოფს უწოდებენ **საინფორმაციო ტექნოლოგიის განყოფილებას**, ხოლო მის თანამშრომლებს – **საინფორმაციო ტექნოლოგიების სპეციალისტებს.** ისინი შემდეგი ორი მიმართულებით საქმიანობენ:

- საინფორმაციო სისტემის აგება და განვითარება (დამპროექტებლების ჯგუფი);
- ტექნიკური და პროგრამული უზრუნველყოფის თანხლება (თანხლების ჯგუფი).

საინფორმაციო ტექნოლოგიის **პირველი მიმართულების** სპეციალისტები:

- აპროექტებენ დაწესებულების ავტომატიზებულ საინფორმაციო სისტემას;
- აწესრიგებენ საინფორმაციო ნაკადებს;
- ქმნიან კომპიუტერების გამოყენებითი ინფორმაციის დამუშავების ტექნოლოგიას;
- აყენებენ კომიუტერებსა და პროგრამულ უზრუნველყოფას;
- აპროექტებენ და ამონტაჟებენ გამოთვლით ქსელს;
- უზრუნველყოფენ ავტომატიზებული საინფორმაციო სისტემის აპარატურულ-პროგრამული კომპლექსის დანერგვისა და თანხლების პროცესის შესრულებას;
- ასწავლიან და ამზადებენ მომხმარებლებს.

არც თუ ისე იშვიათად მათ ფუნქციებში შედის ავტომატიზებული საინფორმაციო სისტემის ფუნქციონირებისათვის საჭირო პროგრამული უზრუნველყოფის დამუშავებაც. სისტემის დაპროექტებისა და შექმნის პროცესში საინფორმაციო ტექნოლოგიის განყოფილების სპეციალისტები ამუშავებენ სხვადასხვა განყოფილებებისა და თანამშრომლების მიერ სისტემის ექსპლუატაციის რეგლამენტებს, აგრეთვე სამართლებრივ უზრუნველყოფას.

საქმიანობის **მეორე მიმართულებას** უზრუნველყოფს **სისტემის ადმინისტრატორი** და **ექსპლუატაციის ჯგუფი.**

სისტემის ადმინისტრატორი:

- პასუხს აგებს ქსელის რესურსების განაწილებაზე;
- განსაზღვრავს მონაცემებთან მომხმარებლის შეღწევის უფლებებს;
- განსაზღვრავს მომხმარებლის მიერ მონაცემების დამუშავების ფუნქციებს;
- აკონტროლებს თუ როგორ იცავს მომხმარებელი მისთვის მინიჭებულ უფლებებსა და როგორ ასრულებს მასზე დაკისრებულ ფუნქციებს.

ექსპლუატაციის ჯგუფი უზრუნველყოფს:

- უსაფრთხოებას, კონფიდენციურობასა და მონაცემების მთლიანობას (ებრძვის ამოვარდნებს, ვირუსებს, არასანქცირებულ შეღწევებს);
- მონაცემთა ბაზის ადმინისტრირებას;
- მონაცემების შეტანის გრაფიკების დამუშავებას და მათი შესრულების კონტროლს;
- მიმდინარე შეკეთებისა და მოწყობილობების პროფილაქტიკის გეგმა-გრაფიკების შედგენას და ა.შ.

მცირე ეკონომიკურ ობიექტებში ასეთი ქვედანაყოფის ფუნქციები შეიძლება დავიყვანოთ ტექნიკის მარტივ მომსახურებასა და მუშა მდგომარეობაში პროგრამული უზრუნველყოფის შენახვაზე. ამას ახორციელებს საინფორმაციო ტექნოლოგიის ერთი ან რამდენიმე სპეციალისტი.

დაწესებულებაში საინფორმაციო ტექნოლოგიის საკუთარი განყოფილების არსებობის დროსაც ავტომატიზებული საინფორმაციო სისტემის აგებისა და განვითარების ფუნქციები ძალიან ხშირად გადაეცემა გარეშე ფირმებს, რომლებიც დასპეციალიზებულია ტირაჟირებადი პროგრამული პროდუქტების დანერგვასა და ადაპტაციაზე ან შეკვეთილი პროგრამული უზრუნველყოფის დამუშავებაზე.

9. ორგანიზაციული უზრუნველყოფის ქვესისტემა; იგი წარმოადგენს ავტომატიზებული საინფორმაციო სისტემის შექმნისა და ფუნქციონირების პროცესში სპეციალისტების მაღალეფექტური და უშეცდომო საქმიანობისათვის საჭირო ოპტიმალური პირობების შექმნისათვის განკუთვნილი საშუალებებისა და მეთოდების ერთობლიობას. იგი მოიცავს სხვადასხვა დოკუმენტაციების კომპლექსს, რომლებშიც მოცემულია:

- სამუშაო ადგილებისადმი წაყენებული მოთხოვნები;
- პერსონალის, პროგრამული უზრუნველყოფის (ეკრანების დიზაინი) მუშაობის პირობები და ა.შ.
- ამ მოთხოვნების რეალიზაციის ყველაზე მიზანშეწონილი ხერხების ნაკრები და მათი რეალიზაციის ერგონომიკული ექსპერტიზა;
- მეთოდების, სასწავლო-მეთოდური მასალებისა და ტექნიკური საშუალებების კომპლექსი, რომლის მეშვეობითაც შესაძლებელია ფორმულირებული იქნეს პერსონალის მომზადების დონისადმი წასაყენებელი მოთხოვნები და შეიქმნას კადრების შერჩევის სისტემა;
- მართვის ავტომატიზებული სისტემის პირობებში სპეციალისტების მუშაობის მაღალი ეფექტურობის უზრუნველმყოფი მეთოდიკების ნაკრები.

1.7. ავტომატიზებული სამუშაო ადგილის დახასიათება და კლასიფიკაცია

1

კონკრეტულ დაწესებულებაში ორგანიზებული ავტომატიზებული საინფორმაციო სისტემა აიგება ამ დაწესებულებაში მომუშავე სპეციალისტებისათვის ავტომატიზებული სამუშაო ადგილების შექმნის გზით. აღნიშნულიდან გამომდინარე ავტომატიზებული სამუშაო ადგილი წარმოადგენს ავტომატიზებული საინფორმაციო სისტემის შემადგენელ ნაწილს და იგი შეიძლება შემდეგნაირად განისაზღვროს:

ავტომატიზებული სამუშაო ადგილი ეწოდება სპეციალისტისათვის მოწყობილ ისეთ სამუშაო ადგილს, რომელიც იმისათვის, რომ აღნიშნულ სპეციალისტს შეეძლოს მონაცემების დამუშავების გზით მოახდინოს მისი პროფესიული ფუნქციების შესასრულებლად საჭირო ინფორმაციის ფორმირება, აღჭურვილია პერსონალური კომპიუტერით, პროგრამული უზრუნველყოფითა და ინდივიდუალური ან კოლექტიური მოხმარების საინფორმაციო რესურსებით.

შემოკლებულად **ავტომატიზებული სამუშაო ადგილი** განსაზღვრული სახის საქმიანობის ავტომატიზაციისათვის განკუთვნილი პროგრამულ-ტექნიკური კომპლექსია, რომელშიც ტექნოლოგიური მოწყობილობები, როგორც წესი, ***SCADA***– **სისტემების** საშუალებით იმართება.

SCADA (ინგლ. ***"S****upervisory* *Control* *And* *Data* *Acquisition****"*** – "მონაცემების დისპეტჩერული მართვა და შეკრება") წარმოადგენს მონიტორინგის ან მართვის ობიექტის შესახებ ინფორმაციის შეკრების, გარდაქმნის, ასახვისა და დაარქივების სისტემის დასამუშავებლად ან რეალურ დროში მისი მუშაობისათვის განკუთვნილ **პროგრამულ პაკეტს**.

პროგრამების პაკეტი, ანუ, როგორც ხშირად მას უწოდებენ, **გამოყენებითი პროგრამების პაკეტი** (ინგლ. "***application package"***), თავის მხრივ, წარმოადგენს *საგნობრივ არეში* არსებული გარკვეული კლასის ამოცანების გადასაწყვეტად განკუთვნილი *მოდულების* ნაკრებს; **მოდული** ეწოდება პროგრამის ფუნქციონალურად დამთავრებულ ნაწილს, რომელიც სხვა პროგრამებში გამოსაყენებლად არის განკუთვნილი და გაფორმებულია განცალკევებული ფაილის სახით ან გააჩნია დასახელება, ხოლო **საგნობრივი არე** წარმოად-

გენს ყველა იმ საგნების სიმრავლეს, რომელთა თვისებები და ურთიერთდამოკიდებულებები განიხილება მოცემულ კონკრეტულ სამეცნიერო თეორიაში.

დაწესებულებაში არსებული განყოფილება, რომელშიც მომუშავე თანამშრომლები ავტომატური სამუშაო ადგილებით არიან უზრუნველყოფილები, წარმოადგენს აღნიშნული დაწესებულების **ავტომატიზებულ ქვეგანყოფილებას.** მასში ინფორმაციის დასამუშავებლად საჭირო რუტინული სამუშაოს მნიშვნელოვან ნაწილს ასრულებს კომპიუტერი. მიუხედავად ამისა, სპეციალისტს შეუძლია აქტიურად ჩაერიოს მონაცემების დამუშავების ამოცანების გადაწყვეტის პროცესში და დამოუკიდებლად მოახდინოს ისეთი ინფორმაციის ფორმირება, რომელიც მას დასაბუთებული გადაწყვეტილების მიღების საშუალებას მისცემს.

კომპიუტერი ორგანულად შეეერწყა სპეციალისტის სამუშაო ტექნოლოგიას და გადაიქცა მისთვის მუდმივად გამოსაყენებელ საგნად. ამავე დროს ინფორმაციის დამუშავების ფორმალურ-ლოგიკური ასპექტებიდან აქცენტის გადატანა მოხდა გადაწყვეტილებების მიღების პროცესზე. ასეთი ტექნოლოგია ამცირებს როგორც საქაღალდე დოკუმენტების ნაკადს, ასევე შესასრულებელი სამუშაოების შრომატევადობას, ამაღლებს მუშაკების პროფესიულ დონეს და მათი მუშაობის კომფორტულობას.

ისევე როგორც შრომის ორგანიზაციის ხელითი შესრულების ტექნოლოგიის დროს, სპეციალისტს სრული პერსონალური პასუხისმგებლობა ეკისრება მთელ პროცესზე, მაგრამ ტრადიციული ფუნქციების შესრულების გაგრძელებასთან ერთად მას პერსონალური კომპიუტერის ოპერატორის როლის შესრულებაც ევალება და ინფორმაციის ავტომატური დამუშავების პროცესის უშუალო მონაწილეც ხდება.

ავტომატიზებული სამუშაო ადგილი შემდეგ ხუთ ძირითად კომპონენტს მოიცავს:

1. **პერსონალურ კომპიუტერს;**
2. ინფორმაციის დამუშავებისათვისათვის საჭირო **პროგრამების კომპლექსს;**
3. **მასწავლებელ სისტემას** (მომხმარებლისათვის განკუთვნილ დოკუმენტაციის ჰიპერტექსტურ სისტემას; კარნახის ინტეგრირებულ სისტემას; სანიშნების, მაჩვენებლებისა და ცნობარების სისტემას,; კონტროლისა და შეცდომების აღმოჩენის სისტემას);
4. **ავტომატიზებული სამუშაო ადგილის აწყობის საშუალებებს** (გამოთვლეის ალგორითმებს; ანალიტიკურ და ტექნოლოგიურ პარამეტრებს; ისეთ მოწყობილობებს, როგორებიცაა პრინტერი, სკანერი, მოდემი; საეკრანო ფორმების ერნომიკას და ა.შ.);
5. **ავტომატიზებული სამუშაო ადგილის ექსპლუატაციის საშუალებებს** (კლასიფიკატორებს; საანგარიშო ფორმების გენერატორს; კავშირის არხებით მონაცემების მიღება-გაგზავნის ინსტრუმენტარიუმს, მონაცემების ბაზის ადმინისტრატორს, კონკრეტული მომხმარებლების მუშაობის მონიტორინგს).

ზემოთ ჩამოთვლილის გარდა ავტომატიზებული სამუშაო ადგილი კომპლექტდება პროგრამების გამოსაყენებლად საჭირო დოკუმენტაციებითა და მეთოდიკური მასალებით, აგრეთვე ინფორმაციის დამუშავების სამუშაოთა შესრულების რეგლამენტებით.

ავტომატური სამუშაო ადგილები შეიძლება ფუნქციონირებდეს ავტონომიურად ან კომპიუტერული ქსელის შემადგენლობაში. მუშაობის **ავტონომიური რეჟიმის** დროს **ავტომატიზებულ სამუშაო ადგილს** ქმნიან ცალკეული ფუნქციონალური ამოცანების გადასაწყვეტად; მათ არ შეუძლია ეკონომიკური ობიექტის მთელი საინფორმაციო ბაზა ოპერატიულად გამოიყენოს, ხოლო სხვადასხვა ავტომატიზებულ სამუშაო ადგილებს შორის ინფორმაციის გაცვლა სამანქანო დამხსომებელი მოწყობილობების (ინფორმაციის მზიდების) დახმარებით ხორციელდება. **ასეთი სამუშაო ადგილები დღეს უკვე იშვიათად გამოიყენება.**

კომპიუტერული ქსელების შემადგენლობაში მუშაობის დროს შესაძლებელია:

● ავტომატიზებულ სამუშაო ადგილებს შორის ინფორმაცია გაიცვალოს კავშირის არხებით;

● გაერთიანდეს მართვის ობიექტების საინფორმაციო სივრცე და ორგანიზებული იქნეს მასში ნებისმიერი მუშაკის შეღწევის ორგანიზება აღნიშნული მუშაკის უფლებამოსილების ფარგლებში.

ავტომატიზებული სამუშაო ადგილიდან შეიძლება ტერიტორიულად დაშორებულ ქვედანაყოფებზე მისაერთებლად და საერთო დანიშნულების გარე საინფორმაციო სამსახურებში (საინფორმაციო-საძიებო სისტემებში, მონაცემების ბაზებში და ა.შ.) შესაღწევად შეიძლება გამოყენებული იქნეს არა მარტო ლოკალური, არამედ გლობალური ქსელებიც.

თითოეული ავტომატიზებული სამუშაო ადგილი დამოუკიდებელ ქვესისტემად განიხილება და ერთობლიობაში ისინი წარმოქმნის ერთ მთლიანობას. აღნიშნულის გამო განყოფილების გამგესათვის ოპერატიულად ხელმისაწვდომი გახდება გადაწყვეტილების მისაღებად საჭირო დამუშავებული ინფორმაცია, რაც დაეხმარება მას უხელმძღვანელოს ფუნქციონალური ამოცანების გადაწყვეტის პროცესს და მოახდინოს ცალკეული სპეციალისტების მუშაობის შედეგების ინტეგრირება. იმავე დროს შენარჩუნებული იქნება თითოეული სპეციალისტის ავტონომიური მუშაობის შესაძლებლობა.

დაწესებულების საორგანიზაციო სტრუქტურა განსაზღვრავს **ავტომატიზებული სამუშაო ადგილების ნომენკლატურას (რაოდენობას),** ხოლო მიზნებისა და ფუნქციების დანაწევრება, აგრეთვე თანამშრომლებს შორის მოვალეობების განაწილება – **კონკრეტული ავტომატიზებული სამუშაო ადგილების ფუნქციონალურ შინაარსს** (კონკრეტულ სამუშაო ადგილზე გადასაწყვეტი ამოცანების შემადგენლობას).

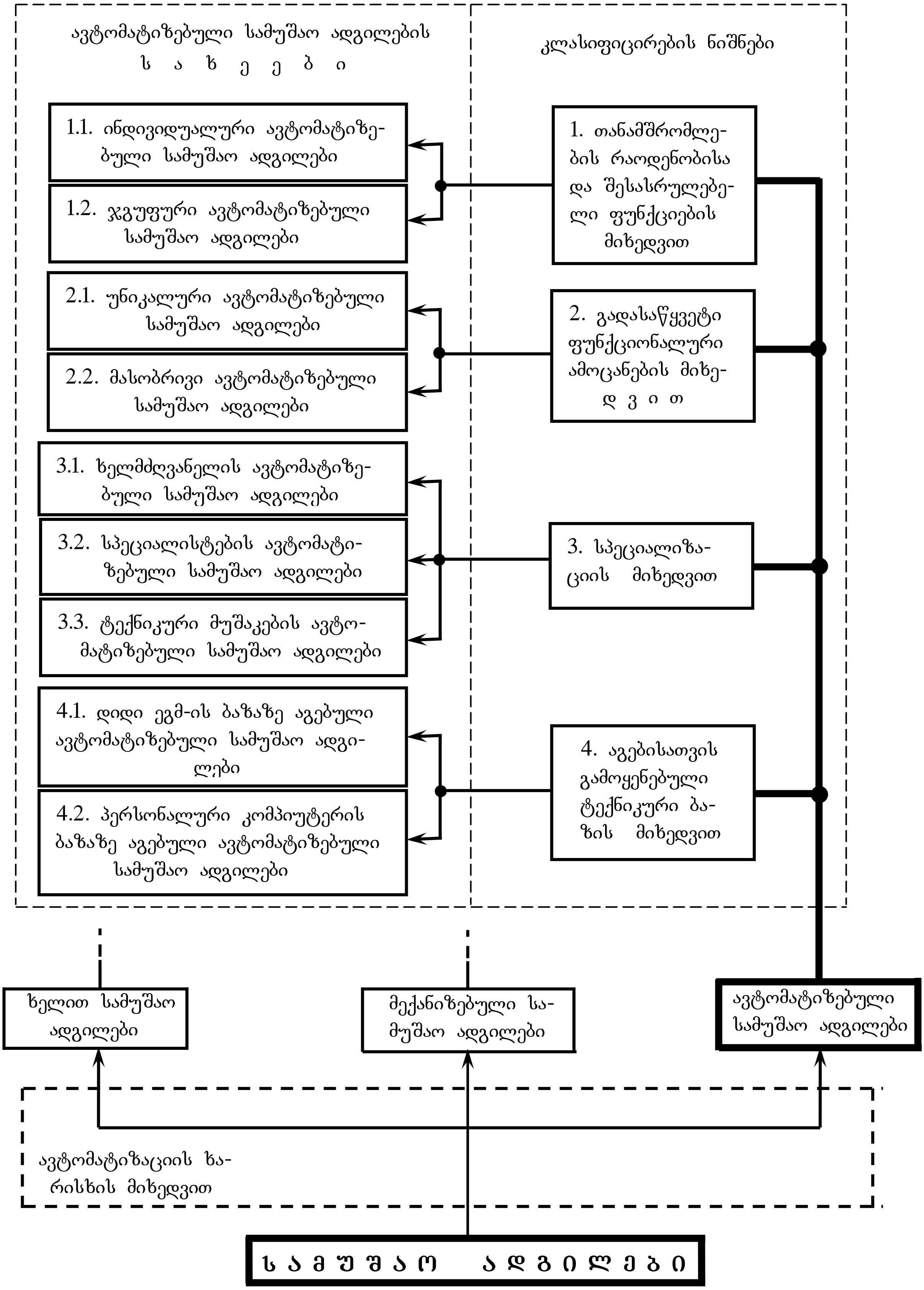
თანამშრომლების მიერ შესასრულებელი ფუნქციების სპეციფიკაზე დამოკიდებულებით თითოეული მათგანი სხვადასხვა საინფორმაციო და პროგრამულ რესურსებს საჭიროებს. სამუშაოების განაწილება დამოკიდებულია როგორც ტექნიკურ ბაზაზე, ასევე სპეციალისტების **კომპიუტერული მომზადების დონეზეც!** როგორც წესი, ავტომატიზებული სამუშაო ადგილი ორგანიზდება სამუშაოთა არსებული განაწილების შესაბამისად. სამუშაოების მოცულობასა და **კომპიუტერების საერთო რაოდენობაზე დამოკიდებულებით** ერთ სამუშაო ადგილზე შესაძლებელია ერთმანეთისაგან განსხვავებული ამოცანების გადაწყვეტა. შესაძლებელია სამუშაოების შესრულება იმგვარად იქნეს ორგანიზებული, რომ ერთი ამოცანა რამდენიმე სამუშაო ადგილს შორის განაწილდეს.

4 ნებისმიერ სპეციალისტს (ბუღალტერს, საკრედიტო-საბანკო სისტემის სპეციალისტს, იურისტს და ა.შ.) მასზე დაკისრებული მოვალეობის წარმატებით შესასრულებლად სჭირდება ჰქონდეს სათანადოდ აღჭურვილი **სამუშაო** ადგილი.

სამუშაო ადგილი წარმოადგენს მუშაკის ყოფნისათვის გამოყოფილ და მისი შრომის გამომყენებელი საშუალებების განთავსების ზონას, რომელიც ორგანიზებულია ტექნიკური და **ერგონომიკული** ნორმატივების დაცვით და აღჭურვილია სხვადასხვა მოწყობილობებით, რომლებიც აღნიშნულ მუშაკს სჭირდება მის წინაშე დასმული ამოცანების წარმატებულად და მოხერხებულად გადაწყვეტისათვის.

ზემოთ მოყვანილ განმარტებაში გამოყენებულია ტერმინი ‘‘ერგონომიკა’’. ერგონომიკის საერთაშორისო ასოციაციის (ინგლ. ***I**nternational **E**rgonomics **A**ssociation,* შემოკლებით ***IEA*) *2010*** წელს ამ ტერმინის განმარტება ასე იქნა ფორმულირებული:

ერგონომიკა (ძვ. ბერძ. ‘‘*ἔργον*’’ – ‘‘სამუშაო’’ და ***‘‘ νόμος’’*** – ‘‘კანონი’’) ეწოდება ადამიანსა და სისტემის სხვა ელემენტებს შორის ურთიერთზემოქმედების შემსწავლელ სამეცნიერო დისციპლინას, აგრეთვე ადამიანის კეთილდღეობის უზრუნველსყოფად და სისტემის საერთო მწარმოებლურობის ოპტიმიზაციისათვის ამ დისციპლინის თეორიის პრინციპების, მონაცემებისა და მეთოდების გამოყენების სფეროს.



ნახ. 1.11. სამუშაო ადგილების კლასიფიკაციის ფრაგმენტი, რომელშიც დეტალიზებულია ავტომატიზებული სამუშაო ადგილების სახეები

ავტომატიზაციის ხარისხზე დამოკიდებულებით განასხვავებენ შემდეგი სამი სახის სამუშაო ადგილს (ნახ. **1.11**):

● **ხელით სამუშაო ადგილს**, რომელშიც მუშაკის განკარგულებაშია სპეციალური ავეჯი (მაგიდა, სკამი, კარადები და ა.შ.), ტელეფონი, სახაზავები, ცხრილები და სხვა დამხმარე საშუალებები;

● **მექანიზებულ სამუშაო ადგილს,** რომელიც ხელით სამუშაო ადგილში არესებულ საშუალებებს გარდა დამატებით შეიცავს უმარტივეს ან დაპროგრამებად კალკულატორებს;

● **ავტომატიზებულ სამუშაო ადგილს**, რომელიც დამატებით აღჭურვილია სათანადო პროგრამული უზრუნველყოფის მქონე პერსონალური კომპიუტერით. ავტომატიზებული სამუშაო ადგილის არსებობის დროს ინფორმაციის შენახვისა და დამუშავების ძირითადი ოპერაციების შემსრულებელია გამომთვლელი ტექნიკა, ხოლო სპეციალისტი ასრულებს მმართველობითი გადაწყვეტილების მისაღებად აუცილებელი ისეთი ოპერაციების ისეთ ნაწილს, რომელიც მოითხოვს შემოქმედებით მიდგომას.

1.11 ნახაზზე მოცემულია ავტომატიზებული სამუშაო ადგილების კლასიფიკაცია. მოკლედ დავახასიათოთ ავტომატიზებული სამუშაო ადგილების სახეები, რომლებიც აღნიშნულ ნახაზზეა მოცემული.

1.1. **ინდივიდუალური ავტომატიზებული სამუშაო ადგილები** იქმნება სხვადასხვა რანგის ხელმძღვანელი მუშაკებისათვის;

1.2. **ჯგუფური ავტომატიზებული სამუშაო ადგილები** განკუთვნილია პირებისათვის, რომლებიც ამზადებენ შემდგომი გამოყენებისა და მმართველობითი გადაწყვეტილებების მისაღებად აუცილებელ ინფორმაციებს (ბუღალტრებისათვის, ფინანსისტებისათვის, საქმის მწარმოებლებისათვის და ა.შ. ორგანიზებული ავტომატიზებული სამუშაო ადგილები);

2.1. **უნიკალური ავტომატიზებული სამუშაო ადგილები** წარმოადგენს არასტანდარტული ამოცანების გადასაწყვეტად ორგანიზებულ ვიწროსპეციალიზებულ სამუშაო ადგილებს;

2.2. **მასობრივი ავტომატიზებული ადგილები** იქმნება სხვადასხვა დარგში ტიპური ამოცანების გადასაწყვეტად;

3.1 **ხელმძღვანელის ავტომატიზებული სამუშაო ადგილისათვის** დამახასიათებელი უნდა იყოს ჩაკეტილობა, რაც იმას ნიშნავს, რომ მან სრულად უნდა უზრუნველყოს ხელმძღვანელის ავტონომიური მუშაობა. ხელმძღვანელის მიერ შესასრულებელი ძირითადი ფუნქციები (ოპერატიული მართვა და გადაწყვეტილების მიღება) განსაზღვრავს მოცემული სამუშაო ადგილისადმი წაყენებულ შემდეგ მოთხოვნებს:

● უნდა არსებობდეს ოპერატიული და სარწმუნო ინფორმაციით მუდმივად შევსებადი საკმაოდ განვითარებული საინფორმაციო ბაზა, რომლის ნაწილის გამოყენება შეეძლება შეზღუდული რაოდენობის პირებს, ხოლო მისი ცალკეული ფრაგმენტების გამოყენების უფლება მხოლოდ ხელმძღვანელს ექნება;

● უზრუნველყოფილი უნდა იყოს ოპერატიული კავშირი ინფორმაციის სხვა (გარეშე) წყაროებთან;

● შესაძლებელი უნდა იყოს ინფორმაციის ოპერატიულად მოპოვება;

● ინფორმაცია წარმოდგენილი უნდა იყოს ადვილად აღსაქმელად მოსახერხებელი ფორმით;

● უნდა არსებობდეს გადაწყვეტილების მიღების პროცესის მხარდამჭერი პროგრამული საშუალებები.

3.2. **სპეციალისტის ავტომატიზებულ სამუშაო ადგილმა** საშუალება უნდა მისცეს სპეციალისტს ყველა აუცილებელი ინფორმაციის მაქსიმალურად გამოყენების გზით გადაწყვიტოს მის წინ მდგარი ნებისმიერი ფუნქციონალური ამოცანა. თანამშრომლის პროფესიული ორიენტაცია განსაზღვრავს მოცემული სამუშაო ადგილისადმი წაყენებულ შემდეგ მოთხოვნებს:

● შესაძლებელი უნდა იყოს მონაცემების პერსონალური და კორპორაციული ბაზების გამოყენება;

- უზრუნველყოფილი უნდა იყოს ოპერატიული კავშირი ინფორმაციის დამატებით (გარეშე) წყაროებთან;
- შესაძლებელი უნდა იყოს დაგროვილი გამოცდილების გამოყენებით გასაანალიზებელი პროცესების მოდელირება;
- უზრუნველყოფილი უნდა იყოს სისტემის მრავალფუნქციონალურობისა და მოქნილობის მაღალი დონე.

3.3. **ტექნიკური მუშაკის ავტომატიზებულმა სამუშაო ადგილმა** შესაძლებლობა უნდა მისცეს ტექნიკურ მუშაკს თავი აარიდოს ყოველდღიურად რუტინული სამუშაოების შესრულებას, რომლებიც გარკვეული პროფესიული ჩვევების ქონას საჭიროებს. მუშაკის პროფესიული ორიენტაცია განსაზღვრავს მოცემული სამუშაო ადგილისადმი წაყენებულ მოთხოვნებს, კერძოდ, შესაძლებელი უნდა იყოს:

- ინფორმაციის შეტანა (ოპერატორისა და მემანქანესათვის);
- კარტოთეკებისა და არქივების შედგენა (არქივარიუსებისათვის);
- ხელმძღვანელის ყოველდღიური პირადი გეგმის გაკონტროლება;
- შემომავალი და გამავალი დოკუმენტაციის დამუშავება (წერილების განყოფიილების ინსპექტორისათვის) და ა.შ.

4.1. **დიდი ეგმ-ის ბაზაზე აგებული ავტომატიზებული სამუშაო ადგილი** სპეციალისტებს საშუალებას აძლევს საკუთარი საინფორმაციო-გამოთვლითი ცენტრის თანამშრომლების ძალისხმევით განხორციელებული ტექნიკური და პროგრამული მხარდაჭერით გამოიყენოს მონაცემების დიდი მასივები. აქვე უნდა აღინიშნოს, რომ ავტომატიზებული სამუშაო ადგილების ორგანიზებისათვის დიდი **ეგმ**-ის გამოყენებას ზღუდავს შემდეგი ფაქტორები:

- ოპერაციული სისტემისადმი წაყენებული მოთხოვნების სიხისტე;
- გამოყენებული პროგრამული საშუალებების არასაკმარისი მოქნილობა;
- მომხმარებლისადმი (არადამპროგრამებელისადმი) გამოთვლითი სისტემის სუსტი ორიენტაცია;
- გამოთვლითი საშუალებების ტექნიკური და პროგრამული უზრუნველყოფის სპეციალური ქვედანაყოფის არსებობის აუცილებლობა;
- სამანქანო რესურსების მაღალი ღირებულება.

4.2. **პერსონალური კომპიუტერის ბაზაზე აგებული ავტომატიზებული სამუშაო ადგილი** წარმოადგენს ავტომატიზებული სამუშაო ადგილის შექმნის ყველაზე მარტივ და გავრცელებულ ვარიანტს, რადგან იგი თავისუფალია დიდი **ეგმ**-ის ბაზაზე აგებული ავტომატიზებული სამუშაო ადგილისათვის დამახასიათებელი ყველა ნაკლოვანებისაგან.

1.8. ყველა გზა მიდის ... რიცხვებისაკენ!

“ნებისმიერი საგანი შეიძლება რიცხვის სახით წარმოვადგინოთ”;
“სამყაროს მართავენ რიცხვები; სამყაროში ყველაფერი არის რიცხვი”;
პითაგორა

“პითაგორელებმა თავიანთ სწავლებაში გამოხატეს ანტიკური ადამიანის უდიდესი მისწრაფება მათემატიკურად ზუსტი ლოგიკური აზროვნებისა და როგორც საკუთარი სივრცულ-გეომეტრიული, ასევე სტრუქტურულ-რიცხვითი მიმართებების გათვალისწინებით სამყაროს ათვისებისაკენ”
ა.თ.ლოსევი

1 ციფრული და ანალოგური კომპიუტერები (**იხ. 1.1 პარაგრაფი**) შესაბამისად ახდენენ წყვეტილი (დისკრეტული) და უწყვეტი (ანალოგური) ფორმის სიგნალების დამუშავებას.

წყვეტილობა და უწყვეტობა მატერიის აგებულებისა და მისი განვითარების დამახასიათებელი ფილოსოფიური კატეგორიებია. ნებისმიერი ობიექტი გარკვეული შინაგანი სტრუქტურის მქონე კონკრეტული სახის (მთლიანობის) რეალობაა; ობიექტის შინაგან სტრუქტურულობის, სირთულის ფაქტს გამოხატავს **წყვეტილობის** ანუ **დისკრეტულობის** კატეგორია, ხოლო მის მთლიან სახეს – **უწყვეტობის** ანუ **ანალოგურობის** კატეგორია. საგნის ან მოვლენის ამომწურავად აღწერაში ნაჩვენები უნდა იყოს როგორც მისი შინაგანი სტრუქტურა, ასევე მთლიანი სახე, რისი მიღწევაც მარტო დისკრეტული ან მარტო ანალოგური მიდგომის გამოყენებით შეუძლებელია; აუცილებელია ეს მიდგომები ერთმანეთს ავსებდნენ, ე.ი. ისინი ურთიერთდამატებითი მიდგომებია.

აღსანიშნავია, რომ წყვეტილობისა და უწყვეტობის კატეგორიები უმნიშვნელოვანეს როლს თამაშობენ **განვითარების პროცესის** აღწერის დროსაც. განვითარებას შეიძლება ჰქონდეს ნახტომისებური ან მემკვიდრეობითი სახე; **ნახტომს** შეესაბამება წყვეტილობის, ანუ დისკრეტულობის კატეგორია, ხოლო **მემკვიდრეობითობას** – უწყვეტობის, ანუ ანალოგურობის კატეგორია.

სამყაროს უწყვეტი ანუ ანალოგური ბუნება აქვს, მაგრამ სამყაროს აღქმა, გააზრება და მასზე ზემოქმედების მოხდენა მხოლოდ დისკრეტული (წყვეტილი) მნიშვნელობებითაა შესაძლებელი. სამყარო წარმოადგენს თავის ტვინის ქერქის მიერ იმ სუსტი ელექტრული იმპულსების (ანუ, დისკრეტული სიდიდეების) ინტერპრეტაციის შედეგს, რომელსაც იგი უმარტივესი ნერვული უჯრედების – რეცეპტორებისაგან იღებს. **“ის, რაც ჩვენ ვიცით – შეზღუდულია (ე.ი. დისკრეტულია, ა.დ.), ხოლო რაც არ ვიცით – უსასრულოა”** აღნიშნავდა ცნობილი ფრანგი მათემატიკოსი და ასტრონომი **პიერ-სიმონ ლაპლასი** (**1749 – 1827**).

2 **ანალოგური კომპიუტერი** რიცხვით მონაცემებს ანალოგური ფიზიკური სიდიდეების (სიჩქარის, სიგრძის, დენის, ძაბვის, წნევისა და ა.შ.) დახმარებით ალაწარმოებს. მისი მუშაობის შედეგებს აქვს ქაღალდზე ან ოსცილოგრამის ეკრანზე გამოსახული გრაფიკების ან გარკვეული პროცესის გასაკონტროლებლად და ფუნქციონირებისათვის საჭირო ელექტრული სიგნალების სახე. ასეთი კომპიუტერები იდეალურად მოსახერხებელია საწარმოო პროცესების ავტომატური კონტროლისათვის, რადგან ისინი მომენტალურად რეაგირებს შემავალი მონაცემების სხვადასხვა ცვლილებებზე.

ანალოგური კომპიუტერები ფართოდ შეიძლება გამოვიყენოთ ზოგიერთი სახის სამეცნიერო კვლევებში; კერძოდ, ისეთ მეცნიერებებში, რომლებშიც შესასწავლი სიტუაციების

იმიტირების უნარის მქონე **იაფი ელექტრული ან მექანიკური მოწყობილობებია** გამოყენებული. მთელ რიგ შემთხვევებში ასეთი კომპიუტერები შეიძლება ისეთ ამოცანებზე მუშაობის დროსაც გამოვიყენოთ, რომელთა ამოხსნის დროს მკაცრი სიზუსტის დაცვა სავალდებულო არ არის. კერძოდ, ისინი შეიძლება გამოვიყენოთ დიფერენციალური განტოლებების ამოსახსნელად, აგრეთვე ინტეგრირებისა და დიფერენცირებების დროს.

ანალოგური კომპიუტერების ძირითადი ღირსებებია: მაღალი სწრაფმოქმედება, მცირე მასა, გაბარიტები და ღირებულება, გამოყენების სიმარტივე (მაგალითად, მასზე მუშაობისათვის აუცილებელი არ არის დაპროგრამების ისეთი რთული ენების შესწავლა, როგორებიცაა პასკალი, C+ და ა.შ.).

ანალოგური კომპიუტერების ძირითადი ნაკლია ოპერაციების ჩატარების დროს შეზღუდული სიზუსტე; ამ დროს დაშვებული **ცდომილების სიდიდე** კომპიუტერის დამზადების ტექნოლოგიასა და ხარისხზეა დამოკიდებული; მან ცალკეულ შემთხვევებში შეიძლება 10%-საც გადააჭარბოს. ამოხსნის სიზუსტე მცირდება კომპიუტერის სტრუქტურაში არსებული გამრავლების ბლოკების რაოდენობის გაზრდის კვალობაზე.

ანალოგური კომპიუტერებისათვის დამახასიათებელი ზემოთ ფორმულირებული ძირითადი ნაკლი ხატოვნად შეიძლება ასე ჩამოვაყალიბოთ; საყოველთაოდ თუ არის ცნობილი რომ **"ორჯერ ორი უდრის ოთხს"** ე.ი. თუ უდავოა ტოლობა 2 x 2 = 4, **ანალოგური კომპიუტერისათვის** აღნიშნული ტოლობა იღებს სახეს: 2 x 2=4 ± 3%, ე.ი. ზემოთ მოყვანილი გამოთქმა ასე გარდაიქმნება: **"ორჯერ ორი დაახლოებით უდრის ოთხს"**; უკანასკნელ ტოლობაში არსებული პროცენტი შეკრების ოპერაციების მრავალჯერადად გამეორებისას იკრიბება და მისმა ჯამურმა სიდიდემ შეიძლება დაუშვებელ (წინასწარ გაუთვალისწინებელ) მნიშვნელობას მიაღწიოს.

ზემოთ აღნიშნულის გარდა **ანალოგურ სისტემას** აქვს შემდეგი ნაკლოვანებები:

- შესასრულებელი ფუნქციების გაფართოება მოითხოვს ანალოგური სისტემის სტრუქტურის შეცვლას, რაც მთელ რიგ ტექნიკურ პრობლემებთანაა დაკავშირებული;
- ფუნქციონირების ალგორითმების სირთულის გაზრდით იზრდება ანალოგური სისტემის ასაგებად საჭირო დანახარჯები;
- რთული გადასაწყვეტია სისტემის ფუნქციონირებისათვის საჭირო მონაცემების შენახვის პრობლემა;
- ანალოგური სისტემის ''დაბერება'' და ცვეთა აუარესებს მასში ჩაწერილი ინფორმაციის ხარისხს და ა.შ.

ზემოთ ჩამოთვლილი ღირსებებისა და ნაკლოვანებების გამო ანალოგური კომპიუტერების გამოყენების არეალი არ არის ფართო და მათ გამოყენება შეზღუდული მოცულობის სამეცნიერო კვლევების დროსაა მიზანშეწონილი. მათ ზოგიერთ შესაძლო გამოყენებაზე ჩვენ ზემოთ ვისაუბრეთ.

დღეს ფართოდ გავრცელებულ უნივერსალურ კომპიუტერებს წარმოადგენს. **ციფრული კომპიუტერები**; მათი აგებისათვის გამოყენებულ ტექნოლოგიას **ციფრული ტექნოლოგია** ეწოდება.

3

უწყვეტი ბუნების მქონე სამყარო ინფორმაციას საკუთარი თავის შესახებ დროში უწყვეტი სიგნალების სახით გვაწვდის; როგორც ზემოთ აღვნიშნეთ, ადამიანს მათი აღქმა და გააზრება მხოლოდ წყვეტილი მნიშვნელობებით შეუძლია. მაშასადამე, ადამიანი, როგორც ბიოლოგიური არსება, უწყვეტი სიგნალების წყვეტილ სიგნალებად გარდაქმნის უნარითაა აღჭურვილი. მეცნიერების მიერ დამუშავებული იქნა აღნიშნული უნარის ხელოვნურად რეალიზების მეთოდი, რამაც წყვეტილ სიგნალებად უწყვეტი, ანუ ანალოგური სიგნალების გარდამქმნელი ხელოვნური მოწყობილობების შექმნა გახადა შესაძლებელი. სანამ ზემოთ აღნიშნული მეთოდის არსისა და მასთან დაკავშირებული ზოგიერთი საკითხის განხილვას შევუდგებოდეთ, აღვნიშნავთ, რომ ცნება ''**წყვეტილს**'' შესაბამება ტერმინი ''**დისკრეტული**''; ეს უკანასკნელი მიღე-

ბული იქნა ლათინური სიტყვისაგან **diskretus,** რაც ქართულად ‘‘დაყოფილს’’, ‘‘დაცალკევებულს’’ ნიშნავს. შემდგომში ჩვენ მხოლოდ აღნიშნულ ტერმინს გამოვიყენებთ.

4

გარე და შიგა სამყაროდან გადმოცემული უწყვეტი ინფორმაციის ყველაზე სრულყოფილად დამამუშავებელ ბუნებრივ ქმნილებას წარმოადგენს ადამიანი. აღნიშნული ინფორმაციის მიღებას, დამახსოვრებას და აღქმას იგი დაახლოებით **100** მილიარდი **ნეირონისაგან** წარმოშობილი **ნერვული სისტემის** დახმარებით ახდენს.

ანატომიურად ადამიანის ნერვული სისტემა შედგება **ცენტრალური ნერვული** და **პერიფერიული ნერვული სისტემებისაგან**. ცენტრალური ნერვულ სისტემა მოიცავს თავის და ზურგის ტვინს, ხოლო პერიფერიული ნერევული სისტემა განკუთვნილია ცენტრალურ ნერვულ სისტემასთან სხეულის ცალკეულ ორგანოების დასაკავშირებლად.

თავის ტვინი შედგება ერთმანეთთან **სინაფსურად** (მის შესახებ ქვემოთ გვექნება საუბარი) დაკავშირებული დაახლოებით **86** მილიარდი **ნეირონისაგან.** აღნიშნული კავშირების საშუალებით ერთმანეთზე ურთიერთმოქმედების შედეგად ნეირონები წარმოშობს რთულ ელექტრულ იმპულსებს (**ნეირონულ იმპულსებს**), რომლებიც მთლიანად აკონტროლებს ადამიანის ქმედებებს. თავის ტვინის კვლევების მიმართულებით ბოლო წლებში მიღწეული მნიშვნელოვანი წარმატებების მიუხედავად თავის ტვინის ფუნქციონირების მრავალი დეტალი დღემდე ამოუცნობი რჩება. ცალკეული ნეირონების ფუნქციონირება საკმაოდ კარგადაა ახსნილი, მაგრამ ძალიან სქემატური სახითაა ახსნილი ტვინის, როგორც გარკვეული მთლიანობის, მიერ ათასობით და მილიონობით ნეირონის ურთიერთზემოქმედების ორგანიზების, სათანადო სიგნალების ფორმირებისა და მათი დამახსოვრების პროცესი. ამ სფეროში დამაკმაყოფილებელი შედეგების მიღებისათვის ჯერ კიდევ დიდი გამოკვლევებია ჩასატარებელი. რაც შეეხება **ზურგის ტვინს**, მისი დანიშნულებაა ადამიანის ტანის, მენჯის, კიდურების, მუცლისა და მკერდის ღრუს შინაგანი ორგანოების კონტროლი და, როგორც გამოკვლევებითაა დადგენილი, შედგება **31** წყვილი სპეციალური ღეროვანი ნეირონისაგან

ცენტრალურ ნერვულ სისტემაში მიმდინარე პროცესების განხილვა სცილდება მოცემულ სახელმძღვანელოში განსახილველი საკითხების ფარგლებს და ყურადღებას მხოლოდ მისთვის გადასაცემი სიგნალების გამომუშავების პროცესზე შევაჩერებთ.

პერიფერიებში განაწილებული დაახლოებით **14** მილიარდამდე **პერიფერიული ნეირონისაგან** წარმოიქმნება სხვადასხვა **ქვესისტემები (ქსელები),** რომელთა გარკვეული ნაწილი განაწილებულია ადამიანის ისეთ ორგანოებში როგორებიცაა თვალი, ყური, ენა, ცხვირი, კანი; ისინი შესაბამისად წარმოქმნის **მხედველობის, სმენის, გემოვნების, ყნოსვისა და შეხების** ორგანოებს და მათ ზოგადად **შეგრძნების ორგანოები** ეწოდება. შეგრძნების ორგანოების ძირითადი დანიშნულებებია:

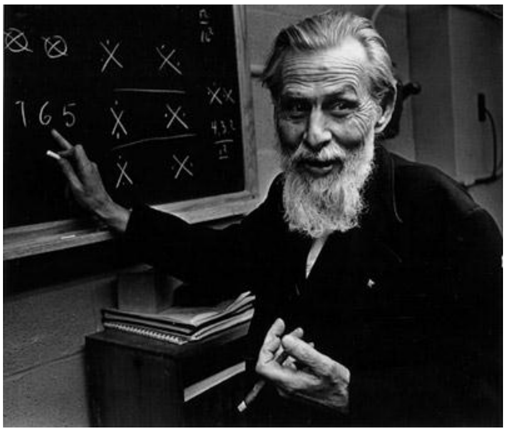
- მიიღოს გარედან მოსული სიგნალები;
- გარდაქმნას ისეთი სახის სიგნალებად, რომელთა აღქმა, დამახსოვრება და დამუშავება მოსახერხებელია თავის ტვინისათვის;
- გარდაქმნილი სიგნალები გადააწოდოს თავის ტვინს.

აღნიშნულიდან გამომდინარე, შეგრძნების ორგანოები შეიძლება განვიხილოთ, როგორც სპეციფიკური სახის ბიოლოგიური **გადამწოდები** ანუ **სენსორები** (ინგ. **sensor** – ‘‘მგრძნობიარე (აღმქმელი) ელემენტი’’, ‘‘გადამწოდი’’), რომლებიც მათზე მოქმედ სიგნალებს აწოდებს თავის ტვინს.

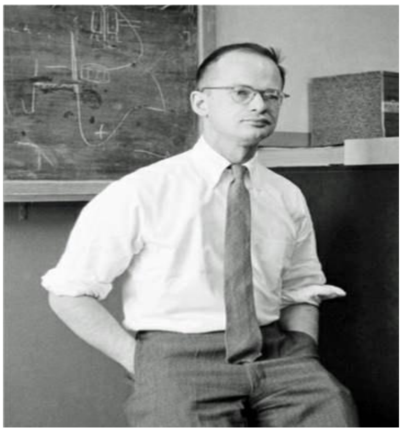
5

განვიხილოთ როგორ აღიქვამენ მიღებულ სიგნალებს ბიოლოგიური სენსორები – შეგრძნების ორგანოები და რა სახის სიგნალებად გარდაქმნიან მათ ისინი. აღნიშნული სენსორები ნეირონებითაა აგებული. **ნეირონი** (ძვ. ბერძ. **νεῦρον** — ‘‘ბოჭკო’’, ‘‘ნერვი’’) წარმოადგენს ნერვიული სისტემის სტრუქტურულ-ფუნქციონალურ ერთეულს, რომელსაც ზოგჯერ სხვადასხვა ავტორები ნევრონის, ნევროციტისა და ნერვიული უჯრედის სახელწოდებითაც მოიხსენებენ.

ნეირონული ქსელის უმსხვილესი თეორეტიკოსია **უორენ მაკ-კალოხი (Warren McCulloch)**, რომელიც **ნორბერტ ვინერთან** ერთად ითვლება კიბერნეტიკის "მამად".



უორენ მაკ-კალოხი
(1898-1969)



ვალტერ პიტსი
(1923-1969)

უორენ მაკ-კალოხმა და მისმა მოსწავლემ **ვალტერ პიტსმა** ჩამოაყალიბეს ჰიპოთეზა, რომლის თანახმადაც ნეირონი ორობით რიცხვებზე ოპერაციების ჩამტარებელ ელექტრონულ მოწყობილობად განიხილება. მათ შემოგვთავაზეს ხელოვნური ნეირონის მოდელი, რომელთა მეშვეობით აგებულ ქსელს, როგორც ქვემოთ ვაჩვენებთ, შეუძლია აღქმული უწყვეტი სიგნალები გარდაქმნას ორობით რიცხვებად. **მაკ-კალოხმა** და **პიტსმა** აგრეთვე დაასაბუთეს, რომ ხელოვნური ნეირონებისაგან აგებულ ქსელს პოტენციურად **ნებისმიერი** არითმეტიკული (რიცხვითი) და ლოგიკური ოპერაციების შესრულების უნარი გააჩნია.

ზოგადად, **უორენ მაკ-კალოხისა** და **ვალტერ პიტსის** მიერ მიღებული იქნა შემდეგი შედეგები:

- დაამუშავეს ნეირონის, როგორც უმარტივესი პროცესორული ელემენტის მოდელი, რომელიც შესასვლელი სიგნალების ვექტორისა და წონითი კოეფიციენტების სკალარული გადამრავლების გზით გამოითვლის **გადაცემით ფუნქციას** (**გადაცემით ფუნქცია** გვიჩვენებს ნეირონის გამოსასვლელი სიგნალი თუ როგორაა დამოკიდებული აღნიშნული ნეირონის შესასვლებზე მოქმედი სიგნალების შეწონილ ჯამზე);
- შემოგვთავაზეს ნეირონებისაგან აგებული ქსელის კონსტრუქცია ლოგიკური და არითმეტიკული ოპერაციების შესასრულებლად;
- გამოთქვეს დასაბუთებული ვარაუდი იმის შესახებ, რომ მათ მიერ ზემოთ აღნიშნულ ქსელს გააჩნია სწავლის, სახეების ამოცნობისა და მიღებული ინფორმაციის განზოგადების უნარი.

უორენ მაკ-კალოხისა და ვალტერ პიტსის მიერ დაწყებული კვლევები შემდგომში გააგრძელა ამერიკელმა მათემატიკოსმა **სტივენ კოულ კლინმა**. მან **უორენ მაკ-კალოხისა** და **ვალტერ პიტსის** მიერ უმარტივესი ავტომატის სახით წარმოდგენილი ნეირონის მოდელი აღწერა **რეგულარულ სიმრავლედ** წოდებული საკუთარი მათემატიკური სისტემის გამოყენებით; თეორიული ინფორმატიკისათვის დაამუშავა სპეციალური ალგებრული სტრუქტურა, რომელსაც **რეგულარული გამოსახულებების ალგებრა** ანუ **კლინის ალგებრა** ეწოდება. როგორც ჩვენს კვლევებში ვაჩვენეთ [4; 5], რეგულარული გამოსახულებების გამოყენება მნიშვნელოვნად ამარტივებს გარკვეული კლასის დისკრეტული მოწყობილობების აგების პროცესს.

6

მაკ-კალოხისა და **პიტსის** დაშვებების შესაბამისად, ნერვიული უჯრედი ანუ **ნეირონი** შედგება სხეულისაგან (ე.წ. სომასგან), საიდანაც **აქსონებად** (ძვ.ბერძ. **ἀξον** - "ღერძი") წოდებული ამოზრდილი ნერვული ბოჭკოები ერთ ან რამდენიმე **ბოლო ფირფიტას** უერთდება. ბოლო ფირფიტები იყოფა **მაპოლარიზებელ** და **დემაპოლარიზებელ** ფირფიტად (ერთი ბოლო ფირფიტა ერთდროულად არ შეიძლება იყოს როგორც მაპოლარიზებელი, ასევე დემაპოლარიზებელიც).

1.8 ნახაზზე **ნეირონის სხეული (სომა)** პირობითად პატარა სამკუთხედითაა აღნიშნული, მისგან ამოზრდილი **აქსონი** – სამკუთხედიდან ამოსული წრფის მონაკვეთით, რომელიც ბოლო ფირფიტასთანაა შეერთებული; ამასთანავე **მაპოლარიზებელი ბოლო ფირფიტა** მსხვილი წერტილით, ხოლო **დემაპოლარიზებელი ბოლო ფირფიტა** – პატარა წრით არის აღნიშნული.

სტივენ კოულ კლინი
(1909-1994)

ნერვული ქსელი წარმოადგენს სასრული რაოდენობის ნეირონებისაგან შედგენილ ბიოლოგიურ "მოწყობილობას", რომელშიც შემავალი ნებისმიერი ნეირონის **ბოლო ფირფიტა** ეხება ერთზე არა უმეტეს (იმავე ან სხვა) **ნეირონის სხეულს (სომას)**, რომელსაც იგი აპოლარიზებს; აღნიშნული ფირფიტისა და სხეულის შეხების ადგილზე წარმოშობილ კონტაქტს **სინაფსი** (ბერძ. **σύναψις, συνάπτειν —** "შემოხვევა"; "ხელის მოჭერა"; "გადახვევა) ეწოდება.

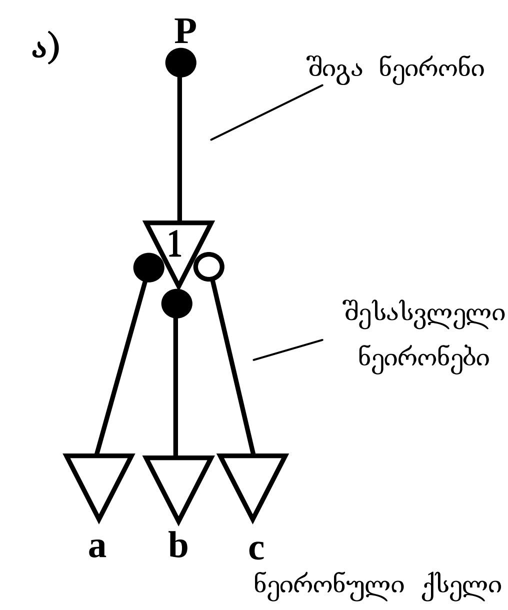
ქსელში შემავალ ნეირონებს, რომელთა სხეულებსაც არ ეხება არც ერთი ბოლო ფირფიტა, **შესასვლელი ნეირონები** ეწოდება (მათი რაოდენობა კერძო შემთხვევებში შეიძლება ნულის ტოლიც იყოს), ხოლო დანარჩენ ნეირონებს – **შიგა ნეირონები** ეწოდება.

დროის ერთმანეთისაგან თანაბრად დაშორებულ მომენტებში (რომლებიც დროით სკალაზე არსებულ მთელ რიცხვებად შეიძლება განვიხილოთ) ქსელის თითოეული ნეირონი შეიძლება იყოს პოლარიზებულ ან დეპოლარიზებულ (მშვიდ) მდგომარეობაში. პოლარიზებულ მდგომარეობაში ყოფნის დროს ნეირონი გამოიმუშავებს **ნერვულ იმპულსს,** ხოლო დეპოლარიზებულ მდგომარეობაში ყოფნის დროს აღნიშნულ იმპულსს არ გამოიმუშავებს.

დროის ნებისმიერ **t** მომენტში **შესასვლელი ნეირონის** პოლარიზება ან დეპოლარიზება განისაზღვრება ქსელის მიმართ გარე პირობებით. შეიძლება ვივარაუდოთ, რომ თითოეულ მათგანთან მიერთებულია მგრძნობიარე აღმქმელი ორგანო (მაგალითად, რეცეპტორი), რომელიც გარემომცველ არეში სათანადო პირობების შემთხვევაში იწვევს შესასვლელი ნეირონის პოლარიზებას.

შიგა ნეირონისათვის დროის **t** მომენტში პოლარიზებისათვის მის სხეულზე (სომაზე) **t-1** მომენტში უნდა ზემოქმედებდეს **h**-ზე ($\mathbf{h} \in \mathbf{N}$, სადაც **N** ნატურალური რიცხვების სიმრავლეა) არანაკლები რაოდენობის მაპოლარიზებელი ფირფიტა და არც ერთი დემაპოლარიზებელი ფირფიტა; მოცემულ შემთხვევაში **h**-ს ნეირონის **ზღურბლი** ეწოდება; ნახაზზე შიგა ნეირონის **ზღურბლის მნიშვნელობა** მისი **სხეულის (სომას)** გამომხატველ პატარა სამკუთხედშია ჩაწერილი. შიგა ნეირონი პოლარიზდება, თუ პოლარიზებულია მისი ზღურბლის ტოლი ან მეტი რაოდენობის მაპოლარიზებელი შესასვლელი ნეირონი და

არ არის პოლარიზებული არც ერთი არც დემაპოლარიზებელი ნეირონი, რომლებიც მის სხეულთან (სომასთან) სინაფსებს წარმოქმნის.



ბ)

a	b	c	P
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

ქსელით რეალიზებული ფუნქციის ჭეშმარიტობის ცხრილი

ნახ. 1.12. ნეირონული ქსელის მიერ ორობითი ფუნქციის ფორმირების მაგალითი

1.12 ნახაზზე ნაჩვენები უმარტივესი ნერვული ქსელი, რომელიც შეიცავს სამ შესასვლელ და ერთ შიგა ნეირონს (რეალურად აღნიშნული ნეირონების რაოდენობა გაცილებით მეტია). შესასვლელი ნეირონებია **a; b** და **c,** ხოლო შიგა ნეირონია – **P** ნეირონი. შესასვლელი ნეირონებიდან მაპოლარიზებელია **a; b**, ხოლო დემაპოლარიზებელი – **c** ნეირონი. შიგა **P** ნეირონის ზღურბლი **1**-ის ტოლია, ე.ი. **h** = 1 (რეალურ ნეირონულ ქსელებში შიგა ნეირონის ზღურბლი ყოველთვის **1**-ზე მეტია, მაგრამ მოცემული დაშვება ამარტივებს გადმოცემას).

განხილულ შემთხვევაში დროის **t** მომენტში შიგა **P** ნეირონის პოლარიზებისათვის დროის **t-1** მომენტში პოლარიზებული უნდა იყოს **a; b** ნეირონებიდან ერთ-ერთი ან ორივე და არ იყოს პოლარიზებული **c** ნეირონი. ნეირონის დეპოლარიზებული მდგომარეობა აღინიშნება ციფრით **0,** ხოლო პოლარიზებული მდგომარეობა – ციფრით **1.**

შიგა **P** ნეირონის მდგომარეობას განსაზღვრავს შესასვლელი **a; b** და **c** ნეირონების მდგომარეობები, ამიტომ **P** ნეირონის მდგომარეობით რეალიზდება გარკვეული ფუნქცია, რომლის არგუმენტებია **a; b; c** ნეირონების მდგომარეობები; მათემატიკურად აღნიშნული ფუნქცია ჩაიწერება როგორც **P = f(a,b,c).** ვინაიდან მოცემულ ფუნქციაში როგორც არგუმენტები, ასევე თავად ფუნქცია იღებს მხოლოდ ორ, კერძოდ **0-ის** ან **1-ის** ტოლ, მნიშვნელობებს (**a; b; c**; **P** $\in$ **(0,1)**), ამიტომ მას **ორობითი ფუნქცია ეწოდება**. ორობით ფუნქციას, როგორც შემდგომში ვაჩვენებთ**, ლოგიკურ** ან **ბულის ფუნქციასაც** უწოდებენ.

ჩვენ მიერ განხილული **P = f(a,b,c)** ფუნქცია **a,b,c** არგუმენტების მნიშვნელობათა თითოეულ ნაკრებზე გარკვეულ მნიშვნელობას მიიღებს; კერძოდ, **010** ნაკრებზე იგი მიიღებს **1-ის** ტოლ მნიშვნელობას რაც ნიშნავს შემდეგს: “მაპოლარიზებული **a; b** ნეირონებიდან თუ პოლარიზებულია **b** და არაა პოლარიზებული **a** ნეირონი (ე.ი. ასეთი სახის პოლარიზებული ნეირონების რაოდენობა **P** ნეირონის ზღურბლის ტოლია) და არაა პოლარიზებული დემაპოლარიზებელი **c** ნეირონი, მაშინ პოლარიზებული იქნება შიგა **P** ნეირონი”.

P = f(a,b,c) ფუნქცია **a,b,c** არგუმენტების თითოეულ შესაძლო მნიშვნელობას თუ მივუწერთ **P** არგუმენტის მნიშვნელობას, მივიღებთ **1.12,ბ** ნახაზზე ნაჩვენებ ცხრილს. ორობითი ცვლადების **1-ის** ტოლ მნიშვნელობას მათი **ჭეშმარიტი მნიშვნელობა,** ხოლო **0**-ის ტოლ მნიშვნელობას – **ყალბი მნიშვნელობა** ეწოდება. რადგან ჭეშმარიტების ძიება კაცობრიობის ერთი მთავარი მიზანთაგანია და **1.12,ბ** ცხრილის მსგავსი ცხრილები თვალის ერთი გადავლებით გვაძლევს განსახილველი ფუნქციების ჭეშმარიტი მნიშვნელობების პოვნის საშუალებას, ამიტომ მათ პირობითად **ჭეშმარიტობის ცხრილი** უწოდეს (თუმცა ანალოგური მსჯელობის ძალით მათთვის შესაძლებელია **სიყალბის ამომცნობი ცხრილებიც** ეწოდებინათ, რადგან ისინი ფუნქციის ყალბი მნიშვნელობების ამოცნობის საშუალებასაც გვაძლევს).

ზემოთ აღნიშნულის საფუძველზე შეიძლება დავასკვნათ, რომ **1.12,ბ** ნახაზზე მოყვანილია **1.12,ა** ნახაზზე ნაჩვენები ნეირონული ქსელის მიერ რეალიზებული ფუნქციის **ჭეშმარიტობის ცხრილი.**

დასასრულს შევნიშნავთ, რომ ზემოთ გამოყენებული ტერმინები 1897 წელს შემოიღო ინგლისელმა ფიზიოლოგმა **ჩარლზ სკოტ შერინგტონმა** (***Charles Scott Sherrington,* 1857–1952*)***, რომელსაც ნეირონების ფუნქციასთან დაკავშირებული აღმოჩენის გამო ბრიტანელ ელექტროფიზიოლოგ **ედგარ დუგლას ედრიანთან** (***Edgar Douglas Adrian***, 1889–1977) ერთად 1932 წელს ნობელის პრემია მიენიჭა.

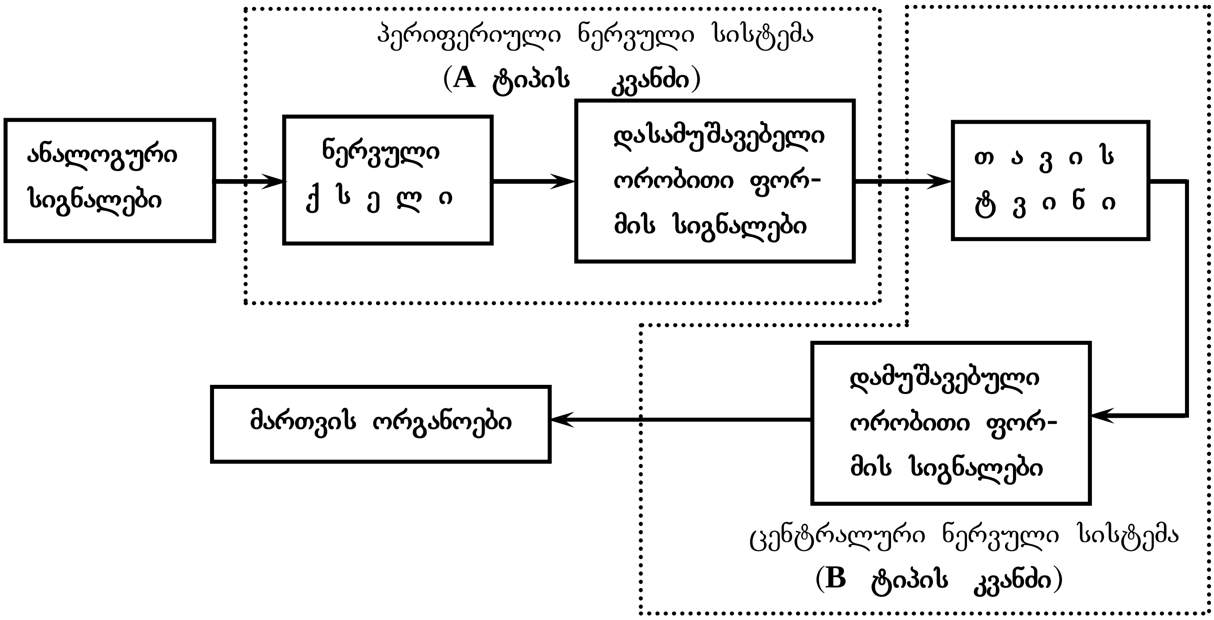
7 ნულებისა და ერთიანებისაგან შემდგარ რიცხვებს **ორობითი რიცხვები** ეწოდება. ზემოთ აღნიშნულის თანახმად, ადამიანი მის გრძნობის ორგანოებზე მოქმედ უწყვეტ (ანალოგურ) სიგნალებს ორობითი რიცხვების ფორმით წარმოდგენილ დისკრეტულ სიგნალებად აღიქვამს და მათ დასამუშავებლად გადასცემს თავის ტვინს.

მაქსიმალურად გამარტივებული მსჯელობის თანახმად თავის ტვინი:

- იმახსოვრებს მიღებულ ორობითი რიცხვების მქონე სიგნალებს, რომლებსაც პირობითად **დასამუშავებელი ორობითი სიგნალები** ვუწოდოთ;
- გამოიმუშავებს დასამუშავებელ ორობით რიცხვებზე ჩასატარებელი ოპერაციების განმსაზღვრელ ინსტრუქციათა (ბრძანებათა) მიმდევრობებს, რომელთა ერთობლიობები თავისებურ **ბუნებრივ სამოქმედო პროგრამებს** წარმოქმნის;
- ფორმირებული სამოქმედო პროგრამების რეალიზების გზით პირველად ორობით სიგნალებს გარდაქმნის ასევე ორობით სიგნალებად, რომლებსაც პირობითად **დამუშავებული ორობითი სიგნალები** ვუწოდოთ;
- დამუშავებული ორობითი სიგნალები გადაეცემა ადამიანის მართვის ორგანოებს და ხდება მათი მართვა.

ზემოთ აღწერილი პროცედურა, რომლის ბლოკ-სქემა **1.13** ნახაზზეა წარმოდგენილი, ვერ ასახავს ადამიანის ორგანიზმის ნერვულ სისტემაში მიმდინარე ურთულეს ფიზიოლოგიურ პროცესებს, მაგრამ ნათლად გამოხატავს იმ რეალობას, რომ აღნიშნული სისტემა ოპერირებს ორობითი რიცხვების სახით წარმოდგენილ დისკრეტულ, ე.ი დისკრეტულ სიგნალებზე. მასზე ნაჩვენები **A** ტიპის კვანძის დანიშნულებაა შეგრძნების ორგანოებზე მოქმედი ანალოგური სახის სიგნალები გარდაქმნას ტვინისთვის დასამუშავებლად მოსახერხებელი სახის დისკრეტიულ სიგნალებად, ხოლო **B** ტიპის კვანძის დანიშნულებაა გამოიმუშაოს ადამიანის შემსრულებელი ორგანოების მმართველი ორობითი სიგნალები.

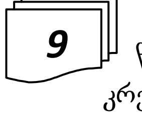
8 ზემოთ აღნიშნულის თანახმად, ადამიანის ნერვული სისტემა გამოიყენებს არა ანალოგურ, არამედ დისკრეტულ (ორობითი რიცხვების სახით წარმოდგენილ) სიგნალებს.



ნახ. 1.13. ადამიანის შეგრძნების ორგანოებზე მოქმედი ანალოგური სიგნალების ორობითი რიცხვების ფორმის დისკრეტილ სიგნალებად გარდაქმნის გამარტივებული ბლოკ-სქემა

ორობითი სახის სიგნალების გამოყენების ფაქტით ბუნება გვასწავლის, რომ სწორედ ასეთ რიცხვებად გარდაქმნილი ინფორმაციაა მოსახერხებელი როგორც დასამახსოვრებლად, ასევე დამუშავებისათვის. ‘‘ბუნებაში ყველაფერი ბრძნულადაა მოწყობილი, ყველა თავის საქმით უნდა იყოს დაკავებული და ამ სიბრძნეშია განივთებული ცხოვრების უზენაესი სამართლიანობა’’ აღნიშნავდა უდიდესი იტალიელი მხატვარი და მეცნიერი **ლეონარდო და ვინჩი,** ხოლო ინგლისელი ფილოსოფოსი **ფრენსის ბეკონი (1561-1686)** გვაფრთხილებდა, რომ **‘‘ბუნება ემორჩილება იმას, ვინც თავად ემორჩილება ბუნებას’’.**

ბუნებისადმი ბრძნული დამორჩილების ერთ-ერთ გამოხატულებას შეიძლება მივაკუთვნოთ ადამიანის მიერ **ციფრული ტექნოლოგიის** დამუშავების ფაქტი. საინტერესოა ტერმინ ‘‘ციფრული ტექნოლოგიის’’ ეტიმოლოგია. სიტყვა ‘‘**ციფრულის**’’ ინგლისური შესატყვისი ფორმა ‘‘**digital**’’ მიღებულია ლათინური სიტყვისაგან **‘‘digitus’’**, რაც **‘‘თითს’’** ნიშნავს. ვინაიდან ადამიანები მცირე რაოდენობის საგნების თვლისათვის თითებს იყენებდნენ, ამიტომ სწორედ თითები იქნა მიღებული ციფრების სინონიმებად. ჩვეულებრივ, თითებით მხოლოდ მთელი (ნატურალური) რიცხვების დათვლაა შესაძლებელი. აქედან გამომდინარე სიტყვა **‘‘ციფრული’’** (‘‘**digital**’’) გამოიყენება ნებისმიერი ობიექტის აღსანიშნავად, რომლებიც თავისი მუშაობისას **დისკრეტულ** მნიშვნელობებს იყენებს.



ციფრული ტექნოლოგია (ინგ. ***Digital technology***) ეფუძნება სიგნალების წარმოდგენას არა უწყვეტი სპექტრის, არამედ ანალოგური დონეების დისკრეტული ზოლების სახით. თითოეული ზოლის ფარგლებში არსებული ყველა დონე სიგნალის ერთნაირ მდგომარეობას წარმოადგენს. ანალოგური ტექნოლოგიისაგან განსხვავებით ციფრულ ტექნოლოგიაში ხდება არა უწყვეტი, არამედ დისკრეტული სიგნალების დამუშავება. გარდა ამისა, სასურველია სიგნალის გამოსახვისათვის მცირე რაოდენობის მნიშვნელობათა ნაკრები იყოს გამოყენებული. რეალურად აღნიშნული რაოდენობა ორის ტოლია და ისინი აღინიშნება ციფრებით: **0** და **1.** აღნიშნული ციფრებით გამოხატულ სიგნალებს **ორობითი სიგნალები** ეწოდება.

ციფრული ტექნოლოგიით დამზადებულ სისტემას, ე.ი. სისტემას, რომელშიც ორობითი სიგნალები გამოიყენება, **ციფრული სისტემა** ეწოდება. ციფრული სისტემების უპირატესობებია შემდეგი:

● ციფრული სიგნალები შეიძლება დაუმახინჯებლად გადაიცეს. მაგალითად, ერთიანებისა და ნულიანების მიმდევრობათა სახით გადაცემული უწყვეტი ბგერითი სიგნალები უშეცდომოდ შეიძლება აღვადგინოთ, თუ გადაცემის პროცესში წარმოშობილი ხელშეშლები აღნიშნული ციფრების იდენტიფიცირების შესაძლებლობას მოგვცემს;

● კომპიუტერული ციფრული სისტემები პროგრამული უზრუნველყოფის დახმარებით იმართება, რის გამოც მათი ფუნქციების გაფართოება-შეცვლა აპარატურული საშუალებების შეუცვლელად შეიძლება მოვახდინოთ. ძალიან ხშირად ეს შესაძლებელია სისტემების დამამზადებელი ქარხნების მონაწილეობის გარეშე, უბრალოდ პროგრამული უზრუნველყოფის განახლების გზით მოვახდინოთ;

● რეალიზებული ალგორითმების სირთულის გაზრდით არ რთულდება ციფრული სისტემების კონსტრუქცია (ანალოგურ სისტემებში კი აღნიშნული სისტემების კონსტრუქცია რთულდება);

● ციფრულ სისტემებში ინფორმაცია გაცილებით უფრო მარტივად შეიძლება შევინახოთ, ვიდრე ანალოგურ სისტემებში; ციფრული სისტემებისადმი დამახასიათებელი დაბრკოლებამდგრადობა საშუალებას გვაძლევს მონაცემები დაუმახინჯებლად შევინახოთ მეხსიერებაში და ასევე დაუმახინჯებლად ამოვიღოთ მეხსიერებიდან.

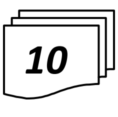
ციფრულ სისტემებს ახასიათებს შემდეგი ნაკლოვანებები:

● ზოგიერთ შემთხვევებში ციფრული სისტემები იმაზე მეტ ენერგიას მოიხმარს, ვიდრე იმავე ამოცანის გადასაწყვეტად ანალოგურ სისტემას დასჭირდებოდა; ამიტომ ანალოგურ სისტემებთან შედარებით ისინი უფრო მეტ სითბოს გამოყოფენ და სისტემაში **ქულერის** (ციფრული სისტემის გაგრილების სისტემის) გამოყენება ხდება საჭირო. ეს გარემოება ზღუდავს ბატარეებიდან მკვებავ პორტატულ მოწყობილობებში ციფრული სქემების გამოყენებას. მაგალითად, აღნიშნულის გამო საბაზისო სადგურიდან მოსული რადიოსიგნალების გასაძლიერებლად და ასაწყობად მობილურ ტელეფონებში ხშირად იყენებს მცირე სიმძლავრიან ანალოგურ ინტერფეისებს;

● ციფრული სქემები ზოგჯერ ანალოგურ სქემებზე ძვირია;

● ციფრულ სიგნალად ანალოგური სიგნალის გარდაქმნის დროს შეიძლება არსებობდეს ინფორმაციის დანაკარგები. მათემატიკურად ეს მოვლენა შეიძლება აღვწეროთ, როგორც **დამრგვალების შეცდომა.**

● ზოგიერთ სისტემაში ციფრული მონაცემების რომელიმე ფრაგმენტის დაკარგვის ან გაფუჭების დროს შეიძლება მთლიანად შეიცვალოს მონაცემთა დიდი ბლოკების შინაარსი.



კომპიუტერული სისტემა წარმოადგენს ინფორმაციის პროგრამულად დამუშავებისათვის ადამიანის მიერ შექმნილ ხელოვნურ მოწყობილობას. ბუნებრივი გამომთვლელი სისტემის ანალოგიურად მასაც უნდა ჰქონდეს **1.13** ნახაზზე ნაჩვენები ბუნებრივი **A** და **B** ტიპის კვანძების ანალოგური ხელოვნურად სინთეზირებული კვანძები, რომლებიც შესაბამისად $\mathbf{A^{(1)}}$ და $\mathbf{B^{(2)}}$ სიმბოლოებით აღვნიშნოთ.

$\mathbf{A^{(1)}}$ ტიპის კვანძის დანიშნულებაა აღიქვას გარედან მოსული ანალოგური სიგნალი და იგი გარდაქმნას ორობით სიგნალად, რომლის დამახსოვრება და დამუშავება მოსახერხებელი იქნება $\mathbf{B^{(2)}}$ ტიპის კვანძში არსებული სათანადო მოწყობილობისათვის. აღნიშნულიდან გამომდინარე მოცემულ კვანძი უნდა შეიცავდეს (**ნახ.1.14**):

● ანალოგური სიგნალის აღმქმელ სენსორს;

● აღქმული ანალოგური სიგნალის დისკრეტულ სიგნალად გარდამქმნელ მოწყობილობას ანუ **ანალოგურ-ციფრულ გარდამქმნელს** (**აცგ**-ს).

● ფიქსირებული დისკრეტული სიგნალის ორობითი კოდებით მაკოდირებელ მოწყობილობას, რომელსაც პირობითად **ორობითი კოდერი** ვუწოდოთ.

ცხადია, რომ კომპიუტერის შესასვლელზე თუ რაიმე დისკრეტული სიგნალი მოქმედებს, იგი **ACG**-ს გვერდის ავლით უშუალოდ ორობით კოდერს მიეწოდება.

$\mathbf{B^{(1)}}$ ტიპის კვანძში თავმოყრილმა მოწყობილობებმა უნდა უზრუნველყს:

● $\mathbf{A^{(1)}}$ ტიპის კვანძიდან მიღებული ორობითი სიგნალების დამახსოვრება; ამისათვის აღნიშნული სიგნალები უნდა ჩაიწეროს სპეციალურ **მახსოვრობის მოწყობილობაში**, ანუ შემოკლებით **მეხსიერებაში**. მეხსიერებაში ჩაწერილ აღნიშნულ ორობით სიგნალებს **საწყისი მონაცემები** ეწოდება;

● მონაცემების დამუშავებისათვის საჭირო ინსტრუქციების (ბრძანებების) ფორმირება, რომელთა ერთობლიობას **კომპიუტერული პროგრამა** ეწოდება;

● კომპიუტერული პროგრამის შესაბამისად მონაცემების დამუშავება და ახალი ორობითი სიგნალების ფორმირება;

● ფორმირებული ორობითი სიგნალების დამახსოვრება, ანუ მეხსიერებაში ჩაწერა, რის შედეგადაც ისინი ახალ მონაცემებად გადაიქცევა;

● წინა პუნქტში აღნიშნული მონაცემები იყოფა საშუალედო და საბოლოო მონაცემებად. **საშუალედო მონაცემებს** კომპიუტერული პროგრამა გამოიყენებს შემდგომი გამოთვლების ორგანიზებისათვის, ხოლო **საბოლო მონაცემები** გადაეცემა შემსრულებელ მოწყობილობებს მართვისათვის.

თავის ტვინს გააჩნია მის მიერ დამახსოვრებული ორობითი სახის დისკრეტული სიგნალების დასამუშავებლად საჭირო ინსტრუქციების (ბრძანებების) თვითინიცირების უნარი, რასაც მოკლებულია ადამიანის მიერ სინთეზირებული ხელოვნური მოწყობილობები.

ამიტომ მეხსიერებაში არსებული მონაცემების დასამუშავებლად საჭირო კომპიუტერული პროგრამა ჩვენ უნდა დავამუშავოთ და მისი შემადგენელი ინსტრუქციები ჩავწეროთ $\mathbf{B^{(1)}}$ ტიპის კვანძში არსებულ მეხსიერებაში; იბადება კითხვა, ამისათვის შეიძლება თუ არა გამოვიყენოთ აღნიშნულ კვანძში არსებული ის მეხსიერება, რომელშიც მონაცემებია ჩაწერილი.

ჰარვარდის უნივერსიტეტის პროფესორის **ჰოვარდ ეიკენის** ხელმძღვანელობით **1941** წელს კონსტრუირებულ გამომთვლელ მანქანა **Mark I**-ში მონაცემები და პროგრამის ინსტრუქციები (ბრძანებები) სხვადასხვა მეხსიერების მოწყობილობებში იყო ჩაწერილი. ასეთი არქიტექტურის კომპიუტერებს შემდეგ **ჰარვარდული არქიტექტურის კომპიუტერები** ეწოდა.

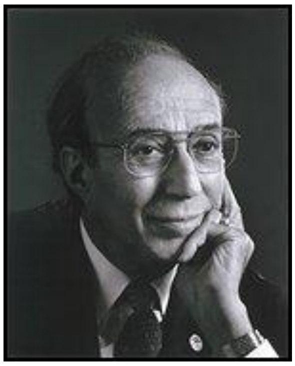
1946 წელს პრისტონის უნივერსიტეტის მეცნიერებმა **არტურ ბიორკსმა** (Arthur Burks), **ჰერმან გოლდსტაინმა** და **ჯონ ფონ ნეიმანმა** გამოაქვეყნეს სტატია **"Preliminary Discussion of the Logical Design of an Electronic Computing Instrument"** (''ელექტრონული გამომთვლელი მოწყობილობის ლოგიკური კონსტრუირების წინასწარი განხილვა''), რომელშიც სხვა წინადადებებთან ერთად (მათ ჩვენ ცალკე განვიხილავთ), წამოაყენეს მონაცემებისა და პროგრამების ერთსა და იგივე მეხსიერებაში შენახვის იდეა. აღნიშნული ავტორებიდან ყველაზე ცნობილ მეცნიერს **ჯონ ფონ ნეიმანი** წარმოადგენდა, ამიტომ სტატიაში ფორმირებული წინადადებების გამოყენებით კონსტრუირებულ კომპიუტერებს **ფონ ნეიმანის არქიტექტურის** მქონე კომპიუტერები უწოდეს. ზოგიერთი ავტორი ტერმინ "**ფონ ნეიმანის არქიტექტურის"** ნაცვლად პრისტონის უნივერსიტეტის საპატივცემლოდ, სადაც მუშაობდნენ სტატიის ავტორები, ხმარობს ტერმინს **"პრისტონული არქიტექტურა".**

ფონ ნეიმანისეული არქიტექტურის მქონე პირველ კომპიუტერია **1948** წლის **21** ივნისს მანჩესტერის უნივერსიტეტში (დიდი ბრიტანეთი) ამოქმედებული კომპიუტერი "**მანჩესტერული Mark I**", რომელიც ზემოთ აღნიშნული კომპიუტერ **Марк I**-ის პროტოტიპს წარმოადგენს. ასეთივე არქიტექტურის მქონე მეორე კომპიუტერი **EDSAC** (ინგლ.***E****lectronic* ***D****elay* ***S****torage* ***A****utomatic* ***C****omputer – ელექტრონული შეყოვნებიანი*

ავტომატური კომპიუტერი) **1949** წლის **6** მაისს აამუშავეს კემბრიჯის უნივერსიტეტის მეცნიერთა ჯგუფმა. თანამედროვე კომპიუტერების აბსოლუტურ უმრავლესობას ფონ ნეიმანისეული არქიტექტურა გააჩნიათ, თუმცა არსებობენ **ჰიბრიდული არქიტექტურის კომპიუტერებიც,** რომლებშიც ჰარმონიულადაა შერწყმული ჰარვარდული და ფონ ნეიმანისეული (პრისტონული) არქიტექტურის ცალკეული ელემენტები.

ჯონ ფონ ნეიმანი
(1903 – 1957)

არტურ ბიორკსი
(1915 – 2008)



ჰერმან გოლდსტაინი
(1913 – 2004)

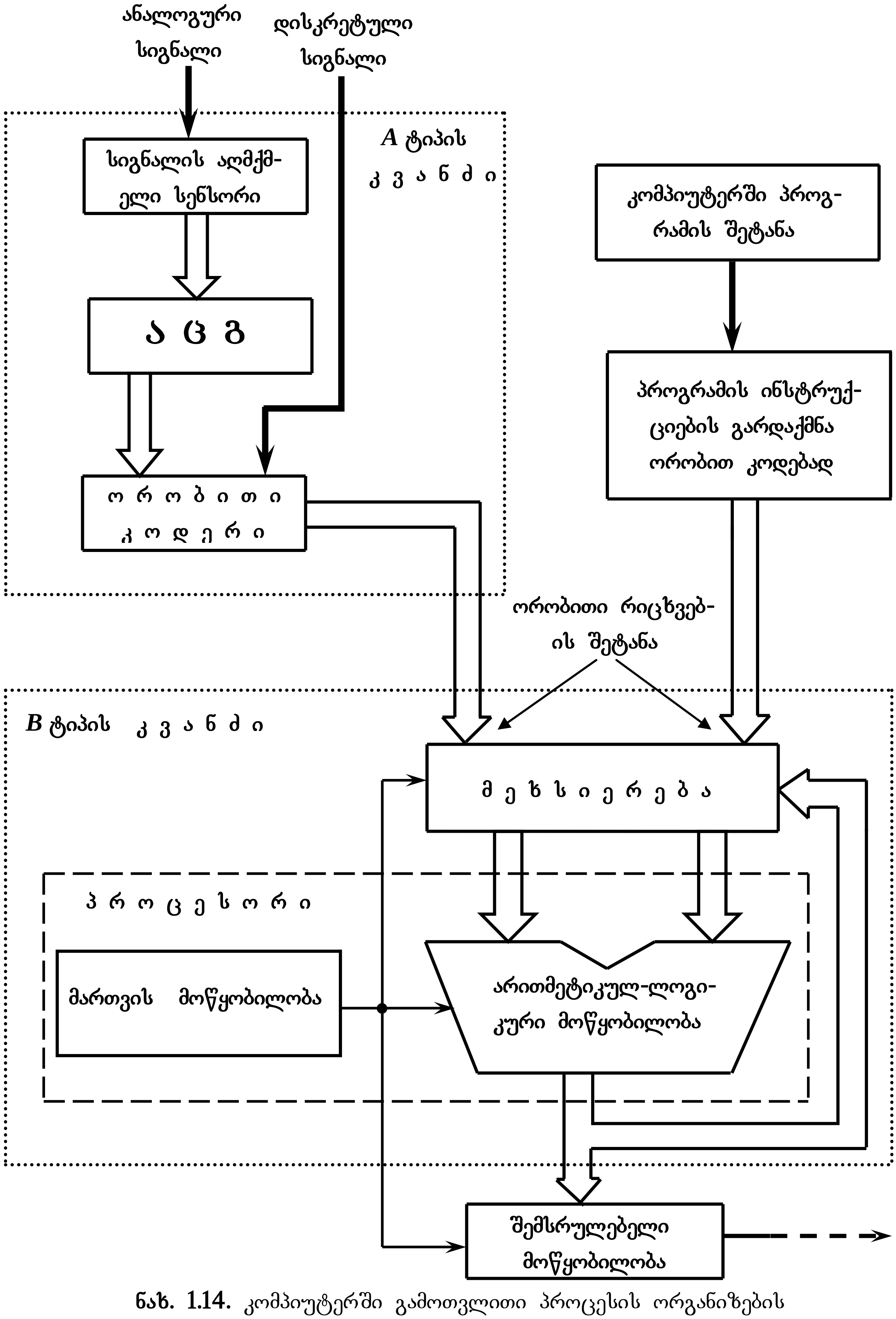
1.14 ნახაზზე მოყვანილ სტრუქტურაში ფონ ნეიმანისეული არქიტექტურაა გამოყენებული. როგორც აღნიშნული ნახაზიდან ჩანს, დამპროგრამებლემის მიერ შედგენილი კომპიუტერული პროგრამის ინსტრუქციები სათანადო ორობითი კოდებით კოდირების, ე.ი. ორობით რიცხვებად გარდაქმნის შემდეგ ჩაიწერება იმავე მეხსიერებაში, რომელშიც ორობითი რიცხვების სახის მონაცემებია ჩაწერილი. მაშასადამე, **მეხსიერება** წარმოადგენს ორობითი რიცხვების სპეციფიკურ "საწყობს", რომელთაგანაც ორობითი რიცხვების ერთი ნაწილი მონაცემებს წარმოადგენს, ხოლო მეორე ნაწილი – კომპიუტერული პროგრამების ინსტრუქციებს.

საშუალედო და საბოლოო მონაცემები მიიღება საწყის მონაცემებზე კომპიუტერის პროგრამის ინსტრუქციებით გათვალისწინებული არითმეტიკულ-ლოგიკური ოპერაციების შესრულების გზით. აღნიშნულ ოპერაციებს ასრულებს სპეციალურად კონსტრუირებული **არითმეტიკულ-ლოგიკური მოწყობილობა,** რომლის ფუნქციონირებასაც ხელმძღვანელობს **მართვის მოწყობილობა**.

კომპიუტერის პროგრამა წარმოადგენს გარკვეული შინაარსის მქონე **პასიური ინსტრუქციების (ბრძანებების) პასიურ ერთობლიობას**. აღნიშნული ერთობლიობის გააქტიურება ნიშნავს მასში შემავალი ინსტრუქციებით გათვალისწინებული ქმედებების შესრულების დაწყებას.

კომპიუტერულ პროგრამაში შემავალი თითოეული ინსტრუქციის რეალიზებისათვის საჭირო ქმედებათა მიმდინარეობას **პროცესი**, ხოლო აღნიშნულ პროცესში მონაწილე მოწყობილობების ერთობლიობას **პროცესორი** ეწოდება.

1.14 ნახაზზე ნაჩვენები ხელოვნურად გამარტივებული შემთხვევისათვის პროცესორი მხოლოდ არითმეტიკულ-ლოგიკურ მოწყობილობასა და მართვის მოწყობილობას შეიცავს. რეალური კომპიუტერული სისტემების პროცესორს გაცილებით რთული სტრუქტურა აქვს, მაგრამ სტრუქტურის სირთულის ზრდა არ ცვლის მის ძირითად დანიშნულებას, რომელსაც ზემოთ აღნიშნული პროცესის ჩატარება წარმოადგენს.



ნახ. 1.14. კომპიუტერში გამოთვლითი პროცესის ორგანიზების გამარტივებული ბლოკ-სქემა

11 კომპიუტერული სისტემა დასამუშავებელ სიგნალებს ორობითი რიცხვების ფორმის მქონე მონაცემების სახით ინახავს თავის მეხსიერებაში; ამავე მეხსიერებაშია ჩაწერილი პროგრამის ცალკეული ინსტრუქციების შესაბამისი ორობითი რიცხვები (კოდები). გამოდის, რომ მეხსიერება **რიცხვების თავისებური "საწყობია";** რაც

შეეხება პროცესორს, სადაც აღნიშნულ რიცხვებზე არითმეტიკული და ლოგიკური ოპერაციები სრულდება, იგი შეიძლება **რიცხვების დამამუშავებელ "წისქვილს"** შეიძლება შევადაროთ.

რიცხვებზე არითმეტიკული და ლოგიკური ოპერაციების ჩატარების შედეგად მიიღება ახალი ორობითი რიცხვები; ეს უკანასკნელები კომპიუტერულ სისტემას შეუძლია გარდაქმნას სტერეო ბგერებად ან სამგანზომილებიან გამოსახულებებად და მომხმარებელი ზღაპრულ სამყაროში შეიყვანოს! ცხადია, რომ კომპიუტერულმა სისტემამ "დღის სინათლეზე" გამოიტანა და ყველასათვის თვალსაჩინო გახადა რიცხვების არსში დამალული პოტენციური შესაძლებლობები; ამ შესაძლებლობებზე ფიქრი პირველად ჯერ კიდევ ჩვენს წელთაღრიცხვამდე მეხუთე საუკუნეში მოღვაწე გენიალურმა ბერძენმა მოაზროვნემ **პითაგორამ** დაიწყო, რომელსაც კომპიუტერზე არც კი შეეძლო ეოცნება! მან და მისმა მოსწავლეებმა დაამუშავეს მწყობრი **მოძღვრება რიცხვების შესახებ**. მიუხედავად ამ მოძღვრების მისტიური ხასიათისა, მასში უხვადაა გაბნეული მრავალი რაციონალური მარცვალი და ამდენად იგი აქტუალობას დღემდე არ კარგავს. აღნიშნული მოძღვრების დამუშავებისა და სკოლაში ნასწავლი ცნობილი თეორემის ელეგანტური დამტკიცების გარდა, პითაგორას სახელთან ტერმინ "**ფილოსოფიის**" შემოღებაცაა დაკავშირებული.

მოწაფეების მტკიცებით კითხვაზე თუ ვინ იყო იგი, პითაგორამ უპასუხა: "მე ბრძენი (**sophos**) არა ვარ, მე ვარ სიბრძნის მოყვარე (**philosophos**) – **ფილოსოფოსი**".

მიიჩნევა, რომ პითაგორას სახელი **par excellence** (ე.ი. უმთავრესად) რიცხვის კონცეფციასთანაა დაკავშირებული. იგი ამტკიცებდა, რომ **"ბედნიერება (ევდონომია) წარმოადგენს რიცხვის სრულოფილობის ცოდნას**" და რომ **"რიცხვი მსგავსია ყოველი საგნისა**". ამ ორ მტკიცებაშია მოქცეული არსი მთელი ფილოსოფიური მიმდინარეობისა, რომელთა წარმომადგენლებს **პითაგორელები** ეწოდება.

პითაგორა
(ჩვენს წელთაღრიცხვამდე 570 – 490 წწ)

ევდონომიზმი (ბერძ. **eudaimonia** – ბედნიერება) წარმოადგენს ეთიკურ მიმართულებას, რომელიც ზნეობის ამოსავალ პრინციპად და უმაღლეს კრიტერიუმად მიიჩნევს ბედნიერებას. ზნეობრივია ისეთი ქცევა, რომელსაც ბედნიერებისაკენ მივყავართ, თვლიდნენ ანტიკური ხანის ფილოსოფოსები; ისინი ბედნიერებას სხვადასხვანაირად განსაზღვრავდნენ; პითაგორელები მიიჩნევდნენ, რომ **ბედნიერება სხვა არაფერია, თუ არა რიცხვების სრულყოფილი ცოდნა.**

პითაგორელების თანახმად ბედნიერება სამყაროს ატრიბუტია; სამყარო თავის მხრივ ჰარმონიულია და ვინაიდან ჰარმონია მხოლოდ რიცხობრივად შეიძლება გამოისახოს, ამიტომ **სამყაროც თავად რიცხვია.** ამ რიცხვის სრულყოფილ ცოდნას კი ბედნიერებისაკენ მივყავართ. ასეთ რელიგიურ-ფილოსოფიურ სწავლებას **პითაგორეიზმი** ეწოდება.

ცნობილი იურიდიული ფრაზის ინტერპრეტირებას თუ მოვახდენთ, შეგვიძლია ვთქვათ, რომ **"პითაგორეიზმი ეს არის რიცხვები, მხოლოდ რიცხვები და სხვა არაფერი რიცხვების გარდა"**. აღსანიშნავია რომ ისინი რიცხვებს ჩვენგან განსხვავებულად აღიქვამდნენ;

კერძოდ, მათთვის რიცხვები გარკვეული სახის ''მკვრივ'', ''სხეულოვან'' რეალობებს წარმოადგენს.

ანტიკურ ეპოქაში თითქმის **ყველა ფილოსოფოსი** განიცდიდა პითაგორელების, ე.ი. რიცხვების შესახებ მოძღვრების გავლენას; თანამედროვე პერიოდში რიცხვებისაკენ ლტოლვა შემცირების ნაცვლად პირიქით, ისე გაიზარდა, რომ რიცხვებით მეცნიერების გარდა საზოგადოების უფართოესი მასებიც დაინტერესდა.

12 ანტიკური ფილოსოფოსებისაგან განსხვავებით, თანამედროვე ადამიანისათვის რიცხვმა დაკარგა სხეულებრივი ფორმა და **უსხეულო** გახდა. ასეთი უსხეულო რიცხვი უკვე არა რეალური საგნების, არამედ მხოლოდ საკუთარი თავის **რეპრეზენტენტად** (ე.ი. წარმომადგენლად) განიხილება.

თანამედროვე პითაგორეიზმის დამახასიათებელ თავისებურებას წარმოადგენს ის გარემოება, რომ საგანს არა საკუთარი სურვილით ვაიგივებთ გარკვეულ რიცხვთან, არამედ საგანი ნერვულ სისტემაში თავად წარმოშობს (როგორც 1.12 **ნახაზზე** მოყვანილი **მაკ-კალოხისა** და **პიტსის** მოდელიდან ჩანს) მისთვის ''სასურველ'' რიცხვს, ანუ, ხატოვნად რომ ვთქვათ, თავად საგანია გარკვეული სახის რიცხვად საკუთარი თავის გაიგივების ''შემოქმედი''.

ნეირონულ სისტემაში ბუნებრივად ხდება საგნის ციფრირება, ე.ი. ისეთ **''იდეალურ'' რიცხვად** გარდაქმნა, რომელიც საგნის საუკეთესოდ აღქმას უზრუნველყოფს. კომპიუტერში ორგანიზებულ ხელოვნურ ციფრირებას **კოდირება** ეწოდება.

კოდირების პროცესის ორგანიზებისათვის წინასწარ უნდა დამუშავდეს კოდები. კოდების დამუშავების პროცესით დაკავებულნი არიან **კოდირების თეორიის** სპეციალისტები, ხოლო კოდების შერჩევა და კოდირების პროცესის შესრულება **დამპროგრამებლების** პრეროგატივაა. ზემოთ ხაზგასმულ ''იდეალური'' რიცხვის სინონიმი მოცემულ შემთხვევაში **''იდეალური'' კოდია**. დამპროგრამებელს კოდების შერჩევისას აქვს შეფარდებითი და არა აბსოლუტური თავისუფლება, რადგან მის მიერ შერჩეული კოდი გარკვეულ კრიტერიუმებს უნდა აკმაყოფილებდეს.

13 თანამედროვეობის უდიდეს აღმოჩენას წარმოადგენს გამოცდილებისა და ცოდნის – ინფორმაციაში, ხოლო ამ უკანასკნელის – რიცხვებში **კონვერტირების** (ლოგიკურ-სტრუქტურული ფორმის შენარჩუნებით გარდაქმნის) უნივერსალური სქემის დამუშავება (რომელიც შემოკლებული სახით ზემოთ განვიხილეთ). ბგერა, სურათი, წიგნი, გამოცდილება, სურვილი – სათანადო სქემების საშუალებით შეიძლება რიცხვებად გარდაიქმნას (ანუ, როგორც ამბობენ, მოხდეს მათი **ციფრირება,** ე.ი. ციფრებად გარდაქმნა).

აღნიშნულმა სქემამ სათავე დაუდო ახალ რეალობას, რომელსაც **ვირტუალური (წარმოსახვითი) რეალობა** ეწოდება. ვირტუალური რეალობის წარმოშობით **თანამედროვე პითაგორეიზმი** თვისობრივად ისეთ ახალ დონეზე ავიდა, რომელსაც ამ მიმართულების ფუძემდებელიც ვერ წარმოიდგენდა. **პითაგორელებისათვის** რიცხვი რჩებოდა რიცხვად, ხოლო ცხოვრება – ცხოვრებად. ახლა ყველაფერი სხვაგვარადაა: რეალური სამყაროს გვერდით არსებობა დაიწყო ვირტუალურმა სამყარომ; ამასთანავე, როგორი აბსტრაქტულიც არ უნდა იყოს ვირტუალური სამყარო, მასში არსებული მექანიზმები რეალურ სამყაროში წამოჭრილი მრავალი პრობლემის გადაწყვეტის საშუალებას იძლევა. საინფორმაციო საზოგადოებას რიცხვებთან ურთიერთობას იმპერატიული (აუცილებელი) ხასიათი აქვს. რიცხვები გამოიყენება არა მარტო მეცნიერებასა და წარმოებაში, არამედ ყოველდღიურ ჩვენს ცხოვრებაში. **რიცხვებად გადაქცეულ სამყაროში** რიცხვებს ცხადად იმიტომ ვერ ვხედავთ, რომ მათ მოახერხეს გარე სამყაროდან ჩვენს შინაგან სამყაროში გადაბარგება; ახლა ისინი ჩვენს ფსიქიკას ებრძვიან გააფთრებით და ამ ბრძოლაში გამარჯვების შემთხვევაში ალბათ ჩვენ აღვმოჩნდებით რიცხვების შიგნით, რაც შეიძლება კატასტროფული აღმოჩნდეს კაცობრიობისათვის.

საინფორმაციო საზოგადოებაში რიცხვებზე ოპერირების გარეშე რადგან არსებობა წარმოუდგენელია, ამიტომ შემდგომ რამდენიმე თავს სწორედ რიცხვებისა და მათზე არითმეტიკულ-ლოგიკური ოპერაციების შესწავლას დავუთმობთ.

II თ ა ვ ი
კომპიუტერებში ინფორმაციის წარმოდგენის საკითხები

2.1. თვლის სისტემების წარმოშობის ისტორიიდან

"აწმყო, შობილი წარსულისაგან, არის მშობელი მომავლისა"

გოტფრიდ ლაიბნიცი

1

კომპიუტერული თეორიის მცოდნე სპეციალისტები კომპიუტერების დამუშავებაზე საუბრისას ხაზს უსვამენ ამ სფეროში ლოგიკის ალგებრის, ალგორითმების თეორიის, კიბერნეტიკისა და სხვა დისციპლინების მნიშვნელობებს; ამ დროს ხშირად ივიწყებენ, რომ თანამედროვე კომპიუტერის წინაპრებად მიჩნეული სათვლელი მოწყობილობები (აბაკები, არითმომეტრები) ამ დისციპლინების გამოჩენამდე დიდი ხნით ადრე იყო დამუშავებული და რომ კომპიუტერის შექმნის იდეა სწორედ თვლასთან დაკავშირებული პრობლემების გადაწყვეტის პროცესში დაიბადა. ეს უკანასკნელი კი უშუალოდ **თვლის სისტემებთან** არის დაკავშირებული.

ზემოთ აღწერილი თავისებურება მარტო კომპიუტერული სპეციალისტებისათვის არ არის დამახასიათებელი. ფრანგი მათემატიკოსის **ანრი ლუი ლებეგის** (**1875 – 1941**) წიგნის "სიდიდეების გაზომვა" წინასიტყვობაში ცნობილი საბჭოთა აკადემიკოსი **ა.ნ.კოლმოგოროვი** (**1903 – 1987**) აღნიშნავდა: "დასრულებული მათემატიკური თეორიის დაუფლების შემდეგ მათემატიკოსებს ხშირად დასჩემდებათ ხოლმე სირცხვილით უყურებდნენ ამ თეორიის წარმოშობის სათავეებს; ძირითადი ცნებებისა და დაშვებების წყალობით კრისტალურად მოელვარე თეორიის გვერდით მათ ჭუჭყიან და უსიამოვნო სამუშაოდ ეჩვენებათ ამ ცნებებისა და დაშვებების წარმოშობის სათავეების ჩიჩქნა. ... სწორედ ასეთი ტენდენციის წინააღმდეგ ილაშქრებს **ლებეგი**".

ანალოგიური მიდგომა შეიმჩნევა ინფორმატიკოსებშიც, რომლებიც **თვლის სისტემების** განხილვას შედარებით ნაკლებ ყურადღებას უთმობენ და ყურადღებას, მათი აზრით, უფრო მნიშვნელოვანი საკითხების განხილვაზე ამახვილებენ.

მიგვაჩნია რა ასეთი მიდგომა პედაგოგიურად გაუმართლებლად, მოცემულ თავს სწორედ თვლის სისტემების წარმოშობის სათავეების განხილვით დავიწყებთ.

2

თვლის სისტემის სწორად შერჩევა მნიშვნელოვანია ინფორმატიკის ისეთი საკითხების დადებითად გადაწყვეტისათვის, როგორებიცაა არითმეტიკულ-ლოგიკურ ოპერაციათა სიმარტივისა და კომპიუტერის შეფარდებითი ეკონომიურობის მაჩვენებელთა მინიმიზაციის უზრუნველყოფა. უშუალოდ კომპიუტერულ სისტემებთან დაკავშირებული საკითხების განხილვამდე მიზანშეწონილად მიგვაჩნია ცოტაოდენი დრო დავუთმოთ თვლის სისტემების წარმოშობის ისტორიის ცალკეული ფრაგმენტების გადმოცემას, რომელთა ცოდნა სასარგებლო იქნება მკითხველის თვალსაწიერის გაფართოებისათვის.

პოზიციურ პრინციპზე დაფუძნებულ ჩვენთვის ცნობილ პირველ სისტემას წარმოადგენს ჩვენს წელთაღრიცხვამდე დაახლოებით **2000** წლის წინათ ძველი ბაბილონების

მიერ დამუშავებული **სამოცობითი** სისტემა. არსებობს ამ სისტემის წარმოშობის ორი ჰიპოთეზა.

პირველი ჰიპოთეზის თანახმად, თვლის სამოცობითი სისტემა **ასტრონომიული** წარმოშობისაა და უკავშირდება ევფრატის ვაკეზე პირველად მცხოვრებ სუმერიელთა ტომებს; ციურ სხეულებზე დაკვირვების შედეგად მათ ჩათვალეს, რომ წელიწადი შედგებოდა **60**-ის ჯერადი **360** რაოდენობის დღისაგან და ამიტომ დაამუშავეს თვლის **სამოცობითი სისტემა.**

მეორე ჰიპოთეზა ემყარება ***თითებით თვლის*** პრინციპს. ამ ჰიპოთეზის თანახმად, ევფრატის ვაკეზე ერთმანეთს შეხვდა ორი ხალხი, რომელთაგანაც ერთ-ერთი მათგანი თვლის დროს იყენებდა ორ ხელზე არსებულ, ხოლო მეორე – ერთ ხელზე არსებულ თითებს. პირველი მათგანისათვის თვლის დროს გამოყენებული ციფრების რაოდენობა იყო **ათის** ტოლი, ხოლო მეორე მათგანისათვის – **ექვსის** ტოლი. მართალია, ერთ ხელზე თითების რაოდენობა ხუთის ტოლია, მაგრამ ციფრ ექვსს მეორე შემთხვევაში შეესაბამებოდა ერთ მუშტად შეკრული თითები. ამ ორი სისტემის შერწყმით იქნა ფორმირებული თვლის სამოცობითი სისტემა.

ჩვენ ვემხრობით თვლის სისტემების ასტრონომიული წარმოშობის ჰიპოთეზას და ამიტომ ყურადღებას სწორედ ამ ჰიპოთეზაზე გავამახვილებთ.

3 ფილოსოფოსების, არქიტექტორების, მათემატიკოსების, მხატვრების ყურადღება უძველეს დროიდან მიიქციეს **სწორმა მრავალწახნაგებმა**. მათ ხიბლავდათ ამ ფიგურების სილამაზე, სრულყოფილება და ჰარმონია. ძველი ბერძნული ფილოსოფიის გაგება სილამაზისა და ჰარმონიის ცნებების გამოუყენებლად შეუძლებელია. **პლატონისათვის** სილამაზე წარმოადგენდა თავისებურ ესთეტიკურ იდეას, რომლის შემეცნება განსაკუთრებულად შთაგონებულ მდგომარეობაში ყოფნის დროს შეიძლებოდა; **არისტოტელეს** მიხედვით სილამაზე საგნის მათემატიკურ პროპორციებში იყო განივთებული და მისი წვდომის შესაძლებლობას მათემატიკური საქმიანობა იძლეოდა; **პითაგორა** სილამაზის ცნებას არა მარტო ზოგადი სამყაროს სურათს უკავშირებდა, არამედ მას მორალურ-რელიგიურ შინაარსსაც აძლევდა; სილამაზის გაგების პრობლემის გადაწყვეტას დიდ დროს უთმობდა **დემოკრიტეც** და ა.შ.

4 **სწორი მრავალწახნაგი** ამოზნექილი გეომეტრიული ფიგურაა, რომლის ყველა წახნაგი ერთნაირი სწორი **მრავალკუთხედია**, ხოლო წახნაგებთან არსებული კუთხეები – ერთმანეთის ტოლია.

ტეტრაედრი **გეკსაედრი (კუბი)** **ოქტაედრი** **დოდეკაედრი** **იკოსაედრი**

ნახ. 2.1 პლატონური სხეულები

არსებობს უამრავი მრავალკუთხედი, მაგრამ მათ შორის მხოლოდ **ხუთი წარმოადგენს** სწორ მრავალკუთხედს. მართალია ისინი უძველესი დროიდანაა ცნობილი, მაგრამ მათ დღეს ძველ საბერძნეთში შერქმეული სახელებით მოვიხსენიებთ. თითოეული ფიგურის სა-

ხელწოდება გამოხატავს თუ რამდენ წახნაგს (რომელიც ბერძნულად "**ედრას**" ნიშნავს) შეიცავს იგი; ამ დროს გამოყენებულია რიცხვები:

ტეტრა, რაც ნიშნავს **4**-ს;
გექსა, რაც ნიშნავს **6**-ს
ოქტა, რაც ნიშნავს **8**-ს;
დოდეკა, რაც ნიშნავს **12**-ს;
იკოსა, რაც ნიშნავს **20**-ს.

ზემოთ აღნიშნულის საფუძველზე მრავალწახნგების სახელწოდებებია: **ტეტრაედრი** (ე.ი. ოთხწახნაგა), **გექსაედრი** (ე.ი. ექვსწახნაგა; მას **კუბსაც** უწოდებენ), **ოქტაედრი** (ე.ი. რვაწახნაგა), **დოდოკაედრი** (ე.ი. თორმეტწახნაგა) და **იკოსაედრი** (ე.ი. ოცწახნაგა).

ცნობილმა ბერძენმა ფილოსოფოსმა **პლატონმა (ძვ.წ.აღ. 428 ან 427 – 348 ან 347 წწ)** სწორი მრავალწახნაგები მისტიკური შინაარსით აღჭურავა; მისი აზრით **ტეტრაედრი** განასახიერებდა ცეცხლს, ვინაიდან მისი წვეროები ცეცხლის ალივით ზემოთკენ იყვნენ მიმართულნი; ყველაზე გარშემოდინებადი (მოკვერცხილი) ფორმის **იკოსაედრი –** წყალს, ყველაზე მდგრადი კონსტრუქციის მქონე **გექსაედრი** (**კუბი**) – მიწას, ხოლო **ოქტაედრი** – ჰაერს განასახიერებდა; რაც შეეხება **დოდოკაედრს,** მას იგი მთლიანად სამყაროს განმასახიერებელ ყველაზე მთავარ მრავალწახნაგად თვლიდა. შემდგომში ზემოთ ჩამოთვლილ სწორ მრავალწახნაგებს პლატონის საპატივცემლოდ **პლატონური სხეულები** ეწოდა (**ნახ.2.1**). მათი ძირითადი მონაცემები **2.1** ცხრილშია მოყვანილი.

ცხრ. 2.1 პლატონური სხეულების ძირითადი მონაცემები

მრავალწახნაგის სახელწოდება	**წვეროების რაოდენობა**	**წიბოების რაოდენობა**	**წახნაგების რაოდენობა და მათი სახელწოდებები**
ტეტრაედრი	4	6	4 სწორი სამკუთხედი
გექსაედრი (კუბი)	8	12	6 კვადრატი
ოქტაედრი	6	12	8 სწორი სამკუთხედი
დოდოკაედრი	20	30	12 სწორი ხუთკუთხედი (პენტოგრამა)
იკოსაედრი	12	30	20 სწორი სამკუთხედი

5 ბაბილონსა და ძველ ეგვიპტეში **კალენდრების შედგენისას** დიდ მნიშვნელობას ანიჭებდნენ გიგანტი პლანეტებიდან ყველაზე დიდ პლანეტა **იუპიტერს,** რომელიც მზის გარშემო შემოვლას დაახლოებით **12** წელს ანდომებს. პლანეტის მიერ მზის გარშემო შემოვლისათვის საჭირო წლების რაოდენობას ამ **პლანეტის ციკლი** ეწოდება; ე.ი. **იუპიტერის ციკლი 12** წლის ტოლია. არანაკლებ როლს თამაშობდა **სატურნიც**, რომლის **ციკლიც 30** წელს უდრის. საკრალურ პლანეტებად იუპიტერისა და სატურნის მიჩნევის გამო მათი ციკლების (**12-ისა** და **30-ის**) უმცირესი საერთო ჯერადი რიცხვი **60** (**60=5x12=2x30)** ძველი კალენდრის შემდგენლებმა მიიჩნიეს **მზის სისტემის ძირითად ციკლად 60.**

რიცხვ **60**-ს განსაკუთრებულ რიცხვად მიიჩნევდნენ კიდევ იმიტომ, რომ იგი დაკავშირებულია უძველესი დროში სამყაროს ჰარმონიის სიმბოლოებად მიჩნეულ განსაკუთრებული მნიშვნელობის მქონე პლატონურ სხეულ **დოდოკაედრთან** (იხ. ნახ. **2.1**). ამ სხეულის წახნაგებია სწორი **ხუთკუთხედები,** ანუ **პენტოგრამები,** ხოლო მის ზედაპირზე

არსებული კუთხეების რაოდენობა **5 x 12 = 60**-ის ტოლია, რაც შეესაბამება მზის სისტემის ზემოთ აღნიშნულ **60**-წლიან ციკლს. დოდოკაედრს აქვს **30** წიბო (**სატურნის ციკლი**) და **12** წახნაგი (**იუპიტერის ციკლი**). ამ რიცხვების ნამრავლით მიიღება რიცხვი **30x12=360.** სწამდათ რა **დოდოკაედრის** მაგიური რიცხვითი სიმბოლიკის, ძველმა ბაბილონელებმა თვლის საკუთარი სისტემის ფუძედ აირჩიეს რიცხვი **60**; ანალოგური მიზეზით ძველმა ეგვიპტელებმა წელიწადი დაყვეს **12** თვედ (დოდოკაედრის წახნაგების რაოდენობა), ხოლო თითოეულ თვეში გააერთიანეს **30** დღე (დოდოკაედრის წიბოების რაოდენობა); თორმეტი თვის ოცდაათზე (თვეში დღეების რაოდენობა) ნამრავლს დაუმატეს **5** დღე (**პენტოგრამის** კუთხეების რაოდენობა), რომლებიც გამოაცხადეს სადღესასწაულო დღეებად და არ შეიყვანეს არცერთ თვეში. ამის შედეგად მიიღეს, რომ თითოეული წელი შედგებოდა **12x30+5=365** დღისაგან. აქვე შევნიშნავთ, რომ დროის აღრიცხვისა და კუთხური სიდიდეების გასაზომადაც ეგვიპტელებმა დოდოკაედრის "მაგიური" რიცხვები გამოიყენეს.

6 რიცხვების აღნიშვნის პოზიციური სისტემა მატერიალური კულტურის ისტორიის ერთ-ერთ ძირითად მიღწევად ითვლება; მის შექმნაში მრავალი ხალხი იღებდა მონაწილეობას. ახალი ერის **მე-6** საუკუნეში მსგავსი სისტემა მაიას ტომებშიც წარმოიშვა. გავრცელებულია შეხედულება, რომ მაიას ტომების მიერ დამუშავებული თვლის სისტემის ფუძედ გამოყენებული იყო რიცხვი **20,** რაც ადამიანის თითების რაოდენობის გამომხატველი რიცხვია.

მაიას ტომები წელიწადს ჰყოფდენ **20** დღის შემცველ **18** თვედ; წელიწადში არსებული დღეების რაოდენობის გამოსათვლელად კი **20**-ისა და **18**-ის ნამრავლს უმატებდნენ რიცხვ **5**-ს და ბაბილონელების მსგავსად **365** დღეს იღებდნენ.

ითვალისწინებენ რა მაიას ტომის მაღალ კულტურულ დონეს, მკვლევრები თვლიან, რომ მისი წარმომადგენლები იცნობდნენ **"პლატონურ სხეულებს"** (იხ. ნახ. **2.1**) და მათ წლიური კალენდრის ასაგებად **დოდოკაედრის** ნაცვლად აირჩიეს **იკოსაედრი**. ამ უკანასკნელს აქვს **20** წახნაგი, **30** წიბო (როგორც დოდოკაედრს) და **12** წვერო; თითოეულ წვეროში იკრიბება **5** კუთხე, ე.ი. **იკოსაედრის** კუთხეების რაოდენობა **5x12=60**-ის ტოლია. ამგვარად, **იკოსაედრის** რიცხვითი მაჩვენებლებიც მზის სისტემის **12-, 30-** და **60**-წლიან ციკლებთანაა დაკავშირებული.

7 ჩვენ ყოველდღიურ გამოთვლებში ვიყენებთ თვლის ათობით სისტემას, რომლის წინაპრად ითვლება ჩვენი წელთაღრიცხვის დაახლოებით **XII** საუკუნეში ჩამოყალიბებული ინდუსური ათობითი სისტემა. ცნობილი ფრანგი მათემატიკოსი **პიერ სიმონ ლაპლასი** პოზიციური პრინციპის გამოყენებით თვლის სისტემების აგების გულმხურვალე მომხრე იყო და აღფრთოვანებას ვერ მალავდა თვლის ათობითი სისტემის მიმართ. კერძოდ, იგი აღნიშნავდა, რომ "ცხრა ნიშნის საშუალებით ნებისმიერი რიცხვის გამოსახვის შესახებ აზრი, რომლის დროსაც თითოეული ნიშნის (ციფრის) საერთო მნიშვნელობა მისი საკუთარი მნიშვნელობის გარდა რიცხვში მის მიერ დაკავებულ ადგილზეა (პოზიციაზეა) დამოკიდებული, იმდენად მარტივია, რომ ეს **სიმარტივე აღნიშნული აზრის საოცრების აღქმას აძნელებს;** ამ მეთოდის გათავისების სირთულე კარგად ჩანს ბერძნული მეცნიერების უდიდესი გენიოსების **არქიმედეს** (ჩვენს წელთაღრიცხვამდე **287-212** წწ.) **აპოლონის** (იგულისხმება **აპოლონ ტიანელი**, რომელმაც სასწაულმოქმედის სახელი მოიხვეჭა; დაიბადა **I** საუკუნეში, გარდაიცვალა **98** ან **100** წელს) მაგალითზე, რომლებისათვისაც აღნიშნული აზრი დაფარული აღმოჩნდა".

სავაჭრო პრაქტიკაში ინდო-არაბული ათობითი სისტემის გამოყენების თავდაჯერებული მომხრე იყო ცნობილი იტალიელი მათემატიკოსი **ლეონარდო პიზანელი** (**ფიბონაჩი, ≈1170–1250**), რომელმაც მათემატიკური განათლება არაბულ ქვეყნებში მიიღო. იგი წერდა: "ინდუსური ცხრა ნიშნის **9; 8; 7; 6; 5; 4; 3; 2; 1** და **zephiruum**-ის სახელწოდების მქონე ნიშან **0**-ის დახმარებით ნებისმიერი რიცხვის დაწერაა შესაძლე-

ბელი''. **ფიბონაჩმა** აქ სიტყვით "**zephiruum**" გადმოსცა არაბული სიტყვა "**as-sifr**", რაც ნიშნავს ''ცარიელს''; ამ უკანასკნელიდან იქნა მიღებული ყველასათვის ცნობილი ტერმინი ''ციფრი''.

ლეონარდო პიზანელი
(ფიბონაჩი ≈1170-1250)

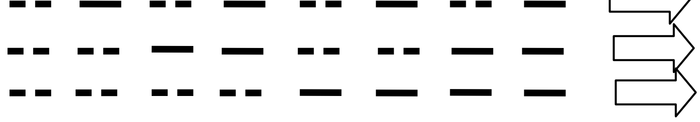
პიერ სიმონ ლაპლასი
(1749-1827)

ზემოთ ჩვენ საუბარი გვქონდა თვლის სამოცობით, ოცობით და ათობით სისტემებზე, რომლებისთვისაც ფუძეებად გამოყენებული იყო შესაბამისად რიცხვები **60; 20** და **10**. შემდეგში აღმოჩნდა, რომ თვლის სისტემის ფუძედ შეიძლება გამოყენებული ყოფილიყო ნებისმიერი (როგორც ნატურალური, ასევე ნამდვილი) რიცხვი და ამ გზით შექმნილიყო თვლის უსასრულო რაოდენობის სისტემები. ამ სისტემებს შორის კომპიუტერული ტექნიკის განვითარების შედეგად პირველ ადგილზე გავიდა **თვლის ორობითი სისტემა,** რომელშიც ნებისმიერი რიცხვი შეიძლება ციფრების **0**-ისა და **1**-ის გარკვეული მიმდევრობის სახით გამოისახოს.

თვლის ორობითი სისტემის ფორმის ჩანასახები მსოფლიოს მრავალი ხალხის კულტურულ მემკვიდრეობაშია აღმოჩენილი. შევჩერდებით მხოლოდ შემდეგ ორ მაგალითზე:

- ძველ ეგვიპტეში თავის დროზე ფართოდ იყო გავრცელებული გაორმაგების პრინციპზე დაფუძნებული გამრავლებისა და გაყოფის მეთოდები;
- ჩინეთში მომუშავე იეზუიტმა მისიონერმა **ბუვემ** (**Bouvet**) ცნობილ გერმანელ ფილოსოფოსს **გოტფრიდ ლაიბნიცს** მიწერა, რომ იქ არსებობს გამოუცნობი წარწერა, რომლის ავტორად ითვლება ჩვენს წელთაღრიცხვამდე **25**-ე მცხოვრები ჩინელი იმპერატორი, ჩინეთის იმპერიის დამფუძნებელი, მეცნიერებისა და ხელოვნების დიდ მფარველი **ფო გი**; აღნიშნული წარწერა შედგება გრძელი და მოკლე ხაზების მწკრივებისაგან (**ნახ. 2.2,ა**); თვლის ორობით სისტემას თუ გამოვიყენებთ და გრძელ ხაზებს შევცვლით ციფრ **1**-ით, ხოლო მოკლე ხაზებს – ციფრ **0**-ით, მაშინ მივიღებთ ორობით სისტემაში ჩაწერილ ნატურალური რიცხვებს (**ნახ. 2.2,ბ**).

ა) **ბ)**

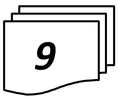


0 0 1 0 0 1 0 0 1 0 0 1
0 0 0 0 1 1 0 0 0 0 1 1
0 0 0 0 0 0 0 0 1 0 1 1

ნახ. 2.2 ჩინური გამოუცნობი წარწერა და მისი ამოცნობისათვის ორობითი რიცხვების გამოყენება

ორობითი რიცხვების შექმნაზე მრავალი მეცნიერი მუშაობდა. ამ მათემატიკოსებს შორის იყო **ფიბონაჩიც**, რომელმაც თავის წიგნში "**Liber abaci** " ჩამოაყალიბა "ბერკე-ტულ სასწორზე ტვირთების ასაწონად წონითი გირების საუკეთესო სისტემის შერჩევის ამოცანა". პრობლემის სრულყოფილად გადაწყვეტის პატივი ერგო **გოტფრიდ ლაიბნიცს,** რომელმაც 1697 წელს დაამუშავა **ორობითი არითმეტიკის წესები**. **ლაიბნიცი** ისე იყო აღფრთოვანებული მიღწეული შედეგით, რომ ამ აღმოჩენის პატივსაცემად სპეციალუ-რი მედალიც კი გამოუშვა! ეს მათემატიკის ისტორიაში იყო უიშვიათესი შემთხვევა, რო-დესაც მედლით დაჯილდოების ღირსი გახდა მათემატიკური აღმოჩენა.

ზემოთ აღნიშნულის მიუხედავად **ლაიბნიცი** არ იყო ორობითი სისტემით ათობითი სისტემის შეცვლის მომხრე. იგი წინასწარმეტყველურად ვარაუდობდა, რომ ორობითი სი-სტემის გამოყენებით გამოთვლების პროცესის კვლევა მომავალში აუცილებლად მიგვი-ყვანდა ახალ აღმოჩენებამდე; ამის განმაპირობებელ მიზეზად იგი მიიჩნევდა იმ რეალო-ბას, რომ უმარტივეს საწყისებზე რიცხვების დაყვანით ყველგან საოცარი წესრიგი დამყა-რდებოდა.



ლაიბნიცის წინასწარმეტყველება ახდა 1946 წელს ამერიკელი მეცნიერე-ბის **არტურ ბიორკერის**, **ჰერმან გოლდსტაინისა** და **ჯონ ფონ ნეიმანის** ჩვენ მიერ პირველ თავში აღნიშნული სტატიის გამოქვეყნების შემდეგ. ამ სტატიაში ჩამოყალიბებული პრინციპებიდან (რომელიც **ნეიმანის პრინციპების** სახელ-წოდებითაა ცნობილი) პირველი პრინციპი მოითხოვდა, რომ ელექტრონულ კომპიუტერებ-ში ინფორმაციის **კოდირების უნივერსალურ ხერხად** მიჩნეულიყო ორობითი რიცხვების სახით წარმოდგენილი კოდური სიტყვებით კოდირების ხერხი, ე.ი. **კომპიუტერული გამო-თვლებისათვის გამოყენებული ყოფილიყო თვლის ორობითი სისტემა!** ასე, რომ თუ საყო-ფაცხოვრებო გამოთვლებისათვის საკმარისია სკოლაში ნასწავლი თვლის ათობითი სის-ტემისა და ათობითი არითმეტიკის ცოდნა, კომპიუტერის შიგნით მიმდინარე გამოთვლებ-ში გასარკვევად დამატებით **თვლის ორობითი სისტემისა და ორობითი არითმეტიკის** შესწავლა დაგვჭირდება.

ამგვარად **ბაბილონელების** მიერ თვლის პოზიციური პრინციპის აღმოჩენა, შემდეგ **ინდუსების** მიერ თვლის ათობითი სისტემის ჩამოყალიბება და, ბოლოს, **ლაიბნიცის** მიერ ორობითი არითმეტიკის დამუშავება შეიძლება მივაკუთვნოთ ეპოქალური მნიშვნელობის მათემატიკურ აღმოჩენებს, რომლებმაც არსებითი გავლენა მოახდინა ზოგადად მატერი-ალური კულტურის განვითარებაზე და კერძოდ, კომპიუტერული ტექნიკის დამუშავებაზე!

ახალ რეალობაში ძველი მეთოდის გამოყენებისას ცალკეული ხარვეზე-ბის სახით თავს აუცილებლად იჩენს აღნიშნული მეთოდისათვის დამა-ხასიათებელი შინაგანი ფარული შეზღუდულობა და ჩნდება მისი აღმო-ფხვრის გზით ძველი მეთოდის "გათანადროულობის" მოთხოვნა. ასე მოხდა მაშინაც, რო-დესაც XX საუკუნის შუა პერიოდში დასამუშავებელ კომპიუტერულ სისტემებში გამოსა-ყენებლად შეირჩა **გოტფრიდ ლაიბნიცის** მიერ XVII საუკუნეში დამუშავებული კლასი-კური ორობითი არითმეტიკა. აღნიშნული არითმეტიკისათვის დამახასიათებელია მთელი რიგი შეზღუდვები, რომელთა შორისაც უმთავრესია უარყოფითი რიცხვების წარმოდგე-ნის პრობლემა და ორობითი რიცხვების ე.წ. "**ნულოვანი სიჭარბე**".

განსაკუთრებით არასასურველია მეორე შეზღუდულობა. რიცხვების ორობითი წარმო-დგენის "ნულოვანი" სიჭარბე ნიშნავს, რომ თვლის ასეთ სისტემებში არ არსებობს იმ შეცდომების აღმოჩენის მექანიზმი, რომლებიც გარეგანი და შინაგანი გავლენების გამო აუცილებლად წარმოიშვება კომპიუტერულ სისტემებში. იმ პირობებში, როდესაც კაცო-ბრიობა სულ უფრო და უფრო ხდება კომპიუტერული რევოლუციის მძევალი და რაკეტე-ბის, თვითმფრინავების, ატომური რეაქტორების მართვის ურთულესი ამოცანების გად-აწყვეტის დროს იგი სწორედ კომპიუტერებზე ამყარებს მთელ იმედებს, ცხადია შეცდომე-ბის აღმოჩენის ეფექტური მექანიზმით კომპიუტერში გამოყენებული თვლის სისტემის აღჭურვის პრობლემა უაქტიურეს პრობლემათა რიგს მიეკუთვნება.

სამობითი კომპიუტერი "**Сетунь**"

ნ.პ. ბრუსენცოვი (1925)

ორობითი სისტემებისათვის დამახასიათებელი ზემოთ აღნიშნული ხარვეზების აღმოსაფხვრელად კომპიუტერული ერის დაწყებამდეც მოხდა თვლის სისტემასთან დაკავშირებული რამდენიმე მათემატიკური აღმოჩენა. ერთ-ერთი მათგანია პოლონელი ლოგიკოსის **იან ლუკასევიჩის** (1878 – 1956) მიერ 1920 წელს დამუშავებული **სამნიშნა ლოგიკა,** რომელიც წარმოადგენს თვლის ორობით სისტემაში გამოყენებული ორობითი ლოგიკის მარტივ გაგრძელებას. **ორობითი ლოგიკის** შემთხვევაში **x** ცვლადი იღებს ორ, კერძოდ, ჭეშმარიტსა და ყალბ მნიშვნელობას; პირველს თუ აღვნიშნავთ ციფრ **1**-ით, ხოლო მეორეს – ციფრ **0**-ით, გვექნება: $\mathbf{x}\in(1;\ 0)$; ე.ი. მივიღებთ ორი განსხვავებული ნიშნისაგან შედგენილ, ანუ ორობით რიცხვებს. **სამობითი ლოგიკის** დროს **x** ცვლადი დამატებით იღებს მესამე, კერძოდ, განუსაზღვრელ მნიშვნელობას; მას თუ აღვნიშნავთ როგორც $\bar{1}$-ს, გვექნება: $\mathbf{x}\in(1;\ \bar{1};\ 0)$; ე.ი. ვიღებთ სამი განსხვავებული ნიშნისაგან შედგენილ, ანუ **სამობით რიცხვებს.**

1959 წელს მოსკოვის უნივერსიტეტში **ნ.პ. ბრუსენცოვის** ხელმძღვანელობით დამუშავდა კომპიუტერი "**Сетунь**", რომელშიც გამოყენებულმა **თვლის სამობითმა სასტემამ**, კომპიუტერების ისტორიაში პირველად დასვა ტოლობის ნიშანი უარყოფით და დადებით რიცხვებს შორის. ამან საშუალება მოგვცა თავი აგვერიდებინა უარყოფითი რიცხვების წარმოსადგენად სხვადასხვა "ხრიკების" (შებრუნებული და დამატებითი კოდების) მოფიქრებისათვის. აღნიშნულმა გარემოებამ, აგრეთვე პროგრამების დამუშავებისათვის სამობითი ლოგიკის გამოყენებამ შესაძლებელი გახადა შექმნილიყო ძალიან სრულყოფილი სტრუქტურა, რომლის რეალიზაცია მოხდა **Сетунь**-ის არქიტექტურაში. **Сетунь** წარმოადგენს იმის ნათელ მაგალითს, თუ **როგორ მნიშვნელოვან გავლენას ახდენს თვლის სისტემა კომპიუტერის არქიტექტურაზე.** ამერიკელი მეცნიერი, სტენფორდის უნივერსიტეტის პროფესორი, საქართველოში საყოველთაოდ ცნობილი მონოგრაფია-ბესტსელერის **"დაპროგრამების ხელოვნების"** (რუსულ ენაზე) ავტორი **დონალდ ერვინ კნუტი** აღნიშნავს, რომ კომპიუტერებისათვის ორობითი კომპონენტების მასობრივად წარმოების გამოთვლითი ტექნიკის ისტორიაში ჯერ კიდევ უმნიშვნელო ადგილი უკავია სამობით კომპიუტერებს; მაგრამ **სამობითი ლოგიკა ორობით ლოგიკაზე ელეგანტური და ეფექტურია** და მომავალში კაცობრიობა ალბათ ისევ დაუბრუნდება სამობითი კომპიუტერების დამუშავებას.

9 დასასრულს, რამდენიმე სიტყვით შევეხებით თვლის ორობითი სისტემისათვის დამახასიათებელ "**ნულოვანი სიჭარბისგან**" თავის აცილების შესაძლებლობას. ჩვენ მიერ ზემოთ განხილული თვლის პოზიციური სისტემების **b** ფუძედ გამოყენებული იყო რიცხვები **60;20;10;2;3**; ე.ი. განხილული სისტე-

მებისათვის **b**$\in$**{60; 20; 10; 3}** და მათ შესაბამისად ეწოდება **სამოცობითი, ოცობითი, ათობითი, ორობითი და სამობითი სისტემები**.

თვლის კონკრეტული პოზიციური სისტემით გამოსახული რიცხვის თანრიგების "წონების" განმსაზღვრელი რიცხვების მიმდევრობას მოცემული თვლის სისტემის **ბაზისი** ეწოდება. ზემოთ ჩამოთვლილ სისტემებში ბაზისის გამომსახველი ნებისმიერ რიცხვთა მიმდევრობები **n** = **0; 1; 2; 3; ...** ხარისხში **q** ფუძის აყვანის გზით მიიღება; კერძოდ, სამოცობით სისტემაში გამოყენებული ბაზისი **q=60** რიცხვის **n** = **0; 1; 2; 3; ...** ხარისხში აყვანით მიიღება:

$$60^n; \ldots, 60^3;\ 60^2;\ 60^1,\ 60^0.$$

სისტემებს, რომლებშიც ფუძის მაჩვენებელი **b** რიცხვის ახარისხების გზით მიღებული ბაზისი გამოიყენებს **b-სისტემები** ვუწოდოთ.

ამერიკელმა მათემატიკოსმა **ჯორჯ ბერგმანმა** 1957 წელს გამოქვეყნებულ სტატიაში **"A Number System With Irrational Base"** შემოგვთავაზა, რომ თვლის პოზიციური სისტემის ბაზისად გამოგვეყენებინა "**ოქროს კვეთის**" **გამომხატველი** $\tau = \frac{1+\sqrt{5}}{2}$ **რიცხვის n** = **... ; ±n; ... ; ±3; ±2; ±1; 0; ...** ხარისხში აყვანით მიღებული რიცხვების მიმდევრობა:

$$\ldots,\ \tau^n;\ \ldots, \tau^3;\ \tau^2;\ \tau^1,\ \tau^0;\ \tau^{-1},\ \tau^{-2};\ \tau^{-3};$$

ჯორჯ ბერგმანი (1945)

საინტერესოა ის გარემოება, რომ τ რიცხვს თუ ავიყვანთ რაიმე **n**-ურ ხარისხში, მაშინ τ^n რიცხვი შეიძლება წინა ორი (კერძოდ, **n-1** და **n-2**) ხარისხის ჯამის სახით გამოისახოს, ე.ი. სამართლიანია ტოლობა:

$$\tau^n = \tau^{n-1} + \tau^{n-2},$$

სადაც **n** მნიშვნელობებს იღებს მთელი რიცხვების

{0; ±1; ±2; ±3, ...}

სიმრავლიდან.

თვლის ორობითი სისტემის თანრიგების წონებად τ^n (**n**=0; ±1; ±2; ±3, ...) რიცხვთა მიმდევრობების გამოყენებისას მიიღება ირაციონალური $\tau = \frac{1+\sqrt{5}}{2}$ რიცხვით ნაწარმოები ბაზისის მქონე თვლის "ორობითი სისტემა"; აღნიშნულ სისტემას, ზემოთ ნახსენები **q-სი-სტემისაგან** განსხვავებით, τ**-სისტემა** ეწოდება; ავტორის პატისაცემად მას ხშირად **ბერგმანის სისტემადაც** მოიხსენებენ. იგი ასე გამოისახება:

$$A = \sum a_i \tau^i,$$

სადაც **A** არის ნებისმიერი ნამდვილი რიცხვი; a_i – ორობითი ციფრები ($a_i \in \{0;1\}$) ,

$i = 0; \pm 1; \pm 2; \pm 3, \ldots$; τ^i – თვლის სისტემაში i –ური ციფრის წონა, ხოლო τ^i – თანრიგის "წონას" გამოსახავს.

გარეგნულად "ბერგმანის სისტემა" თითქოს არაფრით განსხვავდება თვლის კლასიკური ორობითი სისტემისაგან, მაგრამ ეს მხოლოდ მირაჟული მსგავსებაა! საქმე ისაა, რომ გარეგნულად ორობითი სისტემის მსგავს **ბერგმანის სისტემაში** გამოყენებულია არა 2-ის, არამედ ოქროს პროპორციის გამომსახველი $\tau = \frac{1+\sqrt{5}}{2}$ რიცხვის ხარისხები, რაც მიღებულ სისტემას შემდეგ მათემატიკურ თვისებას ანიჭებს: **კლასიკური ორობითი სისტემა** თუ "**არაჭარბ სისტემას**" წარმოადგენს, **ბერგმანის სისტემა "ჭარბი სისტემაა"**; სისტემის სიჭარბეს მეტად დიდი პრაქტიკული მნიშვნელობა აქვს; კერძოდ, ასეთი სისტემის გამოყენების შემთხვევაში შეგვეძლება:

- გავაკონტროლოთ არითმეტიკული ოპერაციების სწორად შესრულება ე.ი. ავაგოთ მტყუნებამდგრადი პროცესორი;
- ანალოგურ-ციფრულ გარდამქმნელებში მოვახდინოთ შეცდომების კორექცია ე.ი. მივანიჭოთ მას თვითკონტროლის უნარი;
- კავშირგაბმულობის ასაგებად მოვახდინოთ კოდური მიმდევრობების თვითსინქრონიზაცია და ა.შ.

აღსანიშნავია, რომ თვლის ასეთი საინტერესო სისტემა **ჯორჯ ბერგმანმა** შექმნა და ამერიკის მეტად პრესტიჟულ ჟურნალ "**Mathematics Magazine**" –ში გამოაქვეყნა ... 12 წლის ასაკში! კალიფორნიის უნივერსიტეტში მათემატიკოსად მომუშავე ჯორჯ ბერგმანი მსოფლიოში სწორედ ბავშვობის ასაკში დამუშავებული თვლის აღნიშნული ორიგინალური სისტემითაა ცნობილი!

2.2. თვლის სისტემების ზოგადი მიმოხილვა

"მათემატიკაში შემოქმედების მთელი ხელოვნება სიმბოლოდან მომდინარეობს და რაც უფრო წარმატებულია სიმბოლო, მით უფრო ძლიერია იგი"

ლაიბნიცი – მარკიზ ლოპიტალს

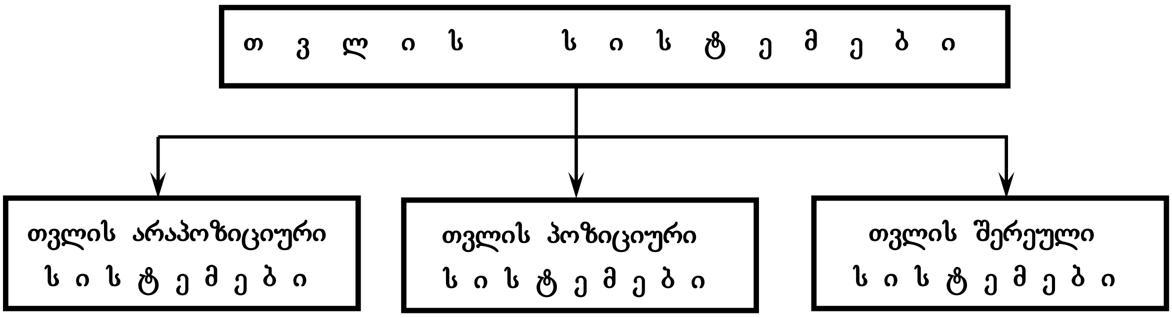
დიდი ხნის განმავლობაში **რიცხვი** მხოლოდ ობიექტების რაოდენობრივი დახასიათებისა და მათი დანომვრისათვის გამოყენებულ აბსტრაქციად მიიჩნეოდა, მაგრამ ადამიანის მოღვაწეობის სფეროს გაფართოების კვალობაზე მისი შინაარსი განუწყვეტლივ მდიდრდებოდა და მრავალ სხვა ფუნქციასაც იძენდა. ამის შედეგად დღეისათვის **რიცხვის**, **რიცხვული ინფორმაციისა** და **ციფრული მონაცემების** ცნებები ინფორმატიკის **ფუნდამენტურ ცნებებად** გადაიქცა.

რიცხვის ცნებასთან უშუალოდაა დაკავშირებული მეორე მნიშვნელოვანი ცნება – **თვლის სისტემა**, რომლის ქვეშაც იგულისხმება ციფრული ნიშნებით ან სიმბოლოებით რიცხვების ჩაწერის ხერხებისა და წესების ერთობლიობა. თვლის სისტემებმა უნდა უზრუნველყოს:

- მნიშვნელობათა განსახილველ დიაპაზონში რიცხვის ნებისმიერი მნიშვნელობის წარმოდგენის შესაძლებლობა;
- წარმოდგენის ერთადერთობა (სიმბოლოების თითოეული კომბინაცია მხოლოდ ერთ მნიშვნელობას ან რაოდენობას უნდა შეესაბამებოდეს);
- რიცხვებზე ოპერაციების ჩატარების სიმარტივე.

თვლის სისტემა წარმოადგენს რიცხვების დასახელება-აღნიშვნისათვის აუცილებელი ხერხებისა და წესების ერთობლიობას, რომელიც საშუალებას გვაძლევს ურთიერთცალსახა შესაბამისობა დავამყაროთ ნებისმიერ რიცხვსა და სასრული რაოდენობის სიმბოლოებისაგან შემდგარ მის გამოსახულებას შორის.

თვლის ნებისმიერ სისტემაში, უპირველეს ყოვლისა, შეირჩევა გარკვეული სიმბოლოების (სიტყვების ან ნიშნების) ერთობლიობა, რომელსაც **ალფაბეტი** ეწოდება. ეს უკანასკნელი გვეხმარება დადგენილი ოპერაციების ჩატარების გზით გამოვსახოთ ნებისმიერი რაოდენობა. ნებისმიერი რაოდენობის გამოსახულებას **რიცხვი,** ხოლო ალფაბეტის სიმბოლოებს – **ციფრები** ეწოდება. ალფაბეტში შემავალი სიმბოლოები ერთმანეთისაგან უნდა განსხვავდებოდეს და წინასწარ უნდა იყოს ცნობილი თითოეული მათგანის მნიშვნელობა.



ნახ. 2.3 თვლის სისტემების კლასიფიკაცია ციფრების მნიშვნელობის განსაზღვრის ნიშნის მიხედვით

რიცხვებში შემავალი ციფრების მნიშვნელობის განსაზღვრის წესზე დამოკიდებულებით ერთმანეთისაგან განასხვავებენ თვლის არაპოზიციურ, პოზიციურ და შერეულ სისტემებს.

თვლის არაპოზიციურ სისტემაში გარკვეული რაოდენობის აღმნიშვნელი სიმბოლოს (ციფრის) მნიშვნელობა რიცხვში მის მიერ დაკავებულ ადგილზე (პოზიციაზე) არ არის დამოკიდებული. აქვე უნდა აღინიშნოს, რომ შეიძლება არსებობდეს რიცხვში სიმბოლოების (ციფრების) განლაგების გარკვეული წესი.

თვლის არაპოზიციური სისტემებია:

- თვლის ბინომიალური სისტემა; მასში რიცხვების წარმოსადგენად ბინომიალური კოეფიციენტებია გამოყენებული;
- თვლის რომაული სისტემა;
- თვლის ბერძნული სისტემა;
- ნაშთების კლასთა სისტემა. მასში რიცხვების წარმოდგენისათვის გამოიყენება **დაქვითვის ცნება** და ჩინური თეორემა ნაშთების შესახებ;
- თვლის **შტერნ-ბროკოს სისტემა;** ეს არის დადებითი რაციონალური რიცხვების ჩაწერის ხერხი, რომელიც შტერნ-ბროკის ხის გამოყენებაზეა დაფუძნებული;

ზემოთ ჩამოთვლილი სისტემებიდან მოკლედ გავიხსენებთ სასკოლო ალგებრის კურსიდან ცნობილ **თვლის რომაულ არაპოზიციურ სისტემას.** აღნიშნული სისტემა დღეს გამოიყენება წიგნის თავების, თხზულებათა ნაკრებების ტომების, საკუნეების დასანომრად. აღნიშნულ სისტემაში ციფრებად ლათინური მთავრული ასოებია გამოყენებული. ცხრილ 2.2-ში მოყვანილია **რომაული ციფრები** და მათი მნიშვნელობები.

ცხრილი 2.2. რომაული ციფრები და მათი მნიშვნელობები

რომაული ციფრები	I	V	X	L	C	D	M
მნიშვნელობები	1	5	10	50	100	500	1000

თვლის მოცემულ სისტემაში რიცხვები შემდეგი წესების მიხედვით იწერება:

- მარცხნივ მდგარი ციფრი თუ მარჯვნივ მდგარ ციფრზე ნაკლებია, მაშინ მარჯვენა რიცხვს უნდა გამოვაკლოთ მარცხენა რიცხვი (**IV: 1 < 5,** მაშასადამე **5–1 = 4, XL: 10 < 50,** მაშასადამე, **50-10-40**);
- მარჯვენივ მდგარი ციფრი თუ ნაკლებია ან ტოლია მარცხნივ მდგარ ციფრზე, მაშინ ეს ციფრები იკრიბება **(VI : 5 + 1 = 6; VIII: = 5+1+1+1 =8; XX : 10+10=20).**

მაგალითად, რიცხვი **1964** თვლის რომაულ სისტემაში ასე ჩაიწერება:

MCMLXIV (**M=1000; CM=900; LX=60; IV=4**).

აქ ცხრაასი მიიღება ათასიდან ასის (**M-C)** გამოკლებით; სამოცი მიიღება ორმოცდაათისა და ათის შეკრებით (**L+X)**, ხოლო ოთხი მიიღება ხუთიდან ერთის გამოკლებით (**V-I)**.

ზოგადად თვლის არაპოზიციური სისტემის გამოყენებით რიცხვების ჩაწერა და მათზე არითმეტიკული ოპერაციების შესრულების წესები რთულია, ამიტომ დღეისათვის თვლის პოზიციური სისტემებია გავრცელებული.

თვლის პოზიციური სისტემის დროს რიცხვის ჩანაწერში არსებული გარკვეული რაოდენობის აღმნიშვნელი სიმბოლოს (ციფრის) მნიშვნელობა რიცხვში მის მიერ დაკავებულ ადგილზე დამოკიდებულებით იცვლება. პოზიციური ნუმერაციის შექმნა შუმერებსა და ძველ ბაბილონელებს მიეწერება. ასეთი ნუმერაცია შემდეგ ინდუსებმა განავითარეს. თვლის პოზიციურ სისტემაში, როგორც ქვემოთ დავინახავთ, გადამწყვეტ როლს სისტემის **b** ფუძე თამაშობს, ამიტომ მათ ხშირად **b-**ურ სისტემებსაც უწოდებენ. გამოთვლით ტექნიკაში ძირითადად ასეთი სისტემებია გამოყენებული, ამიტომ მათ ცალკე განვიხილავთ;

თვლის შერეული სისტემა წარმოადგენს თვლის პოზიციური სისტემის განზოგადებას და არცთუ იშვიათად მათ თვლის პოზიციურ სისტემებსაც მიაკუთვნებენ. თვლის შერეული სისტემის **b** ფუძედ აიღება არა რომელიმე კონკრეტული რიცხვი, არამედ რიცხვების ზრდადი $\{b_k\}_{k=0}^{\infty}$ მიმდევრობა და მასში თითოეული **x** რიცხვი მასში წარმოდგენილია როგორც წრფივი კომბინაცია:

$$x = \sum_{k=0}^{n-1} a_k b_k \quad ,$$

სადაც a_k კოეფიციენტები, რომლებსაც ციფრებსაც უწოდებენ, გარკვეულ მოთხოვნებს უნდა აკმაყოფილებდს. თვლის შერულ სისტემაში x რიცხვის ჩანაწერს უწოდებენ k ინდექსის შემცირების კვალობაზე მისი ციფრების ჩამონათვალს, დაწყებულს პირველი არანულოვანი ციფრიდან; თვლის შერეული სისტემის ყველაზე ცნობილი მაგალითია წამებზე გადაყვანით **d** დღე-ღამის, **h** საათის, **m** წუთისა და **s** წამის გამოსახვა; იგი უდრის: **(d.24.60.60 + h. 60.60 + m. 60 + s)** წამს.

თვლის შერეული სისტემებია:

- თვლის **ფიბონაჩური სისტემა;** მასში **ფიბონაჩის რიცხვებია** გამოყენებული; **ფიბონაჩის რიცხვები** წარმოადგენს შემდეგი მიმდევრობის ელემენტებს:

0, 1. 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597 . . . ,

რომელშიც თითოეული მომდევნო რიცხვი წინა ორი რიცხვის ჯამის ტოლია;

● თვლის ფაქტორიალური სისტემა, რომელშიც ფუძედ ფაქტორიალების მიმდევრობებია გამოყენებული; ნატურალური **n** რიცხვის ფაქტორიალი აღინიშნება როგორც **n!** და იგი წარმოადგენს 1-დან დაწყებული **n**-ის ჩათვლით დამთავრებული ყველა ნატურალური რიცხვის ნამრავლს:

$$n! = 1 \cdot 2 \cdot \ldots \cdot n = \prod_{i=1}^{n} i.$$

მაგალითად: **6! = 1 ·2· 3 · 4· 5 · 6 = 720.**

განსაზღვრების ძალით ითვლება, რომ **0! = 1**

● თვლის მაიას ტომისეული სისტემა. მაიას ტომები, როგორც წინა პარაგრაფში აღვნიშნეთ, თვლის ოცობით სისტემას იყენებდნენ. ჩაწერისათვის გამოყენებულ ძირითად ნიშნებს წარმოადგენდა წერტილები (ერთიანები) და მონაკვეთები (ხუთეულები)

2.3. თვლის პოზიციური სისტემები

"თვლის ათობითი სისტემის უპირატესობა ზოოლოგიურია და არა მათემატიკური; ხელზე ათის ნაცვლად რვა თითი რომ გვქონოდა, მაშინ კაცობრიობა რვაობით სისტემას გამოიყენებდა."

ნ.ნ.ლუზინი (1883 – 1950)

"აუცილებელია შევიმეცნოთ ინდუქციის საშუალებით, ვინაიდან (სწორედ) ასეთი აღქმა წარმოშობს ზოგადს"

არისტოტელე (ჩვენს წ.აღ-მდე 384-322)

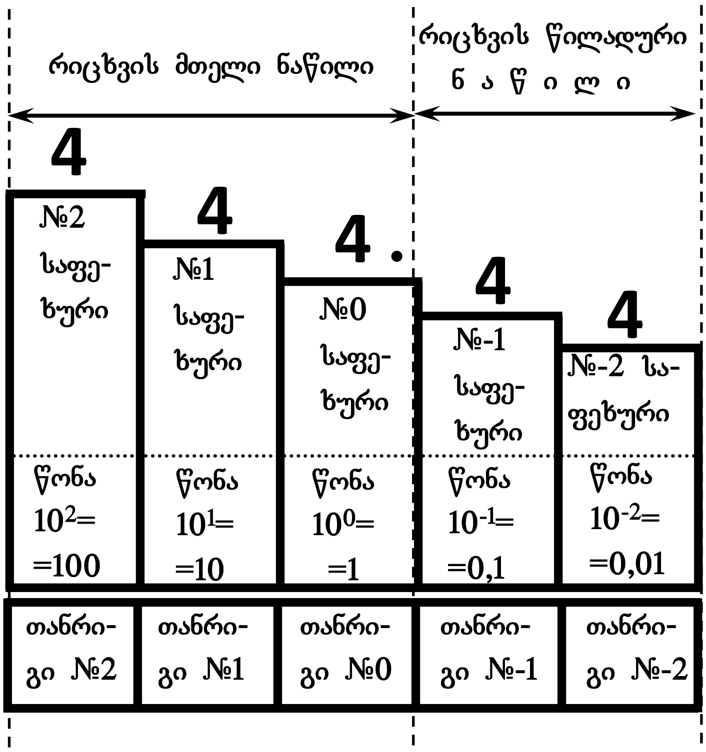
1 თვლის პოზიციურ სისტემაში, როგორც წინა პარაგრაფში აღვნიშნეთ, ციფრის მნიშვნელობა რიცხვის გამოსახულებაში მის მიერ დაკავებული ადგილის შესაბამისად განისაზღვრება.

თვლის მოცემულ სისტემაში ნებისმიერი რიცხვის გამოსახვისათვის გამოყენებული **სიმბოლოების** (ციფრების) მოწესრიგებულ $\mathcal{A}=\{a_0; a_1; \ldots ; a_n\}$ სიმრავლეს ეწოდება მოცემული სისტემის **ალფაბეტი,** ალფაბეტის სიმბოლოების (ციფრების) რაოდენობის გამომსახველ **b = n+1** რიცხვს – **ფუძე,** ხოლო თავად სისტემას – თვლის **b-ობითი სისტემა.**

თვლის პოზიციური **სისტემის ფუძე** ეწოდება თვლის მოცემულ სისტემაში რიცხვების გამოსახვისათვის გამოყენებულ სხვადასხვა სიმბოლოების (ციფრების) რაოდენობას.

თვლის ყველასათვის ნაცნობი პოზიციური სისტემაა სასკოლო არითმეტიკიდან ყველასათვის კარგად ცნობილი თვლის **10-ობითი (ათობითი) სისტემა**. ამ სისტემის ალფაბეტია $\mathcal{A}_{(10)}=\{ a_0=0; a_1=1; a_2=2; a_3=3; a_4=4; a_5=5\ a_6=6; a_7=7; a_8=8; a_9=9 \}$, ხოლო ფუძე – **b = 9 + 1=10,** ე.ი. ამ სისტემაში ნებისმიერი რიცხვის ჩასაწერად მხოლოდ ათი სხვადასხვა სიმბოლო (ციფრი) გამოიყენება. ამ ციფრებით ჩაიწერება ნულიდან დაწყებული ერთმანეთის მომდევნო პირველი ათი რიცხვი; რაც შეეხება **10-ზე** მეტ დანარჩენ რიცხვს, მათი ჩაწერისათვის ახალი ციფრების არ შემოტანა საჭირო არ არის.

თვლის ათობითი სისტემა იმაზეა დაფუძნებული, რომ თითოეული თანრიგის **10** ერთეული მეზობელი უფროსი თანრიგის ერთ ერთეულში ერთიანდება (ე.ი. მეზობელი უფროსი თანრიგის ერთი ერთეული უდრის წინა უმცროსი თანრიგის ათ ერთეულს), ამიტომ თითოეულ თანრიგს აქვს **10-ის** გარკვეული ხარისხის ტოლი წონა (მისი წონა **10-ის** გარკვეულ ხარისხში აყვანით მიიღება). აქედან გამომდინარე, ერთი და იგივე ციფრის მნიშვნელობა განისაზღვრება რიცხვის წერილობით გამოსახულებაში მის მიერ დაკავებული ადგილით, რომელიც რიცხვ **10-ის** ხარისხით ხასიათდება.



ნახ. 2.4. თვლის ათობითი სისტემის სტუქტურის აგების მაილუსტრირებელი სქემა

თვლის ათობითი სისტემის სტრუქტურის აგების მაილუსტრირებელი სქემა 2.4 ნახაზზე რიცხვ 444•44-ის მაგალითზეა მოყვანილი. აღნიშნული რიცხვის მთელი ნაწილის თანრიგები დანომრილია რიცხვებით 0; 1 და 2, ხოლო წილადური ნაწილი – რიცხვებით -1 და -2. თითოეულ თანრიგს შეესაბამება იგივე რიცხვებით დანომრილი სხვადასხვა სიმაღლის მქონე ვირტუალური ‘‘საფეხური’’, რომლებიც ‘‘დაკავებული’’ აქვთ რიცხვში შემავალ სიმბოლოებს (ციფრებს). **n (n∈{2;1;0;-1;-2})** ნომრის ვირტუალურ საფეხურს (და მაშასადამე, **n** ნომრის თანრიგს) გააჩნია 10^n-ის ტოლი ‘‘წონა’’. მასზე ‘‘მოკალათებული’’ (ე.ი. შესაბამის თანრიგზე მდგარი) ციფრის **სრული მნიშვნელობა** ამ ციფრის საკუთარი მნიშვნელობისა და საფეხურის წონის ნამრავლის ტოლია; მაგალითად, №2 საფეხურზე მოკალათებული (ე.ი. №2 თანრიგზე მდგარი) ციფრ 4-ის სრული მნიშვნელობა $4\ 10^2 = 4\ 100 = 400$-ის, ხოლო №-1 საფეხურზე მოკალათებული (ე.ი. №-1 თანრიგზე მდგარი) ციფრ 4-ის სრული მნიშვნელობა - $4 \cdot 10^{-1} = 4 \cdot 0{,}1 = 0{,}4$-ის ტოლია. ასე მიიღება განსახილველ რიცხვში შემავალი **თითოეული სიმბოლოს (ციფრის)** სრული მნიშვნელობა; **რიცხვის სრული მნიშვნელობა** უდრის მისი წარმომქმნელი სიმბოლოების (ციფრების) არითმეტიკულ ჯამს; ე.ი. განსახილველი 444.4 რიცხვის სრული მნიშვნელობა იქნება:

$$444{\bullet}4 = 4{\bullet}10^2 + 4{\bullet}10^1 + 4{\bullet}10^0 + 4{\bullet}10^{-1} + 4{\bullet}10^{-2} =$$
$$= 400 + 40 + 4 + 0{,}4 + 0{,}04.$$

ანალოგურად:

$$945 = 9 \cdot 10^2 + 4\ 10^1 + 5\ 10^0 = 900 + 40 + 5;$$

$$1304.5 = 1\cdot 10^3 + 3\cdot 10^2 + 0\cdot 10^1 + 4\cdot 10^0 + 5\cdot 10^{-1} =$$
$$= 1000 + 300 + 0 + 4 + 0{,}5;$$
$$40538.26 = 4\cdot 10^4 + 0\cdot 10^3 + 5\cdot 10^2 + 3\cdot 10^1 + 8\cdot 10^0 + 2\cdot 10^{-1} +$$
$$+ 6\cdot 10^{-2} = 40000 + 0 + 500 + 30 + 8 + 0{,}2 + 0{,}06$$

თანრიგების (ვირტუალური “საფეხურების”) წონების განმსაზღვრელი რიცხვების მიმდევრობას თვლის მოცემული **სისტემის ბაზისი** ეწოდება; თვლის **ათობითი სისტემის ბაზისს** წარმოადგენს **10**-ის ხარისხისაგან წარმოშობილი მიმდევრობა:

$$\ldots,\ 10^n, \ldots, 10^3, 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, \ldots, 10^m, \ldots$$

2 ზემოთ განხილული კერძო შემთხვევების განზოგადებიდან გამომდინარეობს, რომ **10**-ის ხარისხებად დაშლის გზით ნებისმიერი **n**-ნიშნა **A** რიცხვი შეიძლება წარმოვადგინოთ პოლინომის (მრავალწევრის) სახით:

$$A_{(10)} = a_{n-1}\,10^{n-1} + a_{n-2}10^{n-2} + \ldots + a_1\,10^1 + a_0\,10^0 +$$
$$+ a_{-1}10^{-1} + \ldots + a_{-m}10^{-m}, \qquad (2.1)$$

რომლის კოეფიციენტების მიმდევრობა წარმოადგენს n-ნიშნა **A** რიცხვის ათობით ჩანაწერს:

$$A_{(10)} = a_{n-1}\ a_{n-2} \ldots a_1\, a_0\ \ a_{-1} \ldots a_{-m} \ldots \qquad (2.2)$$

რიცხვის მთელი და წილადური ნაწილების განმაცალკევებელი წერტილი (სასკოლო არითმეტიკაში გამოყენებული მძიმის ნაცვლად) გამოიყენება ამ მიმდევრობაში თითოეული პოზიციის კონკრეტული მნიშვნელობების ფიქსირებისათვის და ათვლის წერტილს წარმოადგენს.

ზოგადად თვლის **b**-ობითი სისტემის ასაგებად აუცილებელია განისაზღვროს:

- **b**-ფუძე;
- **b** რაოდენობის სხვადასხვა სიმბოლოსაგან შედგენილი $\mathcal{A}$ ალფაბეტი;
- სისტემის $\ldots,\ Q^n, \ldots, Q^3, Q^2, Q^1, Q^0, Q^{-1}, Q^{-2}, \ldots, Q^m, \ldots$ ბაზისი; **b** და **Q** სიდიდეები, როგორც წესი, ერთმანეთს ემთხვევა, ე.ი. **b** = **Q;** თუმცა გვაქვს ამ წესიდან გადახრის შემთხვევებიც; მაგალითად, არსებობს სისტემები, რომლებშიც:

1) **Q** წარმოადგენს უარყოფით რიცხვს, ე.ი. $Q \in \{-1, -2, -3, \ldots\}$; ასეთ შემთხვევაში მიღებულ თვლის სისტემების სახელწოდებებს წინ ემატება პრეფიქსი “**ნეგო**”; ე.ი. მიიღება თვლის **ნეგო-ორობითი**, **ნეგო-სამობითი** და ა.შ. სისტემები;

2) **Q** წარმოადგენს რაციონალურ, ირაციონალურ, ტრანსცენდენტულ ან კომპლექსურ რიცხვს. ერთ-ერთი შემთხვევა, როდესაც $Q = \tau$, სადაც τ არის ოქროს კვეთის გამომხატველი $\dfrac{1+\sqrt{5}}{2}$ რიცხვი ჩვენ **2.1** პარაგრაფში განვიხილეთ; ასეთ სისტემას ეწოდება **ბერგმანის,** ანუ **τ–სისტემა.**

3 თვლის სისტემებში ფუძედ შეიძლება ნებისმიერი ნატურალური **b** რიცხვი გამოვიყენოთ. ალფაბეტად ჩვეულებრივ აიღება ერთმანეთის მომდევნო მთელი რიცხვები **0**-დან დაწყებული (**b-1**)-ის ჩათვლით დამთავრებული. თვლის ორობით სისტემაში ნებისმიერი რიცხვის ჩასაწერად გამოიყენება ციფრები **0,1,**

ე.ი. $\mathcal{A}_{(2)}$ ={0,1}; თვლის სამობით სისტემაში – ციფრები **0, 1, 2,** ე.ი. $\mathcal{A}_{(3)}$ ={0,1,2}; ხუთობით სისტემაში – ციფრები **0,1,2,3,4,** ე.ი. $\mathcal{A}_{(5)}$ ={0,1,2,3,4} და ა.შ. როდესაც არაბული ციფრები საკმარისი არ არის **b>10** ფუძის მქონე თვლის სისტემის ალფაბეტის ყველა სიმბოლოს აღნიშვნისათვის, მაშინ ციფრების აღნიშვნისათვის გამოიყენება ლათინური ასოები **A, B, C, D, E, F.**

ნატურალური ფუძის მქონე თვლის **b**-ობითი სისტემის $\mathcal{B}_{(b)}$ ბაზისს წარმოქმნის b^x, $x = \ldots, n, \ldots 2, 1, 0, -1, -2, \ldots, -m, \ldots$ რიცხვების მიმდევრობა.

2.3 ცხრილში მოყვანილია თვლის ზოგიერთი სისტემის პარამეტრები (ფუძეები, ალფაბეტები და ბაზისები). შევნიშნავთ, რომ **b > 10** - ფუძის მქონე თვლის სისტემის ალფაბეტებში გამოყენებული **A; B; C; D; E; F** ციფრებისათვის სამართლიანია ტოლობები **A = 10; B = 11; C = 12; D = 13; E = 14; F = 15.**

ცხრილი 2.3. თვლის ზოგიერთი სისტემების პარამეტრები

თვლის სისტემა	სისტემის ფ უ ძ ე	თვლის სისტემის ალფაბეტი / თვლის სისტემის ბაზისი
ორობითი	b=2	$\mathcal{A}_{(2)}$={0;1} $\mathcal{B}_{(2)} = 2^{n-1}, \ldots, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, \ldots, 2^m$
სამობითი	b=3	$\mathcal{A}_{(3)}$={0;1;2} $\mathcal{B}_{(3)} = 3^{n-1}, \ldots, 3^2, 3^1, 3^0, 3^{-1}, 3^{-2}, \ldots, 3^m$
ოთხობითი	b=4	$\mathcal{A}_{(4)}$={0;1;2;3} $\mathcal{B}_{(4)} = 4^{n-1}, \ldots, 4^2, 4^1, 4^0, 4^{-1}, 4^{-2}, \ldots, 4^m$
ხუთობითი	b=5	$\mathcal{A}_{(5)}$={0;1;2;3;4} $\mathcal{B}_{(5)} = 5^{n-1}, \ldots, 5^2, 5^1, 5^0, 5^{-1}, 5^{-2}, \ldots$
რვაობითი	b=8	$\mathcal{A}_{(8)}$={0;1;2;3;4;5;6;7} $\mathcal{B}_{(8)} = 8^{n-1}, \ldots, 8^2, 8^1, 8^0, 8^{-1}, 8^{-2}, \ldots, 8^m$
ათობითი	b=10	$\mathcal{A}_{(10)}$={0;1;2;3;4;5;6;7;8;9} $\mathcal{B}_{(10)} = 10^{n-1}, \ldots; 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, \ldots, 10^m$
თორმეტობითი	b=12	$\mathcal{A}_{(12)}$={0;1;2;3;4;5;6;7;8;9;A;B} $\mathcal{B}_{(12)} = 12^{n-1}, \ldots, 12^2, 12^1, 12^0, 12^{-1}, \ldots, 12^m$
თექვსმეტობითი	b=16	$\mathcal{A}_{(16)}$={0;1;2;3;4;5;6;7;8;9;A;B; C;D;E;F} $\mathcal{B}_{(16)} = 16^{n-1}, \ldots, 16^2, 16^1, 16^0, 16^{-1}, \ldots, 16^m$

4 მთელ რიცხვთა **Z** სიმრავლეში შემავალი ნებისმიერი **b** ∈ **Z** რიცხვი შეიძლება გამოვიყენოთ თვლის სისტემის ფუძედ; ეს საშუალებას გვაძლევს სათანადო $\mathcal{A}_{(b)}$ ალფაბეტისა და $\mathcal{B}_{(b)}$ ბაზისის შერჩევის შემდეგ ავაგოთ თვლის **b**-ობითი სისტემა. **Z** სიმრავლის უსასრულობიდან გამოდის, რომ არსებობს უსასრულო რა-

ოდენობის თვლის სისტემები. ნებისმიერი **b**$\in$ **Z** ფუძის მქონე თვლის სისტემაში **n**-ნიშნა **A(b)** რიცხვის ჩაწერა გულისხმობს შემდეგი სახის გამოსახულების შემოკლებული სახით ჩაწერას:

$$\mathbf{A_{(b)}}= a_{n-1}b^{n-1} + a_{n-1}b^{n-2} + \ldots + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + \ldots + a_{-m}b^{-m} = \sum_{k=-m}^{n-1} a_k b_k , \qquad (2.3),$$

სადაც a_k არის თვლის სისტემის ციფრები, ანუ $\mathcal{A}_{(b)}$ ალფაბეტში არსებული სიმბოლოები, n და m შესაბამისად მთელი და წილადური თანრიგების რაოდენობები, ხოლო **A(b)** – თვლის **b**-ობითი სისტემაში **A** რიცხვის ჩანაწერი.

თვლის **b**-ობითი სისტემაში **A** რიცხვის გამოსახულებას წარმოადგენს a_k ციფრების მიმდევრობა. მაგალითად, **b** = **12, 8, 4, 3, 2** ფუძეების მქონე სისტემებში ათობითი რიცხვ **35**-ის (შემოკლებულად აღვნიშნოთ, როგორც $35_{(10)}$) ჩანაწერებს ექნებათ **3.3** ცხრილში მოყვანილი ფორმები. ცხრილის თანახმად თვლის სხვადასხვა სისტემებში აღნიშნული რიცხვის ჩანაწერებს ექნება სახე: $\mathbf{2B_{(12)} = 35_{(10)} = 43_{(8)} = 203_{(4)} = 1022_{(3)} = 100011_{(2)}}$.

ცხრილი 2.4. ათობითი რიცხვი 35 თვლის სხვადასხვა სისტემებში

თვლის სისტემა	რიცხვების ჩაწერის პოლინომური ფორმა	რიცხვების ჩაწერის შემოკლებული ფორმა					
თორმეტობითი	$2\cdot 12^1 + B\cdot 12^0$					2	B
ათობითი	$3\cdot 10^1 + 5\cdot 10^0$					3	5
რვაობითი	$4\cdot 8^1 + 3\cdot 8^0$					4	3
ოთხობით	$2\cdot 4^2 + 0\cdot 4^1 + 3\cdot 4^0$				2	0	3
სამობითი	$1\cdot 3^3 + 0\cdot 3^2 + 2\cdot 3^1 + 0\cdot 2^5 + 2\cdot 3^0$			1	0	2	2
ორობითი	$1\cdot 2^5 + 0\cdot 2^4 + 0\cdot 2^5 + 0\cdot 2^5 + 1\cdot 2^5 + 1\cdot 2^5$	1	0	0	0	1	1
თ ა ნ რ ი გ ე ბ ი ს ნ უ მ ე რ ა ც ი ა $\longrightarrow$		5	4	3	2	1	0

2.4 ცხრილიდან ჩანს, რომ თვლის სისტემის ფუძის შემცირებით მცირდება გამოყენებული ციფრების რაოდენობა, მაგრამ იზრდება თანრიგების რაოდენობა; ასე, მაგალითად, რიცხვ **35**-ის ჩასაწერად თვლის თორმეტობით, ათობით და რვაობით სისტემებში ორ-ორი თანრიგია გამოყენებული, მაშინ როდესაც ოთხობით, სამობით და ორობით სისტემებში თანრიგების რაოდენობა შესაბამისად სამამდე, ოთხამდე და ექვსამდეა გაზრდილი.

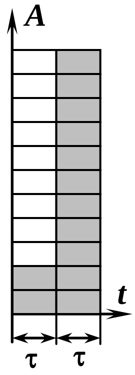
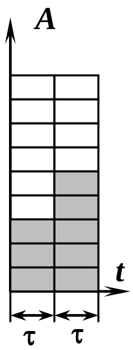
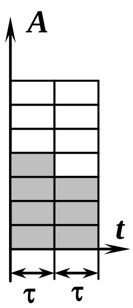
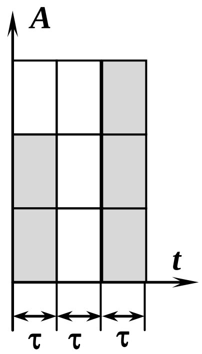
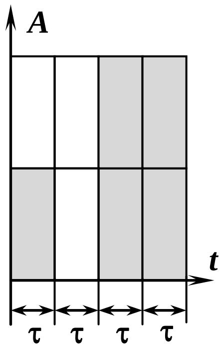
ინფორმაციის გადამცემ, შემნახველ და გარდამქმნელ სისტემებში თვლის სისტემების საფუძველზე სხვადასხვა სახის კოდები აიგება.

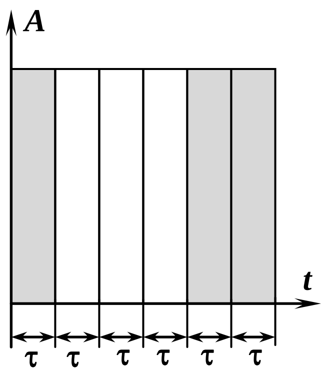
კოდი ეწოდება სხვადასხვა სახის ინფორმაციის წარმოდგენისათვის გამოყენებული პირობითი ნიშნების (სიმბოლოების) სისტემას.

ნებისმიერი **დისკრეტული შეტყობინებინებები** ან **შეტყობინებათა ნიშნები** შეიძლება რიცხვებით დავნომროთ, რაც რიცხვებით ამ ნიშნების შეცვლის შესაძლებლობას გვაძლევს. **ანალოგური (უწყვეტი) სიდიდეებიც** შეიძლება რიცხვების სახით წარმოვადგინოთ, თუ მათ სანიმუშო ზომებთან შედარების გზით გავზომავთ; მაშასადამე, ორივე შემთხვევაში შეტყობინებების გადაცემა ან შენახვა **რიცხვების** გადაცემასა და შენახვაზე დაიყვანე-

ბა. **რიცხვები** თავის მხრივ თვლის რაიმე სისტემის საშუალებით შეიძლება გამოისახოს. ამგვარად მიიღება **თვლის მოცემული სისტემის საფუძველზე აგებული კოდი**.

რიცხვის თითოეულ თანრიგს შეიძლება შევუთანადოთ ელექტრული სიგნალის რაიმე პარამეტრი, ვთქვათ, **A** ამპლიტუდა. **2.5** ნახაზზე მოყვანილია სხვადასხვა ამპლიტუდისა და **τ** ხანგრძლივობის მქონე იმპულსების საშუალებით რიცხვ **35**-ის გამოსახვის მაგალითი (თვლის სხვადასხვა სისტემების დროს).

თვლის სისტემის ფუძე	**b = 12**	**b = 10**	**b = 8**	**b = 4**	**b = 3**
რიცხვის ჩანაწერი (რიცხვის კოდი)	**2B**	**35**	**43**	**203**	**1022**
ელექტრული სიგნალები (კოდური კომბინაცია)					

თვლის სისტემის ფუძე	**b = 2**
რიცხვის ჩანაწერი (რიცხვის კოდი)	**100011**
ელექტრული სიგნალები (კოდური კომბინაცია)	

ნახ. 2.5. თვლის სხვადასხვა სისტემების დროს რიცხვ **35**-ის გამოსახვა სიგნალების სახით

ინფორმაციის გადამცემ, შემნახავ და გარდამქმნელ სისტემებში გამოყენების მოხერხებულობის თვალსაზრისით თუ გავაანალიზებთ თვლის სისტემებსა და მათ საფუძველზე აგებულ კოდებს, დავინახავთ, რომ რაც უფრო დიდია თვლის სისტემის **b** ფუძე, მით უფრო ნაკლები რაოდენობის თანრიგებია საჭირო მოცემული რიცხვის წარმოდგენისათვის. რიცხვის თანრიგების რაოდენობის შემცირება ამცირებს ამ რიცხვის გამომსახველი კოდის გადაცემისათვის საჭირო დროს; აქედან გამომდინარე, **თვლის სისტემის ფუძის გაზრდით მცირდება რიცხვის გამომსახველი კოდის გადაცემისათვის საჭირო დრო.**

სამაგიეროდ, თვლის სისტემის ფუძის გაზრდით მნიშვნელოვნად მაღლდება სხვადასხვა სიმბოლოების შესაბამისი ელემენტალური სიგნალების მაფორმირებელი და ამომცნობი აპარატურისადმი წაყენებული მოთხოვნები. კერძოდ, რაც უფრო მაღალია თვლის სისტემის **b** ფუძე, მით უფრო მეტი რაოდენობის მდგრადი მდგომარეობა უნდა ჰქონდეს გამომთვლელი მოწყობილობების ლოგიკურ ელემენტებს.

აღნიშნული გარემოებების გათვალისწინებით:

თვლის სისტემის ეფექტურობის მაჩვენებლად შეიძლება ავიღოთ რიცხვი, რომელიც უდრის სხვადასხვა სიმბოლოების **b** რაოდენობისა და ნებისმიერი რიცხვის გამოსახვისათვის საჭირო თანრიგების **N** რაოდენობის ნამრავლს.

ყველაზე ეფექტურ თვლის სისტემას წარმოადგენს ისეთი სისტემა, რომლის ეფექტურობის მაჩვენებელი მინიმალურია.

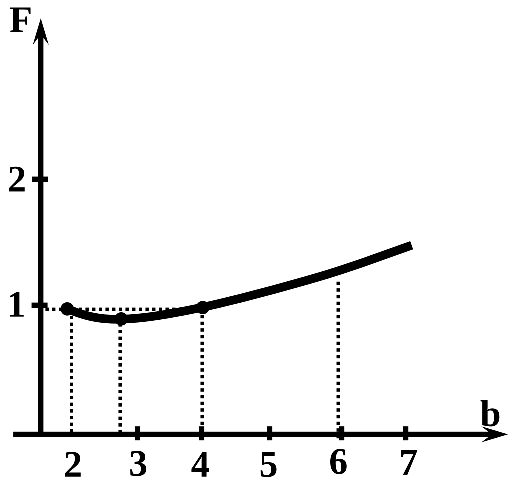
რიცხვების ჩასაწერად შერჩეული თვლის სისტემის **b** ფუძისა და თანრიგობრივი ბადის **N** სიგრძის ნამრავლი აღვნიშნოთ **C** სიმბოლოთი:

$$C = b\,N \tag{2.4}$$

რიცხვის თითოეული თანრიგის წარმოდგენა ხდება არა **b** რაოდენობის მქონე ერთი ელემენტით, არამედ **b** რაოდენობის ელემენტებით, რომელთაგანაც თითოეულს გააჩნია ერთი მდგრადი მდგომარეობა; აქედან გამომდინარე (2.4) გამოსახულება განსაზღვრავს მოწყობილობათა პირობით რაოდენობას, რომელიც თვლის გამოყენებული სისტემის შემთხვევაში საჭიროა რიცხვის წარმოდგენისათვის დაიხარჯოს. ამის გამო (2.4) გამოსახულებას **სისტემის ეკონომიურობის მაჩვენებელი** ეწოდება.

ცხრილი 2.5. **F =** ***f*** **(b)** ფუნქციის მნიშვნელობები

b	2	3	4	6	8	10
F	1,000	0,946	1,000	1,148	1,133	1,505



ნახ.2.6. თვლის სისტემის ფუძეზე შეფარდებითი ეკონომიურობის მაჩვენებლის დამოკიდებულება

N რაოდენობის თანრიგების მქონე მაქსიმალური რიცხვი, რომლის გამოსახვა შესაძლებელია **b** ფუძის მქონე თვლის სისტემის დახმარებით, განისაზღვრება გამოსახულებით

$$A_{b\,max} = b^N - 1. \tag{2.5}$$

(2.5) გამოსახულებიდან შეიძლება განვსაზღვროთ, თუ რა სიგრძე უნდა ჰქონდეს თანრიგობრივ ბადეს იმისათვის, რომ შესაძლებელი იყოს $A_{b\,max}$ რიცხვის წარმოდგენა:

$$N = \log_b (A_{b\,max} + 1). \tag{2.6}$$

უკანასკნელი გამოსახულების გათვალისწინებით (2.5) გამოსახულება იღებს შემდეგ სახეს:

$$C = b \log_b (A_{b\,max} + 1). \tag{2.7}$$

დავუშვათ, რომ **b** წარმოადგენს უწყვეტ სიდიდეს. მაშინ **C** სიდიდე შეიძლება **b** არგუმენტზე დამოკიდებულ ფუნქციად განვიხილოთ, ე.ი. გვექნება, რომ $\mathbf{C = \mathit{f}(b)}$; ზომის ერთეულად თუ მივიღებთ ერთი მდგრადი მდგომარეობის მქონე პირობით ელემენტს, მაშინ თვლის ორი სისტემის ურთიერთშედარებისათვის შეიძლება შემოვიტანოთ **ეკონომიურობის ფარდობითი მაჩვენებელი,** რომელიც საშუალებას მოგვცემს თვლის ნებისმიერი სისტემა შევადაროთ თვლის ორობით სისტემას **:**

$$\mathbf{F} = \frac{\mathbf{b \log_b (A_{b\,max} + 1)}}{\mathbf{[2 \log_b (A_{2\,max} + 1)]}} \;. \qquad \mathbf{(2.8)}$$

b არგუმენტის სიდიდის შესაბამისად ფუნქციის მნიშვნელობას ექნება **2.5** ცხრილში ნაჩვენები სახე.

2.6 ნახაზზე მოყვანილია თვლის სისტემის **b** ფუძეზე შეფარდებითი ეკონომიურობის **F** მაჩვენებლის დამოკიდებულება იმ დაშვებით, რომ ეს უკანასკნელი წარმოადგენს უწყვეტ ფუნქციას. გრაფიკის ქვედა წერტილი შეესაბამება $\mathbf{\frac{dF}{db} = 0}$ პირობის მიხედვით განსაზღვრულ $\mathbf{F = \mathit{f}(b)}$ ფუნქციის მინიმუმს, რაც **b = e = 2,72** –ის ტოლია.

აღნიშნულიდან გამომდინარეობს, რომ **პირობითი მოწყობილობების მინიმალური დანახარჯი გვაქვს თვლის სამობითი სისტემის გამოყენების შემთხვევაში.** ამ გარემოებას ჩვენ **2.1** პარაგრაფშიც გავუსვით ხაზი **მოსკოვის უნივერსიტეტის** პროფესორ **ნ.პ. ბრუსენცევის** ხელმძღვანელობით სამობითი კომპიუტერ **«Сетунь»**-ის დამუშავების ფაქტის აღნიშვნის დროს. სწორედ ამიტომ მიიჩნია **სტენფორდის უნივერსიტეტის** პროფესორმა **დონალდ ერვინ კნუტმა (1938 წ.)** სამობითი ლოგიკა ყველაზე ელეგანტურ და ეფექტურ ლოგიკად.

თვლის **სამობით სისტემას** უმნიშვნელოდ ჩამორჩება თვლის **ორობითი და ოთხობითი სისტემები.** გაცილებით ნაკლებ ეფექტურია ათობითი და სხვა სისტემები. ლოგიკური ელემენტების რეალზაციის მოხერხებულობისა და მათში არითმეტიკულ-ლოგიკური მოქმედებების შესრულების სიმარტივის ნიშნის მიხედვით, თვლის სისტემების შედარების შედეგად, დღეისათვის უპირატესობა **ორობით სისტემას მიენიჭა**. მართლაც:

- ორობითი სისტემის შესაბამის ლოგიკურ ელემენტებს საჭიროა მხოლოდ ორი მდგრადი მდგომარეობა გააჩნდეს;
- სიგნალების ერთმანეთისაგან განრჩევადობის ამოცანა ამ შემთხვევაში დაიყვანება სიგნალების გამომჟღავნებადობის (იმპულსის არსებობა-არარსებობის ფაქტის დადგენის) ამოცანამდე, რაც გაცილებით მარტივია;
- ორობით სისტემებში არითმეტიკული და ლოგიკური ოპერაციებიც ადვილი შესასრულებელია.

ზემოთ აღნიშნულის მიუხედავად, მხედველობიდან არ უნდა გამოგვრჩეს სამობითი სისტემის პერსპექტულობის ფაქტი.

2.4. თვლის ერთი სისტემიდან მეორე სისტემაში რიცხვების გადაყვანა

1

სხვადასხვა ამოცანის გადაწყვეტის დროს გვიხდება თვლის სხვადასხვა სისტემის გამოყენება. კერძოდ:

- ყოველდღიურ ცხოვრებაში ჩვენ ვიყენებთ თვლის **ათობით** (**b=10** ფუძის მქონე) სისტემას;

● კომპიუტერში საჭირო დამისამართების პროცესის რეალიზებისათვის და დაპროგრამირებისათვის სასურველია თვლის **თექვსმეტობით** (**b=16** ფუძის მქონე) სისტემის გამოყენება;

● გასული საუკუნის **50-60**-იან წლებში დაპროგრამებასა და კომპიუტერულ დოკუენტაციაში ფართოდ გამოიყენებოდა თვლის **რვაობითი** (**b=8** ფუძის მქონე) სისტემა, რომელიც დღეს პრაქტიკულად მთლიანად გამოდევნა თვლის თექვსმეტობითმა სისტემამ. მიუხედავად ამისა, ათობითი სისტემიდან რვაობით სისტემაში რიცხვების გადაყვანის ტრადიცია დღემდეა შენარჩუნებული კალკულატორებსა და დაპროგრამების მთელ რიგ ენებში. გარდა ამისა, რვაობითი სისტემა დღეს გამოიყენება **UNIX**-ის ტიპის ოპერაციულ (**LINUX, FreeBSD**) სისტემებში ფაილებითან და კატალოგებთან შეღწევის უფლების გასაზღვრისათვის;

● კომპიუტერებში მონაცემების დამუშავებისათვის ძირითადად გამოიყენება თვლის **ორობითი** (**b=2** ფუძის მქონე) სისტემა;

● ეკონომიურობის ფარდობითი მაჩვენებლის მინიმიზირებისათვის პერსპექტიულ სისტემას წარმოადგენს თვლის **სამობითი** (**b=3** ფუძის მქონე) სისტემა;

აღნიშნულიდან გამომდინარე, ჩვენ საკმაოდ ხშირად ვდგებით თვლის რომელიმე სისტემით წარმოდგენილი რიცხვის სხვა სისტემაში გადაყვანის აუცილებლობის წინაშე.

2 განვიხილოთ თვლის ერთ-ერთი სისტემიდან მეორე სისტემაში რიცხვების გადაყვანის ამოცანა. დავუშვათ, რომ ცნობილია **b** ფუძის მქონე თვლის სისტემაში n-ნიშნა $A_{(b)}$ რიცხვის ჩანაწერი:

$$A_{(b)}=a_{n-1}b^{n-1} + a_{n-1}b^{n-2} + \ldots + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + \ldots + a_{-m}b^{-m} = \sum_{k=-m}^{n-1} a_k b_k \qquad (2.9)$$

სადაც a_i არის **b**-ობითი სისტემის ციფრები ($0 \leq a_i \leq b-1$).

d ფუძის მქონე თვლის სისტემაში იგივე რიცხვის n-ნიშნა $A_{(d)}$ რიცხვის ჩანაწერს ექნება სახე:

$$A_{(d)} = c_{n-1}d^{n-1} + c_{n-2}d^{n-2} + \ldots + c_1d^1 + c_0d^0 + c_{-1}d^{-1} + \ldots + c_{-m}d^{-m} = \sum_{k=-m}^{n-1} c_k d_k, \qquad (2.10)$$

სადაც c_i არის თვლის **d**-ობითი სისტემის საძებნი ციფრები ($0 \leq c_i \leq d-1$). ამ დროს შეიძლება მხოლოდ დადებითი რიცხვების გამოყენების შემთხვევით დავკმაყოფილდეთ, რადგან ნებისმიერი რიცხვის გადაყვანა დაიყვანება ამ რიცხვის მოდულის გადაყვანამდე, რომელსაც საჭირო ნიშანი მიეწერება.

თვლის **b**-ობითი სისტემიდან თვლის **d**-ობითი სისტემაში რიცხვების გადაყვანა პირობითად აღვნიშნოთ როგორც:

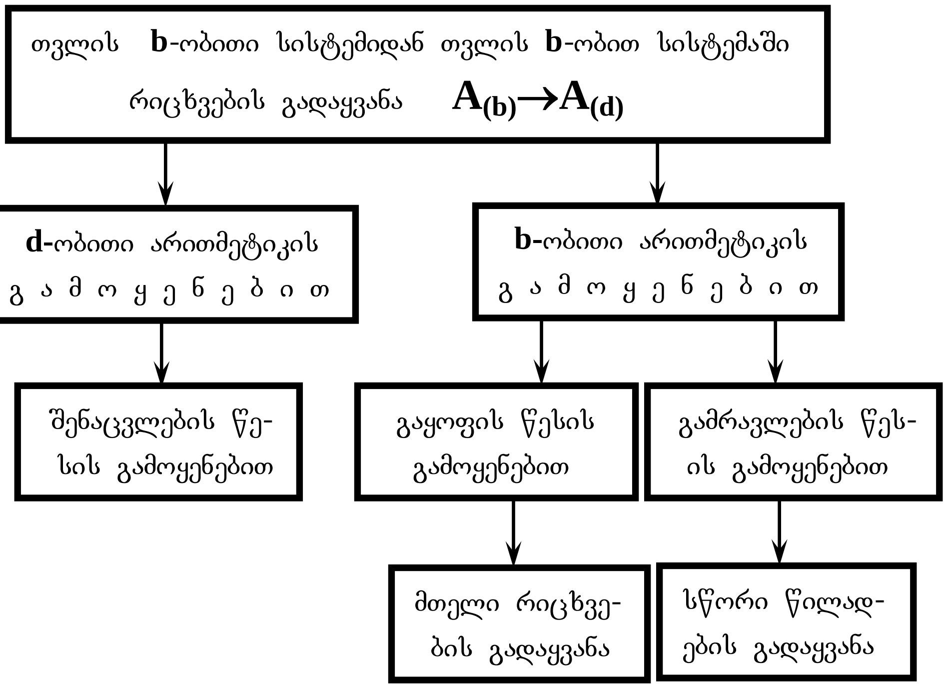
$$A_{(b)} \rightarrow A_{(d)} .$$

არითმეტიკას, რომელშიც **b**-ობით რიცხვები გამოიყენება, **b-ობითი არითმეტიკა** ეწოდება; ანალოგიურად, **d-ობითი არითმეტიკაში** გამოიყენება **d**-ობით რიცხვები. თვლის ერთი სისტემიდან მეორეში რიცხვების გადაყვანის დროს საჭიროა წინასწარ გავითვალისწინოთ, თუ რომელი არითმეტიკის გამოყენებით უნდა განვახორციელოთ გადაყვანისათვის საჭირო მოქმედებები.

სისტემიდან სისტემაში რიცხვების გადაყვანის დროს გამოყენებული მეთოდებისა და წესების კლასიფიკაცია **2.7** ნახაზზეა მოცემული. განვიხილოთ აღნიშნულ ნახაზზე მოყვანილი წესები.

3 თავდაპირველად განვიხილოთ თუ როგორ გადაიყვანება **b** ფუძის მქონე სისტემაში მოცემული ნებისმიერი **A** რიცხვი **d** ფუძის მქონე სისტემაში **შე-**

ნაცვლების წესის გამოყენებით. აღნიშნული წესი გულისხმობს თვლის **b** ფუძის მქონე სისტემისათვის შედგენილი (2.9) პოლინომის გამოთვლისათვის გამოყენებული იქნას თვლის **d** ფუძის მქონე სისტემა. სხვა სიტყვებით (2.9) გამოსახულების **d**-ობითი გამოსახულების მისაღებად აღნიშნულ გამოსახულებაში არსებული ყველა $\mathbf{a_i}$ ციფრი და **b** რიცხვი უნდა შევცვალოთ **d**–ობითი გამოსახულებებით და შევასრულოთ **d**–ობითი არითმეტიკის ოპერაციები (ე.ი აღნიშნულ გამოსახულებაში გამოყენებული **b**–ობითი არითმეტიკის ოპერაციები შევცვალოთ **d**–ობითი არითმეტიკის ოპერაციებით).



ნახ.2.7. რიცხვების სისტემიდან სისტემაში გადაყვანის წესების კლასიფიკაცია

შენაცვლების წესი ყველაზე ხშირად გამოიყენება თვლის ნებისმიერ სისტემით გამოსახული რიცხვების გადასდაყვანად თვლის ათობით სისტემაში.

თვლის **b**–ობით სისტემაში $\mathbf{A_{(b)} = a_{n-1}\, a_{n-2} \ldots a_1\, a_0\, a_{-1} a_{-m}}$ სახით ჩაწერილი n-ნიშნა **A** რიცხვის ათობით სისტემაში გადაყვანა დაიყვანება ათობითი არითმეტიკის გამოყენებით $\mathbf{A_{(10)} = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \ldots + a_1b^1 + a_0b^0 + a_{-1}b^{-1} + \ldots + a_{-m}b^{-m}}$ მრავალწევრის მნიშვნელობის გამოთვლამდე. მაგალითები 2.6 ცხრილშია მოყვანილი.

4

განვიხილოთ **გაყოფის წესის** გამოყენებით **b** ფუძის მქონე თვლის სისტემაში მოცემული მთელი **A** რიცხვის **d** ფუძის მქონე სისტემაში გადაყვანის შემთხვევა. მთელი რიცხვებისათვის (2.9) და (2.10) გამოსახულებები გარდაიქმნებიან შემდეგ გამოსახულებებად:

$$\mathbf{A_{(b)} = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \ldots + a_1b^1 + a_0}\,, \qquad \mathbf{(2.11)}$$

$$\mathbf{A_{(d)} = c_{n-1}d^{n-1} + c_{n-2}\,d^{n-2} + \ldots + c_1d^1 + c_0}\,. \qquad \mathbf{(2.12)}$$

სადაც $\mathbf{a_i}$ არის **b**-ობითი სისტემის ციფრები $\mathbf{(0 \le a_i \le b\text{-}1)}$, ხოლო $\mathbf{c_i}$ არის **d**-ობითი სისტემის ციფრები $\mathbf{(0 \le c_i \le d\text{-}1)}$.

რადგან $\mathbf{A_{(b)}=A_{(d)}}$, ამიტომ შეიძლება (2.11) გამოსახულების მარცხენა ნაწილი გავუტოლოთ (2.12) გამოსახულების მარჯვენა ნაწილს და დავწეროთ:

$$\mathbf{A_{(b)} = c_{n-1}d^{\,n-1} + c_{n-2}d^{\,n-2} + \ldots + c_1d^{\,1} + c_0}\,, \qquad \mathbf{(2.13)}$$

სადაც c_i არის თვლის **d**-ობითი სისტემის **საძებნი** ციფრები.

c_0 ციფრის განსაზღვრისათვის (2.13) ტოლობის ორივე მხარე გავყოთ **d** რიცხვზე; ამასთანავე, აღნიშნული ტოლობის მარცხენა მხარეზე არსებული $\mathbf{A_{(b)}}$ რიცხვის **d** რიცხვზე გაყოფისას ვისარგებლოთ **b**–ობითი არითმეტიკის წესებით (რადგან $\mathbf{A_{(b)}}$ რიცხვი თვლის **b**–ობითი სისტემის გამოყენებითაა ჩაწერილი). (2.13) გამოსახულება ზოგადად მიიღებს სახეს:

$$[A_{(b)}/d] = c_{n-1}d^{n-2} + c_{n-2}\,d^{n-3} + \ldots + c_1 + c_0\,/d\,. \qquad (2.14)$$

ცხრილი 2.6 შენაცვლების წესით რიცხვების სისტემიდან სისტემაში გადაყვანის მაგალითები

ოპერაცია	გადასაყვანი რ ი ც ხ ვ ი	b	შესასრულებელი მოქმედებები	შედეგი
$A_{(2)} \rightarrow A_{(10)}$	1 1 0 1, 1	(2)	$1\bullet 2^3 + 1\bullet 2^2 + 0\bullet 2^1 + 1\bullet 2^0 + 1\bullet 2^{-1}$	$13{,}1_{(10)}$
$A_{(3)} \rightarrow A_{(10)}$	2 0 2 1	(3)	$2\bullet 3^3 + 0\bullet 3^2 + 2\bullet 3^1 + 1\bullet 3^0$	$63_{(10)}$
$A_{(6)} \rightarrow A_{(10)}$	3 1 0 6	(6)	$3\bullet 6^3 + 1\bullet 6^2 + 0\bullet 6^1 + 6\bullet 6^0$	$690_{(10)}$
$A_{(8)} \rightarrow A_{(10)}$	2 5 6	(8)	$2\bullet 8^2 + 5\bullet 8^1 + 6\bullet 8^0$	$174_{(10)}$
$A_{(16)} \rightarrow A_{(10)}$	1 E 2	(16)	$1\bullet 12^2 + E\bullet 12^1 + 2\bullet 12^0$	$314_{(10)}$
თანრიგების ნუმერაცია ⇨	3 2 1 0 -1			

(2.14) გამოსახულების მარცხენა მხარეში მდგარ $[\mathbf{A_{(b)}/d}]$ რიცხვში ერთმანეთისაგან განვაცალკევოთ მთელი და წილადი ნაწილები:

$$[A_{(b)}/d] = [A_{(b)}/d]_{\text{მთელი}} + [A_{(b)}/d]_{\text{წილადი}}\,, \qquad (2.15)$$

სადაც $[\mathbf{A_{(b)}/d}]_{\text{მთელი}}$ არის განაყოფის მთელი ნაწილი, ანუ არასრული განაყოფი, ხოლო

$[A_{(b)}/d]_{\text{წილადი}} = \left|\dfrac{\text{ნაშთი}}{d}\right|$ – განაყოფის წილადური ნაწილი ანუ $\mathbf{A_{(b)}}$ რიცხვის **d** რიცხვზე გაყოფის შედეგად მიღებული **ნაშთი.**

(2.15) შევიტანოთ (2.14)-ში:

$$[A_{(b)}/d]_{\text{მთელი}} + \left|\frac{\text{ნაშთი}}{d}\right| = c_{n-1}d^{n-2} + c_{n-2}\,d^{n-3} + \ldots + c_1 + c_0\,/d \qquad (2.16)$$

მხედველობაში თუ მივიღებთ, რომ c_i<**d** (ე.ი. c_0/d წარმოადგენს (2.16) გამოსახულების წილადურ ნაწილს) და (2.16) გამოსახულებაში ერთმანეთს ცალ-ცალკე გავუტოლებთ მთელ და წილადურ ნაწილებს, მივიღებთ:

$$[A_{(b)}/d]_{\text{მთელი}} = c_{n-1}d^{n-2} + c_{n-2}\,d^{n-3} + \ldots + c_1\;, \qquad (2.17)$$

$$\left|\frac{\text{ნაშთი}}{d}\right| = c_0\,/d \qquad (2.18)$$

(2.18) გამოსახულებიდან გამოდის, რომ:

$$c_0 = \text{ნაშთს}, \qquad (2.19)$$

ე.ი. c_0 უდრის $\mathbf{A_{(b)}}$ რიცხვის **d** რიცხვზე გაყოფის შედეგად მიღებულ ნაშთს.

(2.17) გამოსახულების თანახმად $c_{n-1}d^{n-2} + c_{n-2}\, d^{n-3} + \ldots + c_1$ წარმოადგენს მთელ რიცხვს; მას თუ პირობითად აღვნიშნავთ A_d^1 რიცხვით, შეგვიძლია დავწეროთ:

$$A^1_{(b)} = A_{(b)}/d]_{\text{მთელი}} = c_{n-1}d^{\,n-2} + c_{n-2}\,d^{\,n-3} + \ldots + c_1 \,. \qquad (2.20)$$

რადგან $A^1_{(b)}$ მთელი რიცხვია, ამიტომ შემდეგი c_1 კოეფიციენტის განსასაზღვრის მიზნით მისთვისაც შეიძლება გავიმეოროთ იგივე პროცედურა და ა.შ. ეს პროცესი გაგრძელდება მანამ, სანამ არასრული განაყოფი ნულის ტოლი არ გახდება, ე.ი. სანამ არ შესრულდება პირობა:

$$[A^i_{(b)}/d]_{\text{მთელი}} = 0 \,. \qquad (2.21)$$

რადგან ყველა ოპერაცია სრულდება თვლის **b** ფუძის მქონე სისტემაში, ამიტომ საძებნი c_i კოეფიციენტები ამ სისტემით იქნება წარმოდგენილი; ამიტომ აუცილებელია ისინი **d**–ობითი ციფრებით ჩავწეროთ.

გაყოფის წესი ყველაზე ხშირად გამოიყენება მთელი ათობითი რიცხვების გადასაყვანად თვლის ნებისმიერ სხვა სიტემაში.

ამგვარად, თვლის **b**–ობითი სისტემიდან **b**–ობითი არითმეტიკის საშუალებებით მთელი დადებითი რიცხვების **d**–ობით სისტემაში გადაყვანის წესი შეიძლება ასეთი სახით ჩამოვაყალიბოთ.

თვლის b ფუძის მქონე სისტემიდან მთელი $A_{(b)}$ რიცხვი რომ გადავიყვანოთ თვლის d ფუძის მქონე სისტემაში აუცილებელია:

- $A_{(b)}$ *რიცხვი გავყოთ თვლის* **b** *ფუძის მქონე სისტემით გამოსახულ* **d** *რიცხვზე; მივიღებთ პირველი განაყოფის მთელ* $c_{n-1}d^{\,n-2} + c_{n-2}\,d^{\,n-3} + \ldots + c_1$ *ნაწილსა და* c_0 *ნაშთს. ეს უკანასკნელი წარმოადგენს თვლის ფუძის მქონე ჩაწერილი საძებნი რიცხვის* c_0 *ციფრს;*
- *მიღებული მთელი* $c_{n-1}d^{n-2} + c_{n-2}\,d^{n-3} + \ldots + c_1$ *ნაწილი მეორედ გავყოთ თვლის* **b** *ფუძის მქონე სისტემით გამოსახულ* **d** *რიცხვზე; მივიღებთ მე-2-ე განაყოფის მთელ* $c_{n-1}d^{n-3} + c_{n-2}\,d^{n-4} + \ldots + c_2$ *ნაწილსა და* c_1 *ნაშთს. ეს უკანასკნელი წარმოადგენს თვლის ფუძის მქონე ჩაწერილი საძებნი რიცხვის* c_1 *ციფრს;*
- *აღნიშნული ოპერაციის მე-3-ედ შესრულების შედეგად მივიღებთ საძებნი რიცხვის* c_2 *ციფრს, მე-4-ედ შესრულების შედეგად –* c_3 *ციფრს და ა.შ. მე-n-ედ* შესრულების შემდეგ – c_{n-1} ციფრს;
- $c_i,\ i=0,1,2,\ldots,(n-1)$ ციფრები ჩავწეროთ მათი მიღების შებრუნებული თანამიმდევრობით და მივიღებთ საძებნ n-ნიშნა $A_{(d)} = c_{n-1}\, c_{n-2} \ldots c_1\, c_0$ რიცხვს.

2.8,ა ნახაზზე ნაჩვენებია ათობითი რიცხვ **53**-ის ორობით სისტემაში გადაყვანის, ხოლო **2.8,ბ** ნახაზზე - ასევე ათობითი რიცხვ **75**-ის თექვსმეტობით სისტემაში გადაყვანის მაგალითები. გადაყვანის დროს შესასრულებელი ოპერაციები აღნიშნულ ნახაზზეა ნაჩვენები. როგორც აღნიშნული ნახაზებიდან ჩანს $53_{(10)} = 110101_{(2)}$ და $75_{(10)} = 4B_{(16)}$.

განვიხილოთ **გამრავლების წესის** გამოყენებით **b** ფუძის მქონე თვლის სისტემაში მოცემული სწორი $A_{(b)}$ წილადის **d** ფუძის მქონე სისტემაში გადაყვანის შემთხვევა. სწორი წილადებისათვის (2.9) და (2.10) გამოსახულებები გარდაიქმნება შემდეგ გამოსახულებებად:

$$A_{(b)} = a_{-1}b^{-1} + a_{-2}b^{-2} + \ldots + a_{-m}b^{-m} + \ldots\,, \qquad (2.22)$$

$$A_{(d)} = c_{-1}d^{-1} + c_{-2}\,d^{-2} + \ldots + c_{-m}d^{-m} + \ldots\,, \qquad (2.23)$$

სადაც a_i არის **b**-ობითი სისტემის ციფრები $(0 \le a_i \le b\text{-}1)$, ხოლო c_i არის **d**-ობითი სისტემის ციფრები $(0 \le c_i \le d\text{-}1)$.

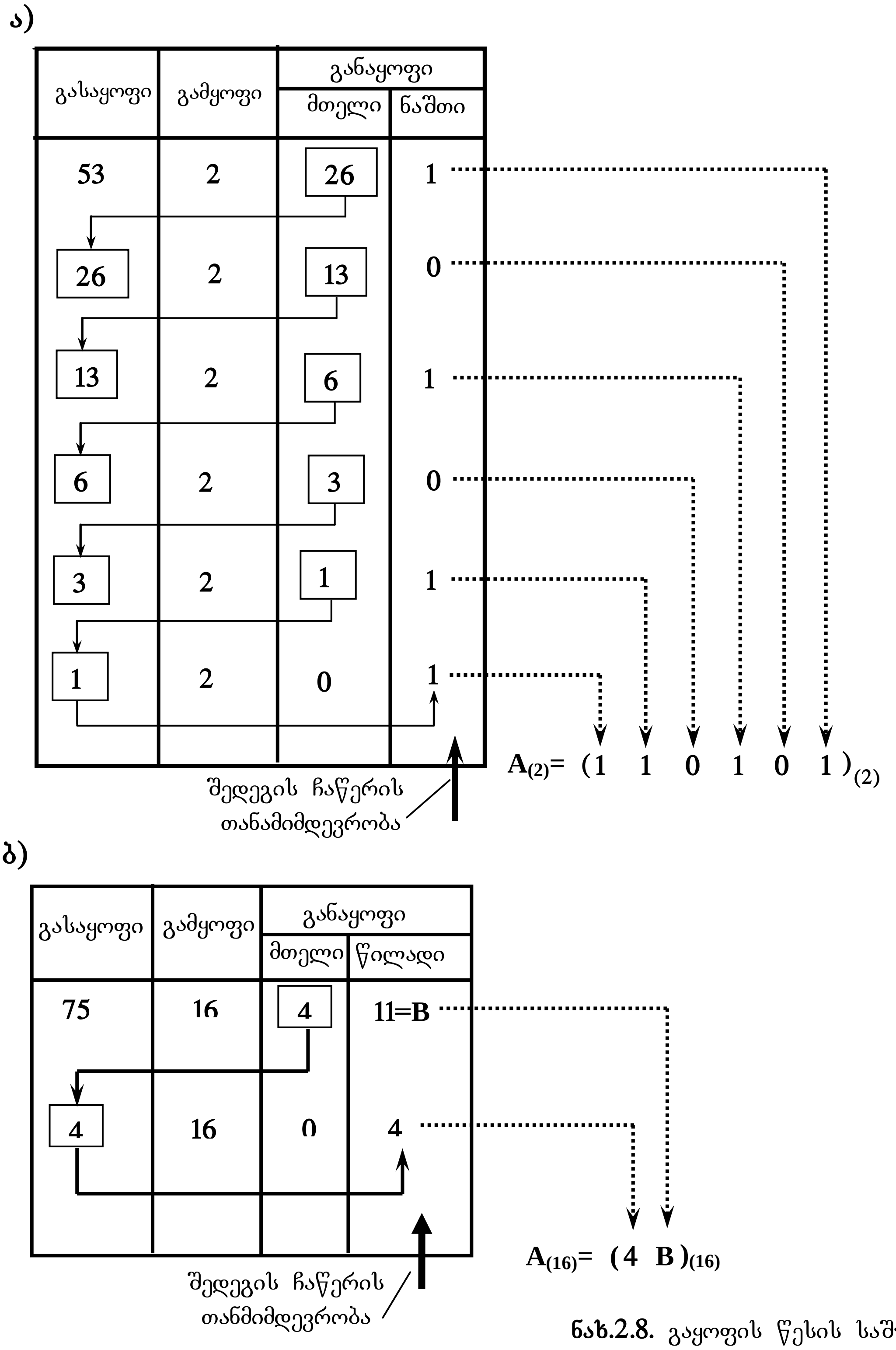
რადგან $A_{(b)}=A_{(d)}$, ამიტომ შეიძლება (2.22) გამოსახულების მარცხენა ნაწილი გავუტოლოთ (2.23) გამოსახულების მარჯვენა ნაწილს და დავწეროთ:

$$A_{(b)} = c_{-1}d^{-1} + c_{-2}\,d^{-2} + \ldots + c_{-m}d^{-m} + \ldots\,, \qquad (2.24)$$

სადაც c_i არის თვლის **d**-ობითი სისტემის **საძებნი** ციფრები.

c_{-1}-ის განსასაზღვრავად (3.24) ტოლობის ორივე ნაწილი გავამრავლოთ **d** რიცხვზე; ამასთანავე, მარცხენა ნაწილში გამრავლების ოპერაციის შესრულებისას ვისარგებლოთ *b*–ობითი არითმეტიკის წესებით (რადგან ჩვენთვის ცნობილია თვლის *b*–ობით სისტემაში $A_{(b)}$ რიცხვის ჩანაწერი). რადგან $d^{-1}d=1, d^{-2}d=d^{-1}, \dots, d^{-m}d=d^{-m+1}$, ამიტომ მივიღებთ:

$$[A_{(b)}\bullet d] = c_{-1}+c_{-2}d^{-1}+\dots+c_{-m}d^{-m+1}+\dots, \qquad (2.25)$$



ნახ.2.8. გაყოფის წესის საშუალებით სისტემიდან სისტემაში მთელი რიცხვების გადაყვანის მაგალითები

$[A_{(b)}\ d]$ ნამრავლში გამოვყოთ მთელი $[A_{(b)}\ d]_{მთელი}$ და წილადური $[A_{(b)}\ d]_{წილადი}$ ნაწილები:

$$[A_{(b)}\bullet d] = [A_{(b)}\ d]_{მთელი} + [A_{(b)}\ d]_{წილადი}. \qquad (2.26)$$

(2.25) გამოსახულებაში მთელი რიცხვია c_{-1}, ხოლო წილადური რიცხვი - $c_{-2}\,d^{12} + \ldots + c_{-m}d^{-m+1} + \ldots$, ამიტომ შეგვიძლია დავწეროთ (2,26) ტოლობის მარჯვენა ნაწილში არსებული თითოეული შესაკრების გამოსახულება:

$$[A_{(b)}\bullet d]_{მთელი} = c_{-1}\ , \qquad (2.27)$$

$$[A_{(b)}\bullet d]_{წილადი} = c_{-2}\,d^{-1} + \ldots + c_{-m}d^{-m+1} + \ldots \qquad (2.28)$$

ამგვარად, (2.26) მრავალწევრის უმცროსი c_{-1} კოეფიციენტი განისაზღვრება თანაფარდობით:

$$c_{-1} = [A_{(b)}\bullet d]_{მთელი}.$$

$[A_{(b)}d]_{წილადი}$ წილადური ნაწილი აღვნიშნოთ $A^{1}_{(b)}$ სიმბოლოთი და (2.28) გამოსახულება ასე გადავწეროთ:

$$A^{1}_{(b)} = c_{-2}\,d^{-1} + \ldots + c_{-m}d^{-m+1} + \ldots \qquad (2.29)$$

$A^{1}_{(b)}$ გამოსახულება სწორი წილადი და c_{-2} კოეფიციენტის განსასაზღვრავად მის მიმართ შეიძლება გავიმეოროთ ზემოთ აღწერილი პროცედურა; კერძოდ იგი შეიძლება d რიცხვზე გავამრავლოთ.

ეს პროცესი მანამ გაგრძელდება, სანამ წილადური ნაწილის ნამრავლი ნულის ტოლი არ გახდება, ე.ი. სანამ არ შესრულდება პირობა:

$$[\ A^{i}_{(b)}\bullet d\ ]_{წილადი} = 0, \qquad (2.30)$$

ან სანამ არ იქნება მიღწეული რიცხვის წარმოდგენის მოთხოვნილი სიზუსტე.

ვინაიდან ყველა ზემოთ აღწერილი ოპერაცია სრულდებოდა **b**-არითმეტიკის წესების გამოყენებით, ამიტომ მიღებული საძებნი c_i კოეფიციენტები წარმოდგენილი იქნება თვლის **b** ფუძის მქონე სისტემის საშუალებით, ამიტომ აღნიშნული კოეფიციენტები საჭიროა **d**-ობითი რიცხვებად გარდავქმნათ.

გამრავლების წესი ყველაზე ხშირად გამოიყენება ათობით სისტემით გამოსახული წესიერი წილადების ნებისმიერ სხვა სისტემაში გადასაყვანად.

7

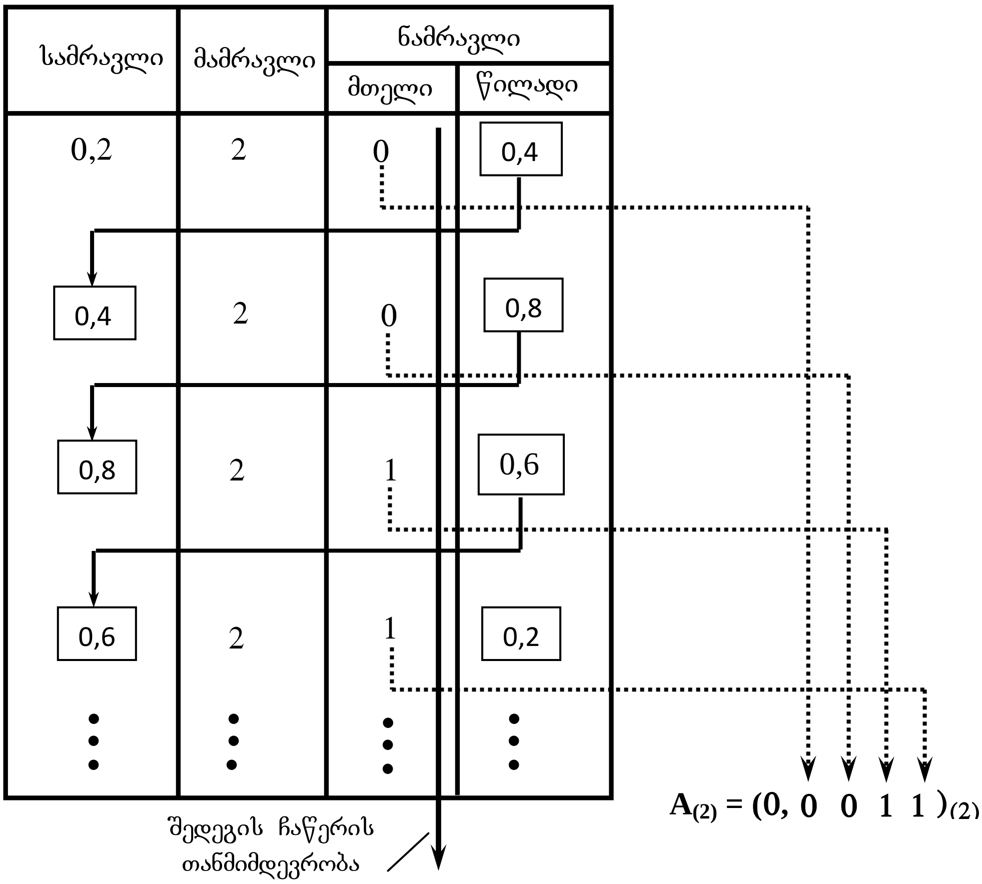
ამგვარად, თვლის **b**–ობითი სისტემიდან **b**–ობითი არითმეტიკის საშუალებებით წესიერი წილადების **d**–ობით სისტემაში გადაყვანის წესი შეიძლება ასეთი სახით ჩამოვყალიბოთ.

სწორი $A_{(b)}$ წილადი თვლის b-ობითი სისტემიდან თვლის d-ობით სისტემაში რომ გადავიყვანოთ საჭიროა:

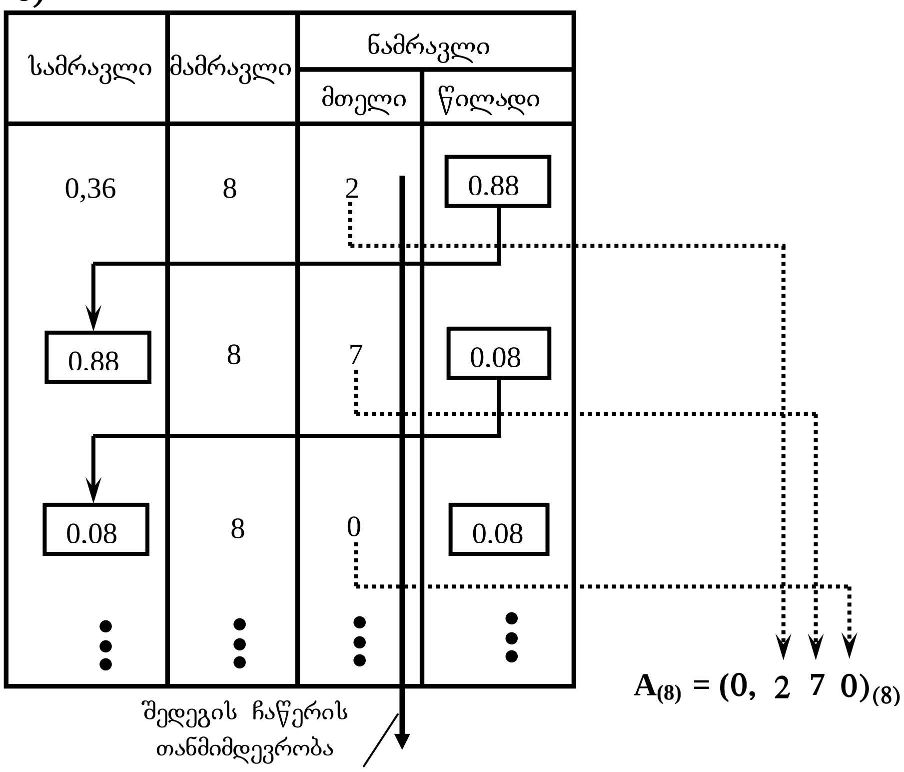
- $A_{(b)}$ რიცხვი გავამრავლოთ **b**–ობითი *სისტემის გამოყენებით ჩაწერილ* **d** რიცხვზე;
- მიღებული ნამრავლის წილადური ნაწილი ხელახლა გავამრავლოთ რიცხვზე და ეს *პროცესი გავაგრძელოთ მანამ, სანამ მორიგი მიღებული წილადური ნაწილი ნულის ტოლი არ გახდება ან არ იქნება მიღწეული* **d**–*ობით სისტემაში* $A_{(b)}$ *რიცხვის გამოსახვის მოთხოვნილი სიზუსტე;*
- *გამრავლების შედეგად მიღებული რიცხვების მთელი ნაწილები* **d**–*ობითი ციფრებით გამოვსახოთ; შევადგინოთ ამ ციფრების მიმდევრობა. მიმდევრობაში ციფრები უნდა შევიდეს მათი შესაბამისი მთელი რიცხვების მიღების რიგითობის დაცვით;*
- **d**–*ობითი ციფრებით ფორმირებული მიმდევრობა წარმოადგენს თვლის* **d**–*ობით სისტემაში გადაყვანილ* $A_{(d)}$ *რიცხვს;*
- *რიცხვის გადაყვანის სიზუსტე თუ მძიმის შემდეგ* j *რაოდენობის ნიშნის არსებობას მოითხოვს, მაშინ* ***გადაყვანის ზღვრული აბსოლუტური ცდომილება გამოითვლება ფორმულით:***

$$Q = \frac{d^{-(j+1)}}{2} \qquad (2.29)$$

ა)



ბ)



ნახ.2.9. გამრავლების წესის საშუალებით სისტემიდან სისტემაში სწორი წილადების გადაყვანის მაგალითები

2.9 ნახაზზე მოყვანილია ათობითი $A_{(10)}$=**0,2** და $A_{(10)}$ =**0,36** რიცხვების შესაბამისად თვლის ორობით და რვაობით სისტემებში გადაყვანის მაგალითები. გადაყვანის დროს შესასრულებელი ოპერაციები აღნიშნულ ნახაზზეა ნაჩვენები.

2.9,ა ნახაზიდან ჩანს, რომ ათობითი $A_{(10)}$=**0,2** რიცხვი წარმოადგენს ზუსტ რიცხვს, მაგრამ ორობით სისტემაში გადაყვანის შემდეგ იგი პერიოდულ $A_{(2)}$=**0,(0011)** ათწილადად გადაიქცევა (ფრჩხილებში მოთავსებულია წილადის პერიოდი).

2.9,ბ ნახაზის თანახმად ათობით $A_{(10)}$=**0,36** რიცხვს თუ რვაობით სისტემაში გადავიყვანთ, მაშინ მივიღებთ $A_{(8)}$=**0,270** რიცხვს; ამ დროს გადაყვანის ზღვრული აბსოლუტური ცდომილება იქნება:

$$\frac{8^{-4}}{2} = 2^{-13}$$

8 განსაკუთრებულ ყურადღებას იმსახურებს ***b*** ფუძის მქონე თვლის სისტემიდან რიცხვის ისეთი ***d*** ფუძის მქონე სისტემაში და პირიქით გადაყვანის შემთხვევა, როდესაც ***b*** და ***d*** რიცხვები ერთმანეთთან დაკავშირებულია თანაფარდობით:

$$b = d^K,$$

სადაც მთელი დადებითი რიცხვია. ასეთ შემთხვევაში ***d***–ობითი სისტემიდან ***b***-ობით სისტემაში რიცხვების გადასაყვანად, ე.ი. $A_{(d)}\rightarrow A_{(b)}$ ოპერაციის შესასრულებლად, საჭიროა შევასრულოთ შემდეგი ოპერაციები:

- *საწყის **d**–ობით სისტემაში წარმოდგენილი $A_{(d)}$ რიცხვის ჩანაწერში არსებული ციფრები მძიმიდან მარცხნივ და მარჯვნივ დავაჯგუფოთ **K** რაოდენობის ციფრებისაგან შემდგარ მიმდევრობებად (საჭიროების შემთხვევაში უფროსი თანრიგის მარცხნივ ან უმცროსი თანრიგის მარჯვნივ შესაბამისად დავუმატოთ საჭირო რაოდენობის ნულები);*
- *თითოეული მიმდევრობა ჩავწეროთ თვლის **b**–ობითი სისტემის ერთი ციფრით; მივიღებთ საძებნ რიცხვს;*

b-ობითი სისტემიდან ***d***–ობით სისტემაში უკუგადასვლისათვის, ე.ი. $A_{(b)}\rightarrow A_{(d)}$ ოპერაციის შესასრულებლად, თვლის ***b***–ობითი სისტემის თითოეული ციფრი უნდა შევცვალოთ მისი ***d*** –ობითი გამოსახულებებით.

მაგალითად:

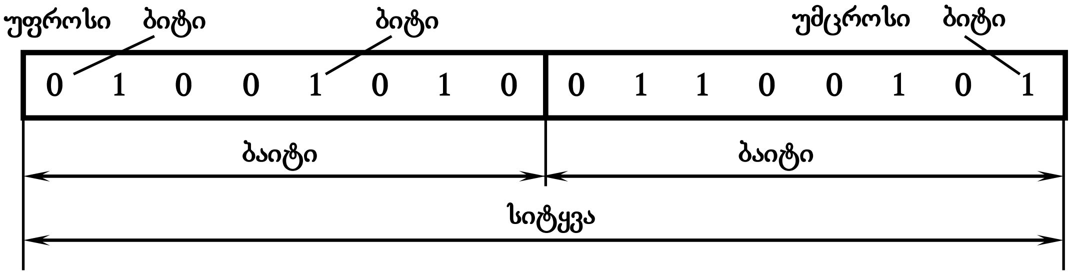
$$0001\ 1001\ 1001\ 0110_{(2)} = 1996_{(16)};$$
$$100\ 010\ 101\ 110_{(2)} = 4256_{(8)};$$
$$01\ 21\ 21\ 12_{(4)} = 1996_{(16)};$$
$$525_{(9)} = 12\ 01\ 12_{(3)};$$
$$B58E3_{(16)} = 1011\ 0101\ 1000\ 1110\ 0011_{(2)}.$$

2.5. კომპიუტერებში რიცხვითი ინფორმაციის წარმოდგენა

1 კომპიუტერის მეხსიერებაში ინფორმაცია ჩაიწერება ორობითი რიცხვების სახით (იხ. პარაგრაფი **1.3**), რომელთა ერთობლიობა წარმოქმნის **ციფრულ ორობით კოდს** (იხ. პარაგრაფი **2.3,** ნახ.**2.5**). ორობითი ინფორმაციის შესანახად კომპიუტერის სტრუქტურაში რეალიზებულია დიდი რაოდენობის მეხსიერების უჯრედები და რეგისტრები (ინგლ. **regestr** – ‘‘შეტანილი’’, ‘‘ჩაწერილი’’).

მეხსიერების უჯრედი ეწოდება კომპიუტერის დამხსომებელი მოწყობილობის დამისამართებად მინიმალურ ელემენტს, ხოლო **რეგისტრი – n** თანრიგიანი ორობითი რიცხვების შენახვისა და მათზე გარკვეული ოპერაციების შესასრულებლად გამოყენებულ მიმდევრობითი ან პარალელური კონფიგურაციის მქონე ლოგიკურ (ორობით) მოწყობილობას.

აღნიშნულ უჯრედების უმრავლესობას **n**-ის ტოლი სიგრძეები აქვთ, ე.ი. ისინი **n** რაოდენობის **ბიტების** შესანახად გამოიყენება (**ბიტი** ეწოდება ერთ ორობით თანრიგს). ასეთ უჯრედში შენახულ ინფორმაციას ეწოდება **სიტყვა.** რვა ბიტისაგან შემდგარ ორობით რიცხვს **ბაიტი** ეწოდება. ორი ბაიტისაგან შემდგარი ორობითი რიცხვი 2.10 ნახაზზეა ნაჩვენები.

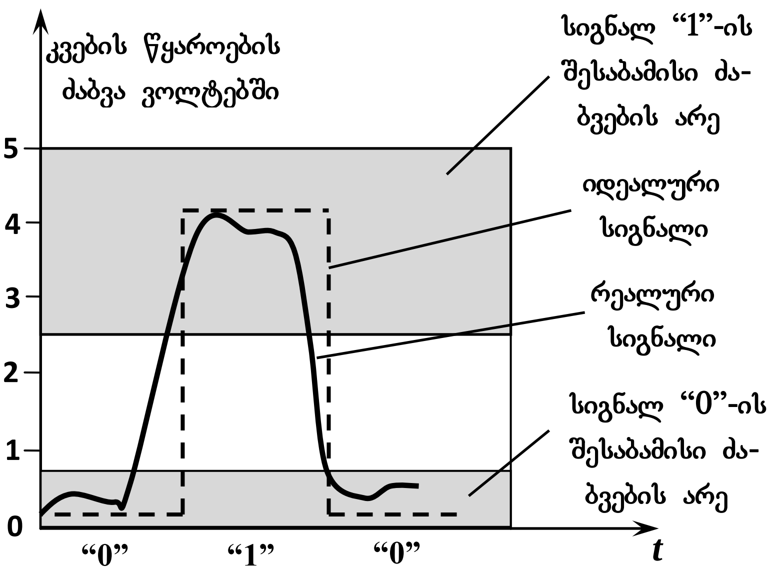


ნახ.2.10. ბიტის, ბაიტის და სიტყვის მაილუსტრირებელი სქემა

მეხსიერების უჯრედები და რეგისტრები მეხსიერების ელემენტებისაგან შედგება. მეხსიერების აღნიშნულ ელექტრონულ ელემენტს ორი მდგრადი მდგომარეობა აქვს და დროის ნებისმიერ მომენტში ერთ-ერთ ასეთ მდგომარეობაშია: ტრანზისტორი იმყოფება გამტარ ან გაუმტარ მდგომარეობაში, კონდენსატორი დამუხტულია ან განმუხტულია, სპეციალურ ნახევარგამტარულ მასალას გააჩნია მაღალი ან დაბალი კუთრი წინაღობა და ა.შ. აღნიშნული ფიზიკური მდგომარეობიდან ერთ-ერთი წარმოშობს მეხსიერების ელემენტის მაღალი დონის გამოსასვლელ ძაბვას, ხოლო მეორე მათგანი – დაბალი დონის გამოსასვლელ ძაბვას. **მაღალი დონის ძაბვად** ჩვეულებრივ ითვლება 4-5 ვოლტის სიდიდის, ხოლო **დაბალი დონის ძაბვად – 0 ვოლტის** სიდიდის ძაბვა. მაღალი დონის ძაბვა ჩვეულებრივ **ლოგიკურ** (**ორობით**) ერთიანად, ხოლო დაბალი დონის ძაბვა – **ლოგიკურ (ორობით) ნულად** მიიღება (შესაძლებელია შებრუნებული კოდირების გამოყენებაც, როდესაც ლოგიკურ ერთიანად თვლიან დაბალი დონის, ხოლო ლოგიკურ ნულად – მაღალი დონის ძაბვას).

2.11 ნახაზზე ნაჩვენებია გარკვეული შესასვლელი სიგნალის ზემოქმედების გავლენით მეხსიერების ასეთი ელემენტის (მაგალითად, რეგისტრის ერთი თანრიგის) მდგომარეობის შეცვლისას მის გამოსასვლელზე ფორმირებული სიგნალი. 0-იდან 1-ში და პირიქით 1-დან 0-ში გადასვლა მყისიერად არ ხდება, მაგრამ დროის გარკვეულ მომენტებში ეს სიგნალი იღებს მნიშვნელობებს, რომლებსაც კომპიუტერის ელემენტები აღიქვამს როგორც 0-ს ან როგორც 1-ს.

კომპიუტერის მეხსიერება შეიცავს სასრული რაოდენობის სიტყვებს, ხოლო სიტყვები შედგება სასრული რაოდენობის ბიტებისაგან, ამიტომ კომპიუტერში წარმოსადგენი ინფორმაციის მოცულობა მისი მეხსიერების ტევადობითაა შეზღუდული, ხოლო ციფრული ინფორმაცია კომპიუტერში მხოლოდ გარკვეული სიზუსტით შეიძლება იქნეს წარმოდგენილი;

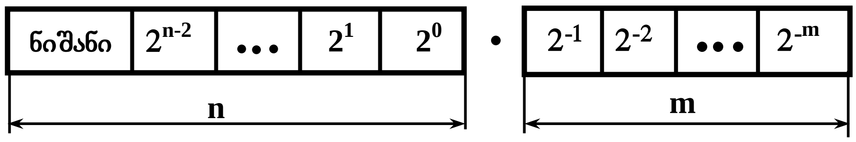


ნახ.2.11. ორობითი სიგნალის გრაფიკული წარმოდგენა

კომპიუტერში გამოიყენება ორობითი რიცხვების წარმოდგენის შემდეგი ორი ფორმა:

- **ბუნებრივი ფორმა** ანუ **ფიქსირებული წერტილიანი (მძიმიანი) ფორმა**;
- **ნორმალური ფორმა** ანუ **მცურავი წერტილიანი (მძიმიანი) ფორმა**

მოკლედ განვიხილოთ თითოეული მათგანი.



ნახ. 2.12. თანრიგობრივი ბადე ფიქსირებული წერტილიანი ფორმისათვის

3

ფიქსირებული წერტილის (მძიმის) გამოყენების შემთხვევაში რიცხვები გამოისახება ციფრების მიმდევრობის სახით, რომელშიც ნებისმიერი რიცხვის მთელი და წილადური ნაწილების განმაცალკევებელ წერტილს (მძიმეს) მუდმივად **ერთი და იგივე ადგილი** უკავია.

ფიქსირებული წერტილიანი რიცხვებისათვის განკუთვნილ თანრიგობრივ ბადეს კომპიუტერში ზოგადად აქვს **2.12** ნახაზზე ნაჩვენები სახე. ასეთი ფორმის გამოყენების დროს რიცხვების მოდულის დიაპაზონია:

$$2^{-m} \le |A| \le 2^{n} - 2^{-m}. \qquad (2.30)$$

რომელიმე ოპერაციის ჩატარების შემდეგ თუ მიიღება რიცხვი, რომელიც ცდება დასაშვებ დიაპაზონს, მაშინ **თანრიგობრივი ბადე გადაივსება,** რაც დაარღვევს კომპიუტერის ნორმალურ მუშაობას. თანამედროვე კომპიუტერებში რიცხვების წარმოდგენის ბუნებრივი ფორმა დამხმარე ფორმას წარმოადგენს და მხოლოდ მთელი რიცხვებისათვის გამოიყენება.

ცხრილი 2.7. უნიშნო მთელი რიცხვების მნიშვნელობათა დიაპაზონები

ბაიტების რაოდენობა	დიაპაზონი	
	ჩანაწერი ხარისხით	ჩვეულებრივი ჩანაწერი
1	$0 \cdots 2^8 - 1$	$0 \cdots 255$
2	$0 \cdots 2^{16} - 1$	$0 \cdots 65535$

4

მცურავი წერტილის (მძიმის) გამოყენების დროს თითოეული რიცხვი ციფრების ორი ჯგუფის სახით გამოისახება. ციფრების პირველ ჯგუფს **მანტისა**, ხოლო მეორე ჯგუფს – **ხარისხი** ეწოდება; მანტისის აბსოლუტური სიდიდე (მოდული) ნულზე ნაკლები უნდა იყოს, ხოლო ხარისხად მხოლოდ მთელი რიცხვი უნდა იყოს გამოყენებული. ზოგადად მცურავ წერტილიან (მძიმიან) რიცხვს აქვს შემდეგი ფორმა:

$$A = \pm M\, b^{\pm p}, \qquad (2.31)$$

სადაც **M** არის რიცხვის მანტისა (ლათ. **mantissa** – "დართვა", "დანამატი"); იგი წარმოადგენს რიცხვს, რომლის აბსოლუტური ნიშანი ერთზე ნაკლებია, ე.ი. $|M| < 1$; **p** –რიცხვის ხარისხი (**p** მთელი რიცხვია); **b** – თვლის სისტემის ფუძე.

თვლის ორობითი სისტემის დროს (ე.ი. როდესაც **b=2**) მანტისას თანრიგების რაოდენობა თუ **n** რიცხვის, ხოლო ხარისხის თანრიგების რაოდენობა (ნიშნადი თანრიგების ჩაუთვლელად) **r** რიცხვის ტოლია, მაშინ მცურავი წერტილიანი (მძიმიანი) რიცხვების დიაპაზონი იქნება:

$$2^{-n} \cdot 2^{-k} \le N \le (1 - 2^{-n})\, 2^{k} \qquad (2.32)$$

სადაც $k = (2^r - 1)$.

რიცხვების წარმოდგენის ნორმალური ფორმა თანამედროვე კომპიუტერებში რიცხვების წარმოდგენის ძირითად ფორმად ითვლება.

5

უნიშნო მთელი რიცხვების წარმოდგენა კომპიუტერში. მთელი რიცხვები კომპიუტერში შეიძლება წარმოდგენილი იყოს უნიშნოდ ან ნიშნით. **უნიშნო მთელი რიცხვები** კომპიუტერის მეხსიერებაში ჩვეულებრივ ერთი ან ორი ბაიტის ტოლ ადგილს იკავებს (**ცხრ.2.7**). რიცხვების მნიშვნელობები ერთბაიტიანი ფორმატის დროს $00000000_{(2)}$–დან $11111111_{(2)}$–მდე, ხოლო ორბაიტიანი ფორმატის დროს 00000000 00000000-დან 11111111 11111111-მდე იცვლება. განვიხილოთ მაგალითები

ა) რიცხვ $72_{(10)} = 1001000_{(2)}$-ს ერთბაიტურ ფორმაში ექნება სახე:

7	6	5	4	3	2	1	0	← თანრიგების ნომრები
0	1	0	0	1	0	0	0	← რიცხვის ბიტები

ბ) იგივე რიცხვს ორბაიტურ ფორმაში ექნება სახე:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	← თანრიგების ნომრები
0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	← რიცხვის ბიტები

გ) რიცხვ **65535**-ს ორბაიტურ ფორმაში ექნება სახე:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	← **თანრიგების ნომრები**
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	← **რიცხვის ბიტები**

6 **ნიშნიანი მთელი რიცხვების წარმოდგენა კომპიუტერში.** კომპიუტერის მეხსიერებაში ნიშნიანი მთელი რიცხვები ჩვეულებრივ იკავებენ ერთი, ორი ან ოთხი ბაიტის ტოლ ადგილს (ცხრ.2.8)

ცხრილი 2.8. ნიშნიანი მთელი რიცხვების მნიშვნელობათა დიაპაზონები

ბაიტების რაოდენობა	დ ი ა პ ა ზ ო ნ ი	
	ჩანაწერი ხარისხით	ჩვეულებრივი ჩანაწერი
1	-2^{7} ••• $2^{7}-1$	0 ••• 65535
2	-2^{15} ••• $2^{15}-1$	-32768 ••• 32767
4	-2^{31} ••• $2^{131}-1$	-2147483648 ••• 2147483647

ციფრულ კომპიუტერებში ნიშნიანი მთელი რიცხვების ჩაწერის შემდეგი სამი ფორმა გამოიყენება:

- ნიშნიანი მთელი რიცხვების წარმოდგენა პირდაპირი კოდით;
- ნიშნიანი მთელი რიცხვების წარმოდგენა დამატებითი კოდით;
- ნიშნიანი მთელი რიცხვების წარმოდგენა შებრუნებული კოდით;

მოკლედ განვიხილოთ თითოეული მათგანი

7 **ნიშნიანი მთელი რიცხვების წარმოდგენა პირდაპირი კოდით.** პირდაპირი კოდის სიტყვები გარეგნულად ორობითი კოდის სიტყვების იდენტურებია და ისინი ზოგადი სახით ჩაიწერება როგორც $a_{n-1}a_{n-2}\dots a_1\,a_0$. განსხვავება ის არის, რომ **ორობითი კოდის** შემთხვევაში უმაღლეს თანრიგში მდგარი $a_{n-1}\in\{0;1\}$ ციფრი შედის კოდური სიტყვის გამომხატველი რიცხვის შემადგენლობაში, ხოლო **პირდაპირი კოდის** შემთხვევაში იგი არ ეკუთვნის აღნიშნულ რიცხვს და გამოხატავს დანარჩენ თან-რიგებში

ა)

რიცხვი $1_{(10)} = 1_{(2)}$

0	0	0	0	0	0	0	1

რიცხვის ნიშანი "+"

რიცხვი $127_{(10)} = 1\,1\,1\,1\,1\,1\,1_{(2)}$

0	1	1	1	1	1	1	1

რიცხვის ნიშანი "+"

ბ)

-1 რიცხვის პირდაპირი კოდი

1	0	0	0	0	0	0	1

რიცხვის ნიშანი "-"

-127 რიცხვის პირდაპირი კოდი

0	1	1	1	1	1	1	1

რიცხვის ნიშანი "-"

ნახ. 2.13. დადებითი (ა) და უარყოფითი (ბ) რიცხვების წარმოდგენა პირდაპირი კოდების სახით

მდგარი ციფრებისაგან წარმოქმნილი $_{n-1}a_{n-2} \dots a_1\, a_0$ რიცხვის ნიშანს; კერძოდ, თუ a_{n-1} = **0**, მაშინ $a_{n-1}a_{n-2} \dots a_1\, a_0$ რიცხვი დადებითი ნიშნისაა, ხოლო თუ a_n = **1**, მაშინ $a_{n-1}a_{n-2} \dots a_1\, a_0$ წარმოადგენს უარყოფით რიცხვს (ნახ.**2.13**). აღნიშნულიდან გამომდინარე n თანრიგს ეწოდება ნიშნური თანრიგი. ნიშნურ თანრიგზე მდგარ ციფრს თუ აღვნიშნავთ $a_{ნშ}$ სიმბოლოთი, მაშინ პირდაპირი კოდის კოდურ სიტყვას ექნება სახე: $a_{ნშ}a_{n-2} \dots a_1\, a_0$.

პირდაპირი კოდისათვის სამართლიანია გამოსახულება:

$$A_{(10)} = (-1)^{a_{ნშ}} \sum_{i=0}^{n-2} a_i 2^i \qquad (2.33)$$

სადაც n არის კოდის თანრიგების რაოდენობა; $a_{ნშ}$ - ნიშნის თანრიგის მნიშვნელობა.

მაგალითად, **1011** სახის პირდაპირ კოდს თვლის ათობით სისტემაში ექნება სახე:

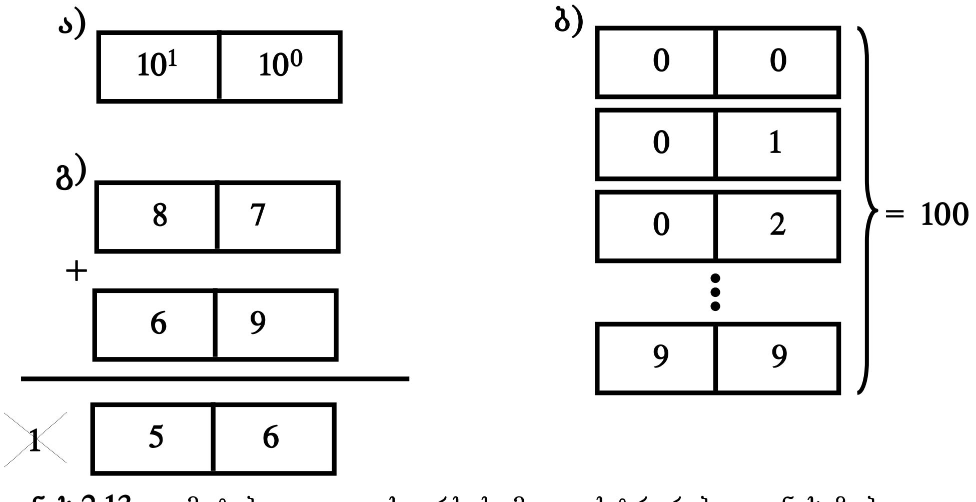
$$A_{(10)} = (-1)^1 [\,0\;\; 2^2 + 1\;\; 2^1 + 1\;\; 2^0\,] = -3$$

8 პირდაპირი კოდების საშუალებით რიცხვების წარმოდგენის დროს გამოიყენება დადებითი და უარყოფითი რიცხვები, რაც ართულებს კომპიუტერის სტუქტურას. ასეთ შემთხვევაში **სხვადასხვა ნიშნებიანი** ორი რიცხვის შეკრების ოპერაციის შესასრულებლად შემდეგი სამი ქვეოპერაციის შესრულებაა საჭირო:

- **ქვეოპერაცია I.** შესაკრები რიცხვებიდან განისაზღვროს უდიდესი აბსოლუტური მნიშვნელობის (მოდულის) მქონე რიცხვი;
- **ქვეოპერაცია II.** დიდი აბსოლუტური მნიშვნელობის რიცხვს გამოაკლდეს მეორე რიცხვი;
- **ქვეოპერაცია III.** მიღებულ შედეგს მიენიჭოს უდიდესი აბსოლუტური მნიშვნელობის (მოდულის) მქონე რიცხვის ნიშანი.

ერთი ოპერაციის სამ ქვეოპერაციად დაშლის თავიდან ასაცილებლად კომპიუტერში უარყოფითი რიცხვების წარმოდგენა ხდება **დამატებითი ან შებრუნებული კოდების** გამოყენებით. ეს საშუალებას გვაძლევს სხვადასხვა ნიშნებიანი ორი რიცხვის შეკრება შევცვალოთ დადებითი რიცხვების შეკრებით, რაც არითმეტიკულ-ლოგიკური მოწყობილობის (იხ. ნახ.**1.10**) სტრუქტურის გამარტივების საშუალებას იძლევა.

9 პირდაპირი კოდიდან დამატებით კოდებში რიცხვების გადაყვანის არსი ავხსნათ სასკოლო არითმეტიკიდან კარგად ცნობილი ათობითი რიცხვების მაგალითზე. ამ მიზნით განვიხილოთ ვირტუალური **n =2** თანრიგიანი კომპიუტერი, რომელიც ოპერირებს ათობით რიცხვებზე. ასეთი კომპიუტერის რეგისტრის თანრიგობრივი ბადე **2.13,ა** ნახაზზეა ნაჩვენები. მასში შეიძლება ჩაიწეროს რიცხვები **00; 01; 02; ...; 99**. აღნიშნული რიცხვების რაოდენობა $b^n = 10^2 = 100$-ის ტოლია (**ნახ.2.13,ბ**).



ნახ.2.13. დამატებითი კოდის არსის მაილუსტრირებელი ნახაზები

დავუშვათ, რომ ჩვენმა ვირტუალურმა კომპიუტერმა უნდა შეკრიბოს სხვადასხვა ნიშნიანი $\mathbf{A_{(10)1} = 87}$ და $\mathbf{A_{(10)2} = -31}$ რიცხვები, ე.ი. შეასრულოს ოპერაცია:

$$A_{(10)1} + A_{(10)2} = 87+(-31). \qquad (2.34)$$

უარყოფითი ათობითი $\mathbf{A_{(10)2}}$ შესაკრების მნიშვნელობა შევცვალოთ რიცხვით, რომელიც უნდა დავუმატოთ ამ შესაკრების აბსოლუტურ $\mathbf{|A_{(10)2}| = 31}$ მნიშვნელობას, რათა მიღებული იქნეს $\mathbf{b^n = 10^2 = \ = 100}$ რიცხვი. მიღებულ შედეგს $\mathbf{A_{(10)2}}$ რიცხვის **100-მდე დამატება** ან უბრალოდ **დამატება** ვუწოდოთ და აღვნიშნოთ $|A_{(10)2}|_{\text{დამ.}}$ სიმბოლოთი. ცხადია, რომ:

$$|A_{(10)2}|_{\text{დამ.}} = 100 - 31 = 100 + A_{(10)2} = 69. \qquad (2.35)$$

(**2.34**)- გამოსახულებაში $\mathbf{A_{(10)2}}$ რიცხვი შევცვალოთ მისი $|A_{(10)2}|_{\text{დამ.}}$ დამატებით და ჩვენი ვირტუალური კომპიუტერის საშუალებით შევასრულოთ ოპერაცია:

$$A_{(10)1} + |A_{(10)2}|_{\text{დამ.}} = 87 + 69. \qquad (2.36)$$

კომპიუტერის მიერ აღნიშნული ოპერაციის შესრულების შედეგი **2.13,გ** ნახზზეა ნაჩვენები. როგორც ნახაზიდან ჩანს, შეკრების შედეგად მიიღება სამნიშნა რიცხვი **156;** ვინაიდან ჩვენი ვირტუალური ათობითი კომპიუტერი ორთანრიგიანია, მიღებული რიცხვის უმაღლეს თანრიგზე მდგარი რიცხვი **1** ვერ მოთავსდება კომპიუტერის თანრიგობრივ ბადეში და დაიკარგება, რაც მიღებული შედეგისათვის რიცხვ **100**-ის დაკლების ტოლფასია.

(**2.35**) გამოსახულების თანახმად, უარყოფით შესაკრებს დაემატა **100,** ხოლო კომპიუტერში შესრულებული მანიპულაციების საშუალებით მიღებულ შედეგს დააკლდა იგივე რიცხვი **100.** ე.ი შესრულდა ოპერაცია:

$$\begin{aligned} A_{(10)1} + |A_{(10)2}|_{\text{დამ.}} - 100 &= A_{(10)1} + |A_{(10)2} + 100| - 100 = \\ &= 87 + |-31 + 100| - 100 = 87 + 69 - 100 = 56. \qquad (2.37) \end{aligned}$$

აქედან გამომდინარე, კომპიუტერის რეგისტრში დარჩენილი რიცხვი **56** წარმოადგენს (**2.34**) ოპერაციის შესრულებისას მისაღებ შედეგს. **100**-ის გამოკლების ოპერაციის არსია ის, რომ არ ხდება გათვალისწინება მესამე ათობითი თანრიგის კოდისა.

განხილული მაგალითიდან ჩანს, რომ დამატებითი კოდების შემოღებით შესაძლებელია სხვადასხვა ნიშნიანი რიცხვების შეკრების ოპერაცია დავიყვანოთ დადებითი რიცხვების შეკრების ოპერაციამდე და თავიდან ავიცილოთ ერთი ოპერაციის სამ ქვეოპერაციად დაშლის აუცილებლობა.

10 ამგვარად, თვლის **b** ფუძის მქონე სისტემით წარმოდგენილი **n**-თანრიგიანი მთელი **K** რიცხვის **დამატება** ეწოდება სხვაობას:

$$M = b^n - |K|. \qquad (2.38)$$

რადგან სხვადასხვა მთელი დადებითი **n**-თანრიგიანი რიცხვების საერთო რაოდენობა $\mathbf{b^n}$-ს ტოლია [(**0**-დან ($\mathbf{b^n}$**-1**)-მდე], ამიტომ ამ რიცხვებიდან ნახევარი [(**0**-დან $\frac{\mathbf{b}^n}{2} - 1$)-მდე] განიხილება როგორც დადებითი, ხოლო მეორე ნახევარი – როგორც უარყოფითი (ანუ როგორც პირველ ნახევარში შესული რიცხვების **დამატებები**). ასეთი რიცხვების სრულ ნაკრებს ეწოდება **დამატებით კოდით წარმოდგენილი რიცხვები** (ცხრ.**2.9**). **[6; 10]**

დამატებების (უარყოფითი რიცხვების კოდების) მიღების სისწორის შესამოწმებლად **0** კოდის ზემოთ და ქვემოთ სიმეტრიულად განთავსებული რიცხვები უნდა შევკრიბოთ. მაგალითად:

ა) **5**-თანრიგიანი ათობითი რიცხვებისათვის:

99999 + 00001 = 1 00000;
99998 + 00002 = 1 00000;

⋮

50001 + 49999 = 1 00000

⋮

ბ) 4-თანრიგიანი თექვსმეტობითი რიცხვებისათვის:

FFFF + 0001 = 1 0000

FFFE + 00000000 –

⋮

ცხრილი 2.9. რიცხვების დამატებითი კოდები

5-თანრიგიანი ათობითი რიცხვების პირდაპირი კოდი	დამატებითი კოდები:		
	5-თანრიგიანი ათობითი რიცხვების	4-თანრიგიანი თექვსმეტობითი რიცხვების	16-თანრიგიანი ორობითი რიცხვების
-50 000	50 000	—	—
-49 999	50 001	—	—
-49 998	50 002	—	—
•••	•••	•••	•••
-32 769	67 231	—	—
-32 768	67 232	8000	1 000 0000 0000 0000
-32 767	67 233	8001	1 000 0000 0000 0001
•••	•••	•••	•••
-3	99 997	FFFD	1 111 1111 1111 1101
-2	99 998	FFFE	1 111 1111 1111 1110
-1	99 999	FFFF	1 111 1111 1111 1111
0	00 000	0000	0 000 0000 0000 0000
+1	00 001	0001	0 000 0000 0000 0001
+2	00 002	0002	0 000 0000 0000 0010
+3	00 003	0003	0 000 0000 0000 0011
•••	•••	•••	•••
+32 766	32 766	7FFE	0 111 1111 1111 1110
+32 767	32 767	7FFF	0 111 1111 1111 1111
+32 768	32 768	—	—
•••	•••	•••	•••
+49 998	49 998	—	—
+49 999	49 999	—	—

უარყოფითი რიცხვები

დადებითი რიცხვები

გ) **16**-თანრიგიანი ორობითი რიცხვებისათვის:

$$\mathbf{1\ 111\ 1111\ 1111\ 1111 + 0\ 000\ 0000\ 0000\ 0001 = 1\ 0\ 000\ 0000\ 0000\ 0000}$$
$$\mathbf{1\ 111\ 1111\ 1111\ 1110 + 0\ 000\ 0000\ 0000\ 0010 = 1\ 0\ 000\ 0000\ 0000\ 0000}$$
$$\vdots$$

მაგალითებიდან ჩანს, რომ ასეთი შეკრების შედეგად მიიღება რიცხვები:

$$\mathbf{10^5} = (1\ 00\ 0000)_{(10)},$$
$$\mathbf{16^4} = (1\ 0000)_{(16)},$$
$$\mathbf{2^{16}} = (1\ 0\ 000\ 0000\ 0000\ 0000)_{(2)}.$$

რიცხვების თანრიგიანობა რადგან შესაბამისად **5; 4** და **16** რაოდენობის თანრიგებითაა შეზღუდული, ამიტომ მიღებული ჯამების უფროს თანრიგებში არსებული ერთიანები არ გაითვალისწინება, ვინაიდან ისინი “ცდებიან” თანრიგობრივ ბადეს. აღნიშნულიდან გამომდინარე, თითოეული ჯამი ნულის ტოლია, ე.ი. სიდიდით ერთნაირი და ნიშნებით განსხვავებული რიცხვების შეკრების შედეგად მიიღება ნული, რაც ადასტურებს უარყოფითი რიცხვებისათვის კოდების მიღების სისწორეს.

10 რიცხვის დამატება შეიძლება მივიღოთ ზემოთ მოყვანილი გამოკლების ოპერაციის შეუსრულებლად. ამის საჩვენებლად (**2.38**) გამოსახულება ასე გადავწეროთ:

$$\mathbf{M = b^n - |K| = [(b^n - 1) - |K|] + 1.} \qquad (2.39)$$

ამ გამოსახულებაში არსებული $(\mathbf{b^n - 1})$ რიცხვი შედგება უმაღლეს (**p-1**) თანრიგში მდგარი **n** რაოდენობის ციფრებისაგან; სხვა სიტყვებით რომ ვთქვათ $(\mathbf{b^n - 1})$ რიცხვი წარმოადგენს **b** ფუძის მქონე თვლის სისტემაში გამოყენებული ყველაზე მაღალი ციფრისაგან შემდგარ **n**-თანრიგიან რიცხვს. მაგალითად **2.9** ცხრილში განხილულ ათობით, თექვსმეტობით და ორობით სისტემებში გამოყენებული ყველაზე მაღალი ციფრებია შესაბამისად **9; F** და **1;** ამიტომ $(\mathbf{b^n - 1})$ რიცხვს ექნება სახე

- **5**-თანრიგიანი ათობითი რიცხვების შემთხვევაში – $(\mathbf{10^5 - 1}) = \mathbf{99999}$;
- **4**-თანრიგიანი თექვსმეტობითი რიცხვების შემთხვევაში – $(\mathbf{16^4 - 1}) = \mathbf{FFFF}$;
- **16**-თანრიგიანი ორობითი რიცხვების შემთხვევაში –

  $$- (\mathbf{2^{16} - 1}) = \mathbf{1\ 111\ 1111\ 1111\ 1111.}$$

აღნიშნულის გამო $(\mathbf{b^n - 1}) - |\mathbf{K}|$ სხვაობა შეიძლება მივიღოთ (**p-1**) ციფრამდე **K** რიცხვში შემავალი თითოეული ციფრის დამატების (ანუ, შემავსებელი რიცხვის) პოვნის გზით; ამ უკანასკნელს თუ დავუმატებთ **1**-ს, მაშინ მივიღებთ **K** რიცხვის საძებნ დამატებით კოდს.

მაგალითად, დავუშვათ, რომ ვიყენებთ თვლის ათობით სისტემის **2**-თანრიგიან რიცხვებს, მოცემული გვაქვს რიცხვი **-31** და საჭიროა ვიპოვოთ ამ რიცხვის დამატებითი კოდი. ასეთ შემთხვევაში (**2.39**) მიიღებს სახეს:

$$\mathbf{M = 10^2 - |-31| = [(10^2 - 1) - 31] + 1 = [(100 - 1) - 31] + 1 =}$$
$$\mathbf{= [\ 99 - 31\ ] + 1 = 68 + 1 = 69\ ,} \qquad \mathbf{(2.40)}$$

რაც ემთხვევა (**2.35**) გამოსახულებით მიღებულ შედეგს.

11 თვლის ორობით სისტემაში გამოყენებული თითოეული ციფრის **1**-მდე დამატება (**1**-მდე შემავსებელი რიცხვი) უდრის ამ ციფრის ინვერსიას (**0**-ის ინვერსიაა **1**, ხოლო **1**-ის ინვერსია – **0**), ამიტომ (**2.39**) გამოსახულების დახმარებით მეტად მარტივია ორობითი რიცხვის დამატებითი კოდის პოვნის წესი, რომელიც შეიძლე-ბა ასე იქნეს ფორმულირებული:

n-ნიშნა ორობით K რიცხვის შესაბამისი დამატებითი კოდის საპოვნელად საჭიროა:

1) მოვახდინოთ K რიცხვის თითოეული ციფრის ინვერსირება (ე.ი. ციფრები 0 შევცვალოთ ციფრებით 1, ხოლო ციფრები 1 შევცვალოთ ციფრებით 0), რის შედეგადაც მიიღება K რიცხვის ინვერსირებული (შებრუნებული) რიცხვი, რომელიც აღვნიშნოთ $\overline{K}$ სიმბოლოთი.

2) K რიცხვის შესაბამისი დამატებითი კოდის მისაღებად შევასრულოთ ოპერაცია: $\overline{K}$ +1

მაგალითი. ვიპოვოთ 16-თანრიგიანი K = 0 000 0010 1100 0101 რიცხვის დამატებითი კოდი. ვისარგებლოთ ზემოთ ფორმულირებული წესებით:

1) მოცემული K რიცხვის თითოეული ციფრის ინვერსირებით მივიღოთ მოცემული რიცხვის შებრუნებული $\overline{K}$ კოდი:

$$\overline{K} = 1\ 111\ 1101\ 1100\ 1010;$$

K რიცხვის შებრუნებულ $\overline{K}$ კოდზე 1-ის მიმატების გზით მივიღებთ K რიცხვის შესაბამის დამატებით კოდს:

$$\begin{array}{r} 1\ 111\ 1101\ 0011\ 1010 \\ +\qquad\qquad\qquad\qquad 1 \\ \hline 1\ 111\ 1101\ 0011\ 1011 \end{array}$$

მაშასადამე, **K = 1 111 1101 1100 1010** რიცხვის დამატებით კოდს წარმოადგენს $\overline{K}$ +1 = 1 111 1101 0011 1011 რიცხვი. მათი შეკრება გვაძლევს 2^{16} = $(10\ 000\ 0000\ 0000\ 0000)_{(2)}$ რიცხვს. მიღებული შედეგის უდიდეს თანრიგში მდგარი ციფრი 1 ცდება 16-თანრიგობრივ ბადეს, ამიტომ იგი არ გაითვალისწინება და შეკრების შედეგად მიიღება ნულოვანი შედეგი, რაც ადასტურებს ჩატარებული ოპერაციის სისწორეს.

12 ორობითი რიცხვის დამატებითი კოდის უფროსი ბიტი ასრულებს რიცხვის ნიშნის ფუნქციას, ე.ი. დადებითი რიცხვებისათვის იგი 0-ის ტოლი იქნება, ხოლო მათი დამატებებისათვის (უარყოფითი რიცხვებისათვის) – 1-ის ტოლი. ამავე დროს დადებითი რიცხვები დამატებით კოდებში ისევე გამოისახება, როგორც პირდაპირ კოდში – ორობითი რიცხვებით, რომლის ნიშნურ თანრიგში დგას ციფრი 0.

დამატებითი კოდისათვის სამართლიანია თანაფარდობა:

$$A_{(10)} = a_{\text{ნშ}}\,(-2^{n-1}) + \sum_{i=0}^{n-2} a_i^{\text{დ}}\, 2^i, \qquad (2.41)$$

სადაც $a_{\text{ნშ}} = \begin{cases} 0, & \text{დადებითი რიცხვებისათვის} \\ 1, & \text{უარყოფითი რიცხვებისათვის} \end{cases}$; **n** - სამანქანო სიტყვის თანრიგიანობაა,

ხოლო $a_i^{\text{დ}}$ - რიცხვის დამატებითი კოდის i-ური ციფრებია.

მაგალითად, ორობით რიცხვს თუ დამატებით კოდში აქვს სახე 1011, მაშინ ასეთი რიცხვის ათობითი ჩანაწერი იქნება:

$$A_{(10)} = 1 \bullet (-2^3) + [\, 0 \bullet 2^2 + 1 \bullet 2^1 + 1 \bullet 2^0] = -8 + 3 = -5.$$

13 **უარყოფითი ორობითი მთელი რიცხვების წარმოდგენა შებრუნებული კოდით.** უარყოფითი ორობითი რიცხვის შებრუნებული კოდის მისაღებად საჭიროა რიცხვის აბსოლუტური მნიშვნელობის ორობითი კოდის თითოეული ციფრის ინვერტირება მოვახდინოთ; ეს ნიშნავს იმას, რომ აღნიშნულ კოდში არსებული ნულები

შევცვალოთ ერთიანებით, ხოლო ერთიანები - ნულებით. ამ დროს უნდა გვახსოვდეს, რომ უარყოფით რიცხვებზე ჩასატარებელი ყველა ოპერაცია **სამანქანო სიტყვის ფორმატში** სრულდება. ეს ნიშნავს, რომ ორობით რიცხვს მარცხნივ მიეწერება იმდენი ნული, რამდენიც აუცილებელია საჭირო რაოდენობის თანრიგების არსებობის უზრუნველსაყოფად. 8-თანრიგიანი სამანქანო სიტყვების მაგალითები 2.15 ნახაზზეა მოყვანილი.

ა)

რიცხვი: -1;

რიცხვის მოდულის კოდი: 00000001;

რიცხვის შებრუნებული კოდი: 11111110;

1	1	1	1	1	1	1	0

ბ)

რიცხვი: -127;

რიცხვის მოდულის კოდი: 01111111;

რიცხვის შებრუნებული კოდი:10000000;

1	0	0	0	0	0	0	0

ნახ. 2.15. 8-თანრიგიანი სამანქანო სიტყვების მაგალითები

შებრუნებული კოდისათვის სამართლიანია შემდეგი თანაფარდობა:

$$A_{(10)} = a_{ნშ}\,(-2^{n-1}+1) + \sum_{i=0}^{n-2} a_i^{შ}\, 2^i, \qquad (2.42)$$

სადაც $a_{ნშ} = \begin{cases} 0, & \text{დადებითი რიცხვებისათვის} \\ 1, & \text{უარყოფითი რიცხვებისათვის} \end{cases}$; **n** - სამანქანო სიტყვის თანრიგიანობაა,

ხოლო $a_i^{შ}$ - რიცხვის შებრუნებული კოდის i-ური ციფრებია.

მაგალითად, ორობით რიცხვს თუ დამატებით კოდში აქვს სახე 1010, მაშინ ასეთი რიცხვის ათობითი ჩანაწერი იქნება:

$$A_{(10)} = 1 \bullet(-2^3 + 1) + [\, 0 \bullet 2^2 + 1 \bullet 2^1 + 0 \bullet 2^0] = -7 + 2 = -5.$$

დადებითი რიცხვებისათვის $a_{ნშ} = 0$ და შებრუნებულ კოდში რიცხვის წარმოდგენა მთლიანად ემთხვევა პირდაპირ და დამატებით კოდებში ამ რიცხვის წარმოდგენას.

ამგვარად, დადებითი რიცხვები როგორც პირდაპირ, ასევე შებრუნებულ და დამატებით კოდებში ერთნაირად გამოისახება: ნიშნის თანრიგში მათ აქვთ ციფრი 0.

ჩვეულებრივად კომპიუტერში შეყვანის დროს უარყოფითი რიცხვები ავტომატურად გარდაიქმნება შებრუნებულ ან დამატებით ორობით კოდებად; ასეთი სახით ხდება მეხსიერებაში მათი შენახვა, აგრეთვე გადაადგილება და ოპერაციებში მონაწილეობა. კომპიუტერიდან ასეთი რიცხვების გამოტანის დროს ისინი ხელახლა გარდაიქმნება უარყოფით ათობით რიცხვებად.

დადებითი რიცხვებისათვის პირდაპირი, შებრუნებული და დამატებითი კოდები ერთმანეთს ემთხვევა.

2.6. კომპიუტერებში ნამდვილი რიცხვების წარმოდგენის პრობლემა

1

ნამდვილი რიცხვი წარმოადგენს მათემატიკურ აბსტრაქციას, რომელიც წარმოიშვა გარე სამყაროს გეომეტრიული და ფიზიკური სიდიდეების გაზომვის საჭიროებისათვის, აგრეთვე ისეთი ოპერაციების ჩასატარებლად, როგორებიცაა ამოფესვა, ლოგარითმების გამოთვლა და ალგებრული განტოლებების ამოხსნა.

კომპიუტერში ნამდვილი რიცხვების წარმოდგენა აუცილებელია ისეთი ამოცანების გადასაწყვეტად, როგორებიცაა:

- ორბიტიდან მიღებული სურათებით ოკეანეში მოწინააღმდეგის წყალქვეშა ნავის აღმოჩენა;
- ბალისტიკური რაკეტებისა და კოსმოსური აპარატების ტრაექტორიების გამოთვლა და ა.შ.

ჩვენ მიერ ზემოთ განხილული მთელი რიცხვების სიმრავლე უსასრულოა, მაგრამ ყოველთვისაა შესაძლებელი ისეთი რაოდენობის თანრიგების შერჩევა, რომ შესაძლებელი იყოს ნებისმიერი მთელი რიცხვის ზუსტად წარმოდგენა.

ნამდვილი რიცხვების სიმრავლე არა მარტო უსასრულოა, არამედ უწყვეტიცაა; უწყვეტობა ნიშნავს, რომ ნებისმიერ რაგინდ უახლოეს ორ რიცხვს შორის არსებობს უსასრულო რაოდენობის ნამდვილი რიცხვები. ასეთი სიმრავლის დროს რა რაოდენობის თანრიგებიც არ უნდა ავიღოთ, გარკვეული ოპერაციების შესრულების პროცესში ყოველთვის შეიძლება წარმოიშვას ისეთი ნამდვილი რიცხვი, რომლის ზუსტად წარმოდგენა შეუძლებელი იქნება.

აღნიშნულიდან გამომდინარე, ნამდვილი რიცხვების წარმოდგენისათვის საჭიროა ისეთი ფორმა იქნეს შერჩეული, რომ მისაღები იყოს აღნიშნული რიცხვების როგორც სიზუსტე, ასევე დიაპაზონი.

სიზუსტესა და მიღებული მნიშვნელობების დიაპაზონს შორის კომპრომისული გადაწყვეტის ერთ-ერთ შესაძლო ხერხს წარმოადგენს ნამდვილი რიცხვების წარმოსადგენად მცურავი მძიმიანი (წერტილიანი) რიცხვების ფორმის გამოყენება (იხ. გამოსახულება (**2.31**)):

$$\mathbf{A = \pm M\, b^{\pm p}},$$

სადაც **M** არის რიცხვის მანტისა, რომელიც წარმოადგენს აბსოლუტური მნიშვნელობით ერთზე ნაკლებ რიცხვს ($|\mathbf{M}| < 1$); **p** –რიცხვის ხარისხია (**p** მთელი რიცხვია), ხოლო **b** – თვლის სისტემის ფუძე.

პროგრამებში ნამდვილი რიცხვების ჩასაწერად ჩვენთვის კარგად ნაცნობი მძიმის ნაცვლად გამოიყენება წერტილი, ამიტომ მათი ჩაწერის ფორმას **მცურავი წერტილიანი ფორმა ეწოდება.**

მაგალითად, ათობითი **347.38** და ორობითი **1101.01** რიცხვი შეიძლება ასე იქნეს წარმოდგენილი:

$$\mathbf{347 \bullet 38 = 347380 \bullet 10^{-3} = 34738 \bullet 10^{-2} = 34738 \bullet 10^{-1} = 347.38 \bullet 10^{0} = 3.4738 \bullet 10^{1} =}$$

$$\mathbf{= 3.4738 \bullet 10^{2} = 0.34738 \bullet 10^{3} = 0 \;\; 034738 \;\; 10^{4}}$$

$$\mathbf{1101.01 = 110101 \bullet 10^{-10} = 1101.01 \bullet 10^{0} = 11.0101 \;\;\; 10^{10} =}$$

$$\mathbf{= 0.110101 \bullet 10^{100} = 0.0100101 \bullet 10^{101}}$$

ზემოთ განხილული მაგალითებიდან ჩანს, რომ რიცხვის მთელი და წილადური ნაწილის გამყოფი წერტილი $\mathbf{b^{\pm p}}$ სიდიდის ცვლილებაზე დამოკიდებულებით იცვლის თავის ადგილს; ე.ი. მისი განთავსების ადგილი ზუსტად კი არ არის დაფიქსირებული, არამედ ‘‘ცურავს’’; ამიტომ ეწოდება რიცხვის წარმოდგენის ასეთ ფორმას **მცურავი წერტილიანი ფორმა**.

"მცურავი" წერტილი თუ მანტისის პირველი ციფრის წინ არის განთავსებლი, მაშინ მანტისისათვის გამოყოფილი ფიქსირებული რაოდენობის თანრიგების დროს შესაძლებელია რიცხვის მაქსიმალური რაოდენობის ნიშნადი ციფრების ჩაწერა იქნეს უზრუნველყოფილი; ეს ზრდის რიცხვის წარმოდგენის სიზუსტეს. აქედან გამოდის, რომ მანტისა უნდა იყოს სწორი წილადი (**|M| < 1**), რომელშიც წილადური ნაწილის (ე.ი. წერტილის შემდეგ მდგარი) პირველი a ციფრი **განსხვავდება ნულისაგან** (**|M| = 0,*a*…**); რადგან თვლის ორობითი სისტემისათვის ნულისაგან განსხვავებული ციფრი ერთის ტოლია, ე.ი. $a = 1$, ამიტომ ასეთი სისტემის გამოყენების შემთხვევაში **|M| = 0,*1*…**;

სწორი წილადის სახით გამოსახულ რიცხვს, რომელშიც წილადური ნაწილის პირველი ციფრი განსხვავდება ნულისაგან, **ნორმალიზებული რიცხვი** ეწოდება. თვლის ორობით სისტემაში ნორმალიზებული მანტისა ყოველთვის წარმოიდგინება 0,5-დან 1-მდე დიაპაზონში მდებარე ათობითი n რიცხვით, ე.ი სრულდება პირობა

$$0,5 \le n < 1. \qquad (2.43)$$

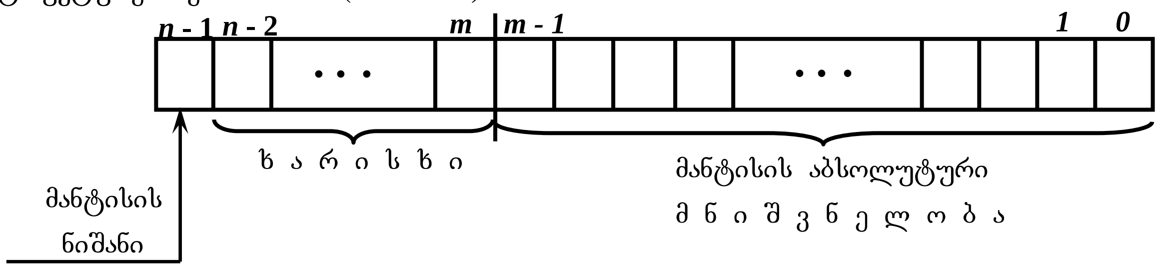
თვლის ათობით და ორობით სისტემებში წარმოდგენილი ნორმალიზებული რიცხვების მაგალითებია:

ათობითი სისტემა	ორობითი სისტემა
$865.28 = 0.86528 \cdot 10^3$	$-101.01 = -0.10101 \cdot 2^{11}$ (ხარისხი $11_{(2)} = 3_{(10)}$)
$-0.000067 = -0.67 \cdot 10^4$	$0.000011 = 0.11 \cdot 2^{-101}$ (ხარისხი $-100_{(2)} = -4_{(10)}$)

კომპიუტერში ნამდვილი რიცხვების წარმოსადგენად, როგორც წესი, გამოიყენება შემდეგი სამი საერთაშორისო სტანდარტული ფორმატი:

- ერთმაგი სიზუსტის ფორმატი;
- ორმაგი სიზუსტის ფორმატი;
- გაფართოებული სიზუსტის ფორმატი.

აღნიშნული ფორმატები რიცხვებს სხვადასხვა სიზუსტით წარმოადგენს, მაგრამ მათი სტრუქტურები ერთნაირია (ნახ. 2.16).



ნახ. 2.16. ნამდვილი რიცხვების წარმოდგენის ფორმატის ზოგადი სტრუქტურა

საჭიროა აღინიშნოს, რომ ნამდვილი რიცხვების წარმოსადგენად გამოყენებული ზემოთ აღნიშნული ფორმატები, რომლებშიც მანტისის შესანახად **m** რაოდენობის თანრიგებია გამოყოფილი, შესაძლებელია გამოვიყენოთ **m**–თანრიგიანი მთელი რიცხვების აბსოლუტურად ზუსტად წარმოდგენისათვისაც. სხვა სიტყვებით რომ ვთქვათ, ნებისმიერი მთელი რიცხვი, რომლის თანრიგების რაოდენობა **m**-ს არ აღემატება, დაუმახინჯებლად შეიძლება გარდაიქმნას ნამდვილ ფორმატად.

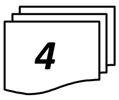
n-თანრიგიანი ნორმალიზებული რიცხვის **ხარისხის** ჩასაწერად გამოიყენება **წანაცვლების მქონე კოდი.** ასეთი კოდი საშუალებას გვაძლევს ხარისხებზე ოპერაციები უნიშნო რიცხვებზე ჩასატარებელი ოპერაციების მსგავსად ჩავატაროთ; ეს ზოგადად ამარტივებს შედარების, შეკრებისა და გამოკლების ოპერაციებს,

მათ შორის თავად ნორმალიზებული რიცხვების შედარების ოპერაციასაც. ასეთი კოდების დიდი მნიშვნელობის გამო მათ ქვემოთ ცალკე განვიხილავთ.

ნიშნის წარმოსადგენად გამოიყენება ერთი ბიტი. აღნიშნული ბიტი თუ 0-ის ტოლია, მაშინ საქმე გვაქვს დადებით ნამდვილ რიცხვთან, ხოლო თუ იგი 1-ის ტოლია, მაშინ – უარყოფით ნამდვილ რიცხვთან;

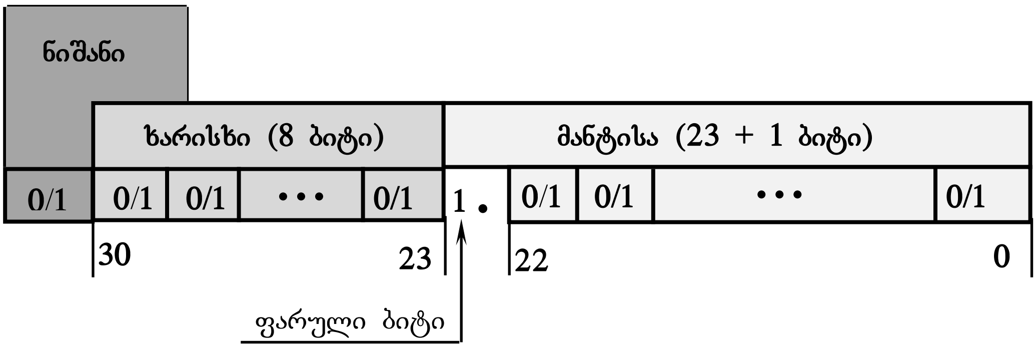
მანტისა, რომელსაც "წილადიც" ეწოდება, გვიჩვენებს წარმოსადგენი ნამდვილი რიცხვის ნიშნად ციფრებს. მანტისის ჩასაწერად ჩვენ მიერ ზემოთ განხილული **პირდაპირი კოდი** შეიძლება გამოვიყენოთ.

რაც უფრო მეტი რაოდენობის თანრიგები გამოიყოფა მანტისის ჩასაწერად, მით უფრო მეტია რიცხვის წარმოდგენის სიზუსტე. რაც უფრო მეტ თანრიგებს შეიცავს ხარისხი, კომპიუტერში გამოყენებული ფორმატის დროს მით უფრო ფართოა რიცხვების წარმოდგენის დიაპაზონი; ეს ნიშნავს, რომ მით უფრო მეტია უმცირეს და უდიდეს რიცხვებს შორის მოთავსებული რიცხვების რაოდენობა.



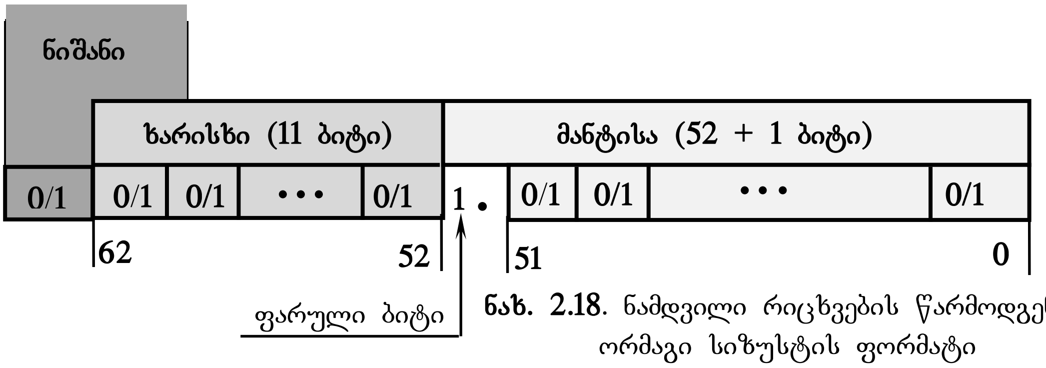
რამდენადმე დაწვრილებით განვიხილოთ ნამდვილი რიცხვების წარმოსადგენად გამოყენებული საერთაშორისო სტანდარტული ფორმატები.

●**ერთმაგი სიზუსტის ფორმატი** გამოიყენება 32-თანრიგიანი ნორმალიზებული ნამდვილი რიცხვების წარმოსადგენად; მისი ზოგადი სტრუქტურული სქემა 2.17 ნახაზზეა ნაჩვენები.



ნახ. 2.17. ნამდვილი რიცხვების წარმოდგენის ერთმაგი სიზუსტის ფორმატი

როგორც ნახაზიდან ჩანს, ერთმაგი სიზუსტის ფორმატის დროს ნიშნის ჩასაწერად გამოიყოფა ერთი, ხოლო ხარისხის ჩასაწერად – 8 (23-დან დაწყებული 30-ის ჩათვლით დამთავრებული) ბიტი. მანტისა თავისებურად ჩაიწერება. ვინაიდან თვლის ორობითი სისტემის გამოყენების დროს მანტისის პირველი ციფრი ყოველთვის 1-ის ტოლია, იგი მეხსიერებაში არ ინახება და **ფარული ბიტი** ეწოდება; მანტისის ჩასაწერად მეხსიერებაში მხოლოდ 23 (0-დან დაწყებული 22-ის ჩათვლით დამთავრებული) ბიტია გამოყოფილი. ამ უკანასკნელს, როგორც 2.16 ნახაზიდან ჩანს, ემატება ფარული ბიტი, და მანტისის ჩასაწერად სულ (23+1)=24 რაოდენობის ბიტი გამოიყენება. მანტისის ასეთი სახით წარმოდგენა რამდენადმე ამაღლებს ნამდვილ რიცხვებზე ჩასატარებელი ოპერაციების შესრულების სიჩქარეს.



ნახ. 2.18. ნამდვილი რიცხვების წარმოდგენის ორმაგი სიზუსტის ფორმატი

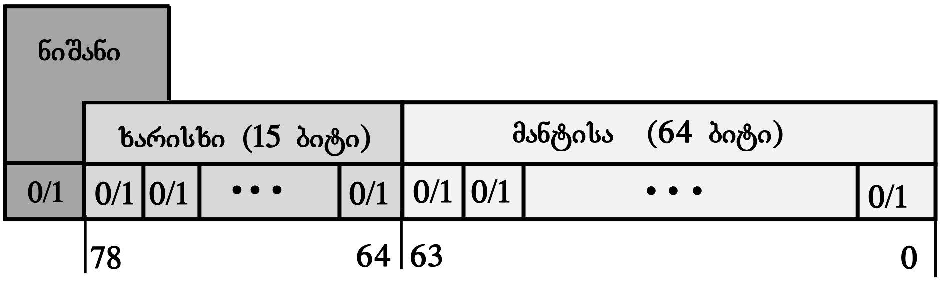
აღნიშნული ფორმატის გამოყენების დროს ნამდვილი რიცხვების წარმოდგენის დიაპაზონია 10^{-42}... 10^{38}.

● **ორმაგი სიზუსტის ფორმატი** გამოიყენება 64-თანრიგიანი ნორმალიზებული ნამდვილი რიცხვების წარმოსადგენად; მისი ზოგადი სტრუქტურული სქემა 2.18 ნახაზზეა ნაჩვენები.

როგორც ნახაზიდან ჩანს, ორმაგი სიზუსტის ფორმატის დროს ნიშნის ჩასაწერად გამოიყოფა ერთი, ხოლო ხარისხის ჩასაწერად – 11 (52-დან დაწყებული 62-ის ჩათვლით დამთავრებული) ბიტი. მანტისის ჩასაწერად მეხსიერებაში 52 (0-დან დაწყებული 51-ის ჩათვლით დამთავრებული) ბიტია გამოყოფილი. მას ემატება ფარული ბიტი და ამის გამო მანტისის ჩასაწერად სულ (52+1)=53 რაოდენობის ბიტი გამოიყენება.

აღნიშნული ფორმატის გამოყენების დროს ნამდვილი რიცხვების წარმოდგენის დიაპაზონია 10^{-324} ... 10^{308}.

● **გაფართოებული სიზუსტის ფორმატი** გამოიყენება 80-თანრიგიანი არანორმალიზებული ნამდვილი რიცხვების შესანახად (**ნახ.2.19**).



ნახ. 2.19. ნამდვილი რიცხვების წარმოდგენის გაფართოებული სიზუსტის ფორმატი

მოცემულ შემთხვევაში ნიშნის ჩასაწერად გამოყოფილია ერთი, ხოლო ხარისხის ჩასაწერად – 15 (64-დან დაწყებული 78-ის ჩათვლით დამთავრებული) ბიტი. **ფარული ბიტი არ გამოიყენება** და მანტისის ჩასაწერად სულ გამოყოფილია მეხსიერებაში არსებული 64 (0-დან დაწყებული 63-ის ჩათვლით დამთავრებული) ბიტი.

5 განვიხილოთ ნამდვილი რიცხვების მცურავი წერტილიანი ფორმით წარმოდგენის მაგალითი.

ათობითი -247,375 რიცხვი წარმოვადგინოთ ნამდვილი რიცხვებისათვის განკუთვნილ ზემოთ განხილულ ფორმატებში. ამისათვის იგი, უპირველეს ყოვლისა, გარდავქმნათ ორობით რიცხვად:

$$(247{,}375)_{(10)} = 11110111.011_{(2)}.$$

მიღებული რიცხვისათვის ჭეშმარიტი ხარისხი $+7_{(10)} = 111_{(2)}$-ის ტოლია. წანაცვლებული კოდის გამოყენების შემთხვევაში ხარისხი შემდეგნაირად იქნება წარმოდგენილი:

- ერთმაგი სიზუსტის ფორმატისათვის: $134_{(10)} = 10000110$;
- ორმაგი სიზუსტის ფორმატისათვის: $1030_{(10)} = 10000000110$;
- გაფართოებული სიზუსტის ფორმატისათვის: $16390_{(10)} = 100000000000110$.

2.20 ნახაზზე განსახილველი რიცხვი წარმოდგენილია ზემოთ განხილული ფორმატების გამარტივებული გამოსახულებებით.

ნორმალიზებული ფორმით შეუძლებელია წარმოდგენილი იყოს **ნიშნიანი ნული;** ასეთი რიცხვის წარმოსადგენად კომპიუტერში დარეზერვებულია მანტისის და ხარისხის სპეციალური მნიშვნელობები (ნახ.2.21); მანტისისა და ხარისხის ასეთივე სპეციალური მნიშვნელობებია დარეზერვებული **ნიშნიანი უსასრულობის** წარმოსადგენადაც (ნახ. 2.22).

	ნიშანი	ხარისხი	ფარული ბიტი	მანტისა
ერთმაგი სიზუსტის	1	10000110	1	111011101100 • • • 00
ორმაგი სიზუსტის	1	10000000110	1	111011101100 • • • 00
გაფართოებული სიზუსტის	1	100000000000110		1111011101100 • • • 00

ნახ.2.20. რიცხვის წარმოდგენა გამარტივებული საერთაშორისო ფორმატებით

ნიშანი	ხარისხი	მანტისა	
0/1	0 0 0 0 0	1. 0 0 0 0 0 0 0 0 0 0	= ± 0

ნახ. 2.21. ნიშნიანი ნულის წარმოდგენა

ნიშანი	ხარისხი	მანტისა	
0/1	1 1 1 1 1	1. 0 0 0 0 0 0 0 0 0 0	= ± ∞

ნახ. 2.22. ნიშნიანი უსასრულობის წარმოდგენა

საინტერესოა აღინიშნოს, რომ კომპიუტერულ მეცნიერებაში ცნობილ მეცნიერ **ფონ ნეიმანს** (რომელსაც ფენომანური მეხსიერების გამო შეეძლო დაემახსოვრებია ყველაფერი, რაც კი ოდესმე გაეგონა, ენახა თუ წაეკითხა [36]) მიაჩნდა, რომ ნებისმიერ მცოდნე მათემატიკოსს შეეძლო რიცხვის წილადური ნაწილის დამახსოვრება და ***1952*** წელს კონსტრუირებულ ***IAS*** კომპიუტერში არ გამოიყენა მცურავი წერტილიანი რიცხვის დამამუშავებელი პროცესორი. ასეთი პროცესორი პირველად ***IBM*** ფირმამ გამოიყენა ***1956*** წელს მის მიერ გამოშვებულ ელექტრონულ მილაკიან ***704***–ტიპის კომპიუტერში.

2.7. ᲬᲐᲜᲐᲪᲕᲚᲔᲑᲣᲚᲘ ᲐᲜᲣ ᲡᲘᲭᲐᲠᲑᲘᲡ ᲛᲥᲝᲜᲔ ᲙᲝᲓᲘ

წანაცვლებული ანუ **სიჭარბის მქონე ორობითი კოდი** გამოიყენება მცურავი წერტილიანი რიცხვების ხარისხის ჩასაწერად. როგორც წინა პარაგრაფში ვაჩვენეთ, მცურავი წერტილის ფორმით რიცხვების წარმოდგენის დროს ხარისხებად შეიძლება გამოყენებული იქნეს როგორც დადებითი ისე უარყოფითი მთელი რიცხვები. აღნიშნულ რიცხვებს თუ შევცვლით წანაცვლებული ორობითი კოდებით, მაშინ ხარისხის მაჩვენებლებად გამოყენებული იქნება მხოლოდ დადებითი ორობითი რიცხვები; ეს ამარტივებს ხარისხებზე შედარების, შეკრებისა და გამოკლების ოპერაციების შესრულებას. გარდა ამისა, წანაცვლებულ კოდებს გააჩნია მეტად საინტერესო შემდეგი თვისება:

A_1 და A_2 რიცხვების ხარისხის მაჩვენებლად გამოყენებული წანაცვლებული $[A_1]_{\text{წ.}}$ და $[A_2]_{\text{წ.}}$ კოდებისათვის თუ სრულდება $[A_1]_{\text{წ.}} < [A_2]_{\text{წ.}}$ უტოლობა, მაშინ $A_1 < A_2$.

ერთმაგი, ორმაგი და გაფართოებული ფორმატით წარმოდგენილი ნამდვილი რიცხვების თანრიგების მაქსიმალური რაოდენობები შესაბამისად **24**-ის, **53**-ისა და **64**-ის ტოლია, ხოლო მათ ხარისხებად გამოიყენება შესაბამისად **8; 11** და **15** თანრიგებიანი წანაცვლებული კოდური სიტყვები. ზემოთ ფორმულირებული თვისების თანახმად **24**-, **53**- და **64**-თანრიგებიანი რიცხვების შედარების ოპერაციები შესაძლებელია დავიყვანოთ შესაბამისად **8-, 11-** და **15-**თანრიგებიანი რიცხვების შედარების ოპერაციებამდე, რაც გაცილებით ადვილი შესასრულებელია.

ზემოთ აღნიშნულიდან გამომდინარე, განვიხილოთ წანაცვლებული, ანუ სიჭარბის მქო-ნე ორობითი კოდის ფორმირების პროცესი.

წანაცვლებული კოდის ფორმირების ზოგადი ალგორითმი ასეთია:

1. შევარჩიოთ თანრიგობრივი ბადის n სიგრძე (n წარმოადგენს რიცხვს რომელიც გვიჩვენებს წანაცვლებული კოდური სიტყვების ჩასაწერად გამოყოფილი თანრიგების რაოდენობას; მაგალითად, კომპიუტერში ნამდვილი რიცხვების ჩასაწერად თუ შერჩეულია საერთაშორისო სტანდარტით გათვალისწინებული ერთამაგი სიზუსტის, ორმაგი სიზუსტის ან გაფართოებული სიზუსტის ფორმატი, მაშინ n-ის მნიშვნელობად უნდა შევარჩიოთ შესაბამისად **8, 11** ან **15** რიცხვები);

2. სვეტის სახით (ერთმანეთის ქვემოთ) კლებადობის ნიშნის მიხედვით ჩამოვწეროთ n–თანრიგიანი ყველა ორობითი რიცხვი;

3. ჩამოწერილი ორობითი რიცხვებიდან ამოვარჩიოთ 2^{n-1} მნიშვნელობის მქონე რიცხვი და მას **ათვლის სათავე** ვუწოდოთ; ასეთია რიცხვი, რომელშიც ციფრი **1** დგას მხოლოდ n-**1** თანრიგში, ხოლო ყველა დანარჩენი თანრიგი შევსებულია ციფრებით **0**. მაგალითად n = **8; 11, 15**-ის დროს ათვლის სათავეებს წარმოადგენს რიცხვები, რომელთა მნიშვნელობებია შესაბამისად: $2^7 = 128$; $2^{10} = 1024$; $2^{14} = 16384$. ასეთი რიცხვებია:

10000000;

10000000000;

100000000000000;

როგორც ჩანს, თითოეული ასეთი რიცხვისთვის ციფრი **1** მხოლოდ (n-**1**)-ე თანრიგშია მოთავსებული, ხოლო დანარჩენი თანრიგები **0**-ებითაა შევსებული.

4. ათვლის სათავედ შერჩეულ n-თანრიგიან ორობით რიცხვს მივანიჭოთ ათობითი მნიშვნელობა **0** და ჩავთვალოთ, რომ მისი საშუალებით წარმოდგენილია ათობითი რიცხვი **0**;

5. ათვლის სათავის ზემოთ მდებარე n-თანრიგიან ორობით რიცხვებს ზრდის კვალობაზე მივანიჭოთ ათობითი მნიშვნელობები **1; 2; 3; ...** და ჩავთვალოთ, რომ მათი საშუალებით წარმოდგენილია შესაბამისად დადებითი ათობითი რიცხვები **1; 2; 3; ...;**

6. ათვლის სათავის ქვემოთ მდებარე n-თანრიგიან ორობით რიცხვებს კლებადობის კვალობაზე მივანიჭოთ ათობითი მნიშვნელობები **-1; -2; -3; ...** და ჩავთვალოთ, რომ მათი საშუალებით წარმოდგენილია შესაბამისად უარყოფითი ათობითი რიცხვები **-1; -2; -3; ...;**

7. n-თანრიგიანი ორობითი რიცხვების ერთობლიობას, რომლებითაც წარმოდგენილია ათობითი **...; 3; 2; 1; 0; -1; -2; -3; ...** რიცხვები, ეწოდება **წანაცვლებული კოდი** ანუ 2^{n-1} **სიჭარბის მქონე ორობითი კოდი**; ეს უკანასკნელი სახელი კოდს იმიტომ ეწოდება, რომ მასში შემავალი თითოეული ორობითი რიცხვის მნიშვნელობა 2^{n-1}-ით აღემატება მის მიერ წარმოდგენილი ათობითი რიცხვის მნიშვნელობას.

8. ალგორითმის დასასრული.

თანრიგების ბადის სიგრძე თუ 3-ის ან 4-ის ტოლია, ამ შემთხვევაში ზემოთ ფორმულირებული ალგორითმით ფორმირდება შესაბამისად $2^{3-1}=4$ და $2^{4-1}=8$ სიჭარბის მქონე ორობითი კოდები. ასეთი კოდების ფორმირების პროცესი შესაბამისად **2.10ა,ბ** ნახაზებზე მოყვანილ ცხრილებშია ილუსტრირებული.

3-თანრიგობრივი ბადის გამოყენების შემთხვევაში წანაცვლების მქონე ფორმატში ათობითი **3; 2; 1; 0; -1; -2; -3; -4** რიცხვები წარმოიდგინება **3**-თანრიგიანი შემდეგი ორობითი რიცხვებით: **111; 110; 101; 100; 011; 010; 001; 000** (ნახ.**2.23,ა**).

4-თანრიგობრივი ბადის გამოყენების შემთხვევაში წანაცვლების მქონე ფორმატში ათობითი **7; 6; 5; 4; 3; 2; 1; 1; 0; -1; -3; -4; -5; -6; -7; -8** რიცხვები წარმოიდგინება **4**-თანრიგიანი ორობითი რიცხვებით: **1111; 1110; 1101; 1100; 1011; 1010; 1001; 1000; 0111; 0110; 0101; 0100; 0011; 0010; 0001; 0000** (ნახ.**2.21,ბ**).

ა)

კოდური კომბინაციის ნომერი	კოდი 4-ის სიჭარბით	ათობითი მნიშვნელობა
7	111	3
6	110	2
5	101	1
4	100	0
3	011	-1
2	010	-2
1	001	-3
0	000	-4

ათვლის სათავე

$\{3;2;1;0;-1;-2;-3;-4\}\ =$

$=\ \{111;110;101;100;011;010;001;000\}.$

ბ)

კოდური კომბინაციის ნომერი	კოდი 8-ის სიჭარბით	ათობითი მნიშვნელობა
15	1111	7
14	1110	6
13	1101	5
12	1100	4
11	1011	3
10	1010	2
9	1001	1
8	1000	0
7	0111	-1
6	0110	-2
5	0101	-3
4	0100	-4
3	0011	-5
2	0010	-6
1	0001	-7
0	0000	-8

$\{7;6;5;4;3;2;1;0;-1;-2;-3;-4;-5;-6;-7;-8\}\ =$

$\{1111;1110;1101;1011;1010;1001;1000;0111;0110;0101;0100;0011;0010;0001;0000\}.$

ნახ.2.23. **4**-ის სიჭარბის (**ა**) და **8**-ის სიჭარბის მქონე (ბ) ორობითი კოდების (წანაცვლებული კოდების) ფორმირების მაილუსტრირებელი ცხრილები

3 3 და -3 რიცხვები წანაცვლების მქონე ფორმატში 3-თანრიგობრივი ბადის შემთხვევაში შესაბამისად იღებს 111 და 001 რიცხვების (ნახ.**2.23,ა**) სახეს, ე.ი. 111=3, ხოლო 001=-3. ჩვეულებრივ, ორობითი კოდის შემთხვევაში 111 ნაკრებს შეესაბამება ათობითი რიცხვი 7, ხოლო 001 კომბინაციას – ათობითი რიცხვი 1, ე.ი. 111=7 და 001=1. მგვარად, ჩვეულებრივ, ორობით კოდსა და წანაცვლებულ კოდში წარმოდგენილ კოდური კომბინაციების მნიშვნელობებს შორის სხვაობები იქნება:

$$7 - 3 = 4 \quad \text{და} \quad 1 - (-3) = 4.$$

რადგან სხვაობები 4-ის ტოლია, ამიტომ წანაცვლებული კოდის თითოეული კოდური კომბინაცია 4-ით ჭარბობს ჩვეულებრივი ორობითი კოდის შესაბამის კომბინაციას; ამის გამო 3-თანრიგიანმა წანაცვლებულმა კოდმა მიიღო **4-ის ტოლი სიჭარბის მქონე კოდის** სახელწოდება.

იგივე 3 და -3 რიცხვები წანაცვლების მქონე ფორმატში 4-თანრიგობრივი ბადის შემთხვევაში იღებს შესაბამისად 1011 და 0101 რიცხვების (ნახ.**2.23,ა**) სახეს, ე.ი. 1011=3, ხოლო 0101=-3. ჩვეულებრივ, ორობითი კოდის შემთხვევაში 1011 ნაკრებს შეესაბამება ათობითი რიცხვი 11, ხოლო 0101 კომბინაციას – ათობითი რიცხვი 5, ე.ი. 1011=7 და 0101=5. ამგვარად, ჩვეულებრივ, ორობით კოდსა და წანაცვლებულ კოდში წარმოდგენილ კოდური კომბინაციების მნიშვნელობებს შორის სხვაობები იქნება:

$$11 - 3 = 8 \quad \text{და} \quad 5 - (-3) = 8.$$

რადგან სხვაობები 8-ის ტოლია, ამიტომ წანაცვლებული კოდის თითოეული კოდური კომბინაცია 8-ით ჭარბობს ჩვეულებრივი ორობითი კოდის შესაბამის კომბინაციას; ამის გამო 4-თანრიგიანმა წანაცვლებულმა კოდმა მიიღო **8-ის ტოლი სიჭარბის მქონე კოდის** სახელწოდება.

ზოგადად n-თანრიგიან წანაცვლებული კოდის სიჭარბე, ჩვეულებრივ, კოდურ კომბინა-ციასთან შედარებით 2^{n-1}–ის ტოლია, ამიტომ მას ზოგადად **სიჭარბის მქონე კოდსაც უწოდებენ.**

მოცემული კოდები გამოიყენება მცურავი მძიმიანი რიცხვების ხარისხების გამოსახვისათვის. ასე, მაგალითად, დავუშვათ რომ ხარისხის მაჩვენებელი რიცხვის განსათავსებლად რიცხვთა ბადეში 8 თანრიგია გამოყოფილი, ე.ი. ხარისხის მაჩვენებლად მხოლოდ n=8 თანრიგიანი ორობითი რიცხვის გამოყენებაა შესაძლებელი. ასეთ შემთხვევაში ხარისხის მაჩვენებლად თუ ჩვეულებრივ ორობით კოდს გამოვიყენებთ, მაშინ ხარისხის მაჩვენებელი ორობითი რიცხვები -128-დან +128-მდე დიაპაზონში იქნება მოთავსებული, ანუ **ხარისხის მაჩვენებლად გამოყენებული იქნება როგორც დადებითი, ასევე უარყოფითი ორობითი რიცხვები.**

მოცემულ შემთხვევაში ხარისხის მაჩვენებლად თუ გამოვიყენებთ $2^{8-1}=2^7=128$-ის ტოლი სიჭარბის მქონე წანაცვლებულ კოდს, მაშინ ხარისხის მაჩვენებელი ორობითი რიცხვები მოთავსებული იქნება 0-დან 255-მდე დიაპაზონში, ანუ **ხარისხის მაჩვენებლად გამოყენებული იქნება მხოლოდ დადებითი ორობითი რიცხვები,** რაც წარმოადგენს ჩვეულებრივ ორობითი კოდის ნაცვლად წანაცვლებული ანუ სიჭარბის მქონე კოდის გამოყენების უპირატესობას.

2.8. ბგერითი ინფორმაციის წარმოდგენა ორობითი კოდებით

1 ბგერა წარმოადგენს საჰაერო არეში გავრცელებულ დრეკად გასწვრივ ტალღას. კომპიუტერის მიერ წაკითხვადი ფორმით მისი წარმოდგენისათვის შემდეგი ოპერაციები სრულდება (ნახ. 2.24):

● მიკროფონის (**მ**-ის) საშუალებით ბგერითი ტალღა გარდაიქმნება ანალოგურ ელექტრულ სიგნალად;

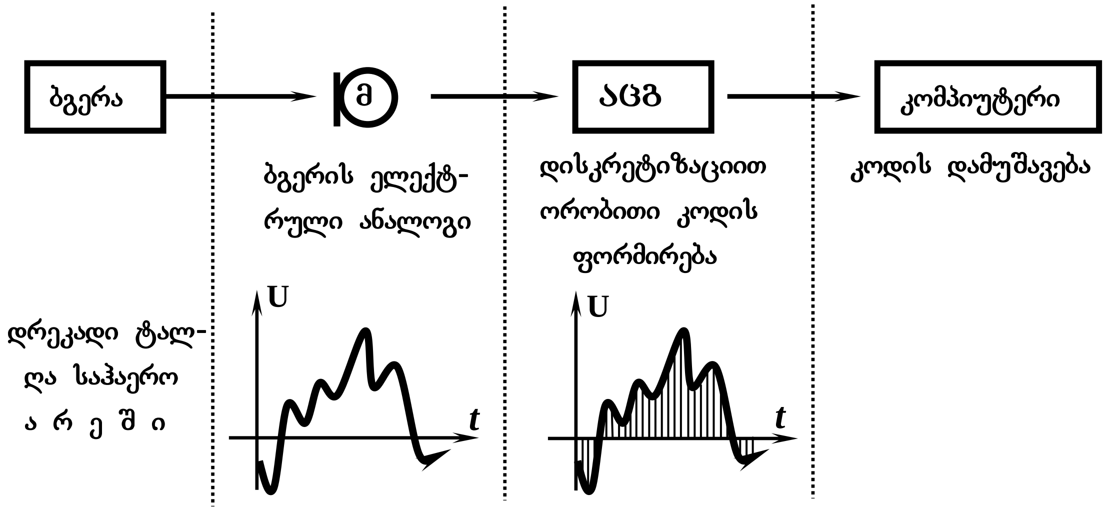
● მიკროფონის დახმარებით მიღებული ბგერის ანალოგური ელექტრული სიგნალი **დისკრეტიზაციის, დაკვანტვისა და კოდირების** ოპერაციათა მეშვეობით წარმოიდგინება ორობითი კოდების სახით;

● მიღებული ორობითი კოდები მიეწოდება კომპიუტერს, სადაც მოხდება მისი დამახსოვრება (მეხსიერებაში ჩაწერა) და, საჭიროების შემთხვევაში, სათანადოდ დამუშავება.

2

ბგერის ანალოგურ ელექტრულ სიგნალს ორობით კოდებად გარდაქმნის ე.წ. ანალოგურ-ციფრული გარდამქმნელი, ანუ შემოკლებით **აცგ** (იხ.ნახ.2.24).

ანალოგურ-ციფრული გარდამქმნელი (ინგლ. ***Analog-to-digital converter; ADC***) ეწოდება დისკრეტულ კოდად შემავალი ანალოგური სიგნალის გარდამქმნელ მოწყობილობას. **აცგ,** როგორც წესი, წარმოადგენს ორობით ციფრულ კოდად ძაბვის გარდამქმნელ ელექტრონულ მოწყობილობას. უკუგარდაქმნა ხორციელდება **ციფრულ-ანალოგური გარდამქმნელით** (ინგლ. ***Digital-to-analog converter; DAC***), ანუ შემოკლებით **ცაგ**-ით.



ნახ. 2.24. ბგერითი სიგნალის დამუშავების სქემა

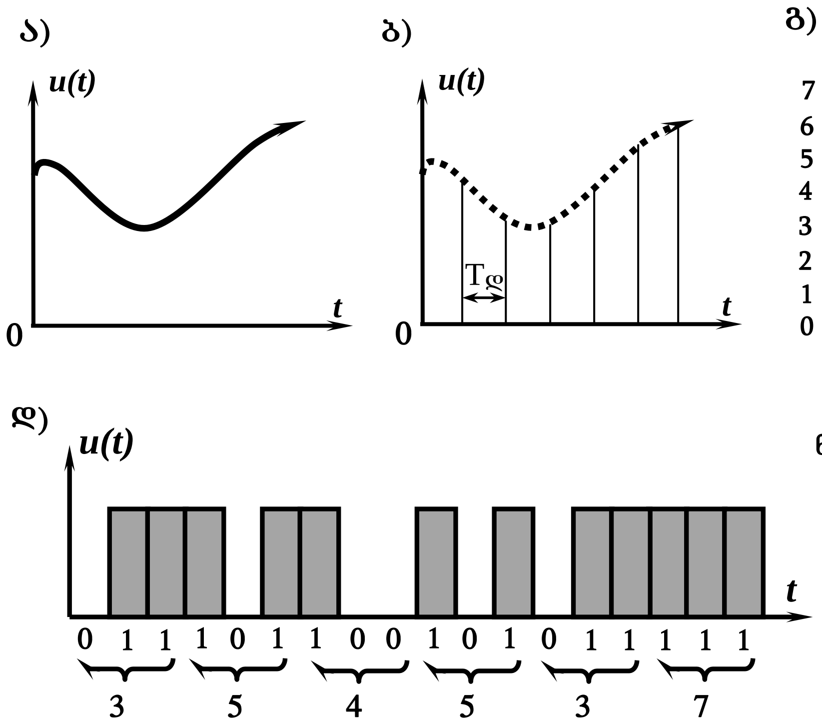
ზოგადად ანალოგურ-ციფრული გარდაქმნა შედგება შემდეგი სამი ეტაპისაგან:

● დისკრეტიზაცია;

● დონის მიხედვით დაკვანტვა;

● კოდირება.

დისკრეტიზაცია ეწოდება უწყვეტი დროის ფუნქციის გარდაქმნას დისკრეტული დროის ფუნქციად, ხოლო **დისკრეტიზაციის პროცესი** წარმოადგენს უწყვეტი ფუნქციის შეცვლას ამ ფუნქციის მიერ დროის ფიქსირებულ მომენტებში მიღებული ცალკეული მნიშვნელობების ერთობლიობით.



ნახ. 2.25. ანალოგურ-ციფრული გარდაქმნის პროცესი

უწყვეტი ფუნქციის ცალკეული მნიშვნელობების აღების მომენტებს შორის არსებულ დროით ინტერვალს **დისკრეტიზაციის პერიოდი** ეწოდება და აღინიშნება $\mathbf{T_დ}$ სიმბოლოთი. დისკრეტიზაციის პროცესის დროს აღნიშნული პერიოდის სიდიდე შეიძლება იყოს მუდმივი (ე.ი. $\mathbf{T_დ}$=const) ან ცვლადი (ე.ი. $\mathbf{T_დ}$=var). პირველ შემთხვევაში საქმე გვაქვს **თანაბარზომიერ დისკრეტიზაციასთან**, ხოლო მეორე შემთხვევაში – **არათანაბარზომიერ დისკრეტიზაციასთან.** სიმარტივისათვის ჩვენ მხოლოდ **თანაბარზომიერ დისკრეტიზაციას** განვიხილავთ.

უწყვეტი $\boldsymbol{u(t)}$ სიგნალის დისკრეტიზაციის $\mathbf{T_დ}$ პერიოდი (იხ. ნახ.2.23,ბ) **კოტელნიკოვის თეორემის** შესაბამისად შეირჩევა და განისაზღვრება ფორმულით:

$$\mathbf{T_დ} = \frac{\mathbf{1}}{\mathbf{2F_{მღ}}}, \qquad (2.44)$$

სადაც $\mathbf{F_{მღ}}$ არის $\boldsymbol{u(t)}$ სიგნალის სპექტრში არსებული უმაღლესი სიხშირე.

სიგნალის მნიშვნელობათა მთელ დიაპაზონს **სკალა** ეწოდება, რომელიც შეიძლება იყოს უწყვეტი ან დისკრეტული (ანუ წყვეტილი).

უწყვეტი სკალის მქონე რაიმე სიგნალის გარდაქმნას დისკრეტული სკალის მქონე სიგნალად აღნიშნული **სიგნალის დაკვანტვა** ეწოდება.

უწყვეტი $\boldsymbol{u(t)}$ სიგნალის დაკვანტვისათვის მისი სკალა (ე.ი. მნიშვნელობების მთელი დიაპაზონი) იყოფა $\boldsymbol{h}$-ის ტოლ ნაწილებად (იხ. ნახ.2.25,გ) ანუ **კვანტებად**; $\boldsymbol{h}$-ს უწოდებენ დაკვანტვის ბიჯს.

კვანტი (ლათ. “**quantum**” – “რამდენი”) ფიზიკური ტერმინია და რაიმე სიდიდის განუყოფად ნაწილს ეწოდება.

დაკვანტვის პროცესის დროს სიგნალის ნებისმიერი მყისი მნიშვნელობა იცვლება ნებადართული მნიშვნელობების სასრულ სიმრავლეში შემავალი მნიშვნელობებიდან ერთ-

ერთი მნიშვნელობით. აღნიშნულ სიმრავლეში შემავალ მნიშვნელობებს **დაკვანტვის დონეები** ეწოდება.

დისკრეტიზაციისა და დაკვანტვის შემდეგ ***u(t)*** სიგნალი იღებს **2.25,გ** ნახაზზე მოცემულ სახეს.

u(t) სიგნალის დისკრეტიზირებული მნიშვნელობა, რომელიც დაკვანტვის ორ დონეს შორის არის მოქცეული, იგივდება დაკვანტვის უახლოეს დონესთან, ან უახლოეს მაღალ (დაბალ) დონესთან. ეს იწვევს დაკვანტვის შეცდომებს, რომლებიც ყოველთვის დაკვანტვის ბიჯზე ნაკლებია; აქედან გამომდინარე ნათელია, რომ დაკვანტვის ***h*** ბიჯის შემცირებით მცირდება დაკვანტვის შეცდომები, მაგრამ იზრდება დაკვანტვის დონეების რაოდენობა. გარდაქმნის პროცესის დაჩქარების, გამარტივებისა და გაიაფების მიზნით საჭიროა შევარჩიოთ დაკვანტვის მაქსიმალურად დასაშვები ისეთი ბიჯი, რომლის დროსაც წარმოშობილი შეცდომები დასაშვებ ზღვარს არ სცდება.

2.25 ნახაზზე ნაჩვენები მაგალითისათვის დაკვანტვის დონეთა რაოდენობა რვის ტოლია. დონეები შეიძლება დავნომროთ ათობითი რიცხვებით და შემდეგ ისინი გადავიყვანოთ თვლის ორობით სისტემაში. რვა დონისათვის საკმარისია სამი ორობითი თანრიგი. ამ შემთხვევაში სიგნალის თითოეული დისკრეტული მნიშვნელობა გამოისახება ორობითი კოდით. განსახილველი ***u(t)*** სიგნალის დაკვანტვისათვის ანათვლების აღების წერტილებში აღნიშნული სიგნალის მნიშვნელობად თუ ავიღებთ დაკვანტვის უახლოეს მაღალ დონეთა მნიშვნელობებს, მაშინ **2.25,გ** ნახაზიდან გამომდინარე მივიღებთ აღნიშნულ მნიშვნელობათა შემდეგ მიმდევრობას: **654567.** მოცემულ მიმდევრობაში მდგარი ათობით რიცხვები შევცვალოთ მათი შესაბამისი ორობითი რიცხვებით (ცხრ. **2.10**). ამ პროცესს დისკრეტული მნიშვნელობების **კოდირება** ეწოდება. მოცემული მიმდევრობის თითოეული წევრის კოდირების შედეგად მივიღებთ ორობით რიცხვს: **110101100101110111,** რომელიც წარმოადგენს **2.25,ა** ნახაზზე მოცემულ უწყვეტ ***u(t)*** სიგნალის დისკრეტულ (ციფრულ) ფორმას; იგი შეიძლება გარდავქმნათ ორობით სიგნალად (**2.25,დ**) და მივაწოდოთ კომპიუტერს.

ცხრ. 2.10. დონეთა მნიშვნელობების გამოსახვა ორობითი რიცხვებით

დონის მნიშვნელობა	**დონის მნიშვნელობის ორობითი წარმოდგენა**
3	011
5	101
4	100
5	101
3	011
7	111

ამგვარად, ანალოგური სიგნალის დისკრეტიზაციის, დაკვანტვისა და კოდირების შედეგად მიიღება ***n***-თანრიგიანი კოდური სიგნალი, რომლის შემადგენელი ცალკეული ელემენტები ერთმანეთს მისდევენ დისკრეტიზაციის **Tდ** პერიოდის დაცვით. დისკრეტიზაციისა და დაკვანტვის ოპერაციების რაციონალურად შესრულების შემთხვევაში:

- მცირდება ინფორმაციის შენახვასა და დამუშავებაზე საჭირო დანახარჯები;

ამგვარად, ანალოგური სიგნალის დისკრეტიზაციის, დაკვანტვისა და კოდირების შედეგად მიღება ***n***-თანრიგიანი კოდური სიგნალი, რომლის შემადგენელი ცალკეული ელემენტები ერთმანეთს მისდევს დისკრეტიზაციის **Tდ** პერიოდის დაცვით. დისკრეტიზაციისა და დაკვანტვის ოპერაციების რაციონალურად შესრულების შემთხვევაში:

მცირდება ინფორმაციის შენახვასა და დამუშავებაზე საჭირო დანახარჯები;

- მცირდება ინფორმაციის დამუშავების დრო, რაც ჯამურად მნიშვნელოვან ეკონომიკურ ეფექტს გვაძლევს.

მაგალითისათვის განვიხილოთ ციფრულ სისტემებით (მაგალითად, ინტერნეტით) ბგერითი სიგნალის გადაცემის მაგალითი. საერთაშორისო საკონსულტაციო კომიტეტის რეკომენდაციების თანახმად, ტელეფონითა და ტელეგრაფით სატელეფონო შეტყობინებების

გადასაცემად საკმარისია სიხშირის 300-დან 3400 ჰერცამდე ზოლი და 35 დეციბელამდე დინამიკური დიაპაზონი. ამ მონაცემებისათვის ექსპერიმენტულადაა განსაზღვრული მარცვლოვანი განრჩევადობა და იგი შეადგენს 90 %-ს. რადგან რეალურ სატელეფონო არხში სიხშირეთა მოცემული ზოლი ფილტრის საშუალებით გამოიყოფა, რომელსაც სიხშირული მახასიიათებლის სასრული ვარდნა აქვს, ამიტომ სტანდარტული სატელეფონო არხის სპექტრის საანგარიშო სიგანედ იყენებენ 4 კილოჰერცის ტოლ ზოლს.

ასეთი სიგნალისათვის დისკრეტიზაციის $T_დ$ პერიოდი ასე გამოითვლება:

$$T_დ = \frac{1}{f_დ} = \frac{1}{2F_{მდ}} = \frac{1}{2 \text{ x } 4} = \frac{1}{8} = 125 \text{ მიკროწამი.}$$

დისკრეტიზაციის მოცემული პერიოდი და სიხშირე გადაცემის ციფრული სისტემების სტანდარტების დამუშავების საფუძვლად მიიღება.

ანალოგური სიგნალის ამპლიტუდის დაკვანტული მნიშვნელობების კოდირებისათვის, როგორც წესი, გამოიყენება 7- ან 8-თანრიგიანი ორობითი კოდი; დაკვანტვის დონეების N რაოდენობა პირველ შემთხვევაში დაკვანტვის დონეების რაოდენობა $N = 2^7 = 128$-ის, ხოლო მეორე შემთხვევაში $N = 2^8 = 256$-ის ტოლია. ეს უზრუნველყოფს ხარისხობრივი ბგერითი სიგნალის გადაცემას, რომლის ამპლიტუდის მიხედვით D **დინამიკური დიაპაზონი** პირველ შემთხვევაში უდრის

$$D = 2\lg128 = 42 \text{ დეციბელს,}$$

ხოლო მეორე შემთხვევაში –

$$D = 2\lg256 = 48 \text{ დეციბელს.}$$

ასეთი ციფრული ნაკადის **გადაცემის სიჩქარე** პირველ შემთხვევაში უდრის

8 კჰც x 7 ბიტი =56 კბიტი/წმ-ს,

ხოლო მეორე შემთხვევაში -

8 კჰც x 8 ბიტი = = 64 კბიტი/წმ-ს.

2.9. სიმბოლური ინფორმაციის წარმოდგენა ორობითი კოდებით

1

კომპიუტერის ‘‘თვალსაზრისით’’ ტექსტი შედგება ცალკეული სიმბოლოებისაგან. სიმბოლოებს მიეკუთვნება არა მარტო ასოები, არამედ ციფრები, სასვენი ნიშნები, სპეციალური ნიშნები (როგორებიცაა მაგალითად ∃, &, ∀, % და ა.შ.). აღსანიშნავია ის გარემოება, რომ სიგნალად ითვლება სიტყვებს შორის არსებული დაშორებაც (ე.წ. **პრობელი**), ე.ი. ცარიელ ადგილსაც აქვს თავისი აღნიშვნა.

ოპერატიულ მეხსიერებაში ტექსტი შეიძლება შევიტანოთ **კლავიატურის** დახმარებით; კლავიატურის კლავიშებზე დაწერილია ჩვენთვის ნაცნობი ასოები, ციფრები, სასვენი ნიშნები და სხვა სიმბოლოები. ოპერატიულ მეხსიერებაში ისინი ორობითი კოდების საშუალებით ხვდება. ეს ნიშნავს, რომ თითოეული სიმბოლო წარმოიდგინება ორობითი კოდით.

თითოეული სიმბოლოსათვის გარკვეული ორობითი რიცხვის ანუ კოდის შეთანადებას **კოდირება** ეწოდება. შეთანადებისათვის გამოიყენება ორობითი რიცხვების სხვადასხვა ჯგუფები. ერთ-ერთ ყველაზე გავრცელებულ ჯგუფს წარმოადგენს 00000000-დან დაწყებული და 11111111-ით დამთავრებული (ანუ, თვლის ათობითი სისტემის გამოყენების დროს 0-დან დაწყებული და 255-ით დამთავრებული) ორობითი რიცხვები. ამგვარად, **ადამიანი სიმბოლოებს ერთმანეთისაგან მათი მოხაზულობის მეშვეობით განასხვავებს, ხოლო კომპიუტერი – მათთვის მიკუთვნებულ ორობითი კოდებით.**

სიმბოლოების კოდირებისათვის გამოიყენება ე.წ. კოდური ცხრილი. **კოდური ცხრილი ეწოდება** ცხრილს, რომელშიც:

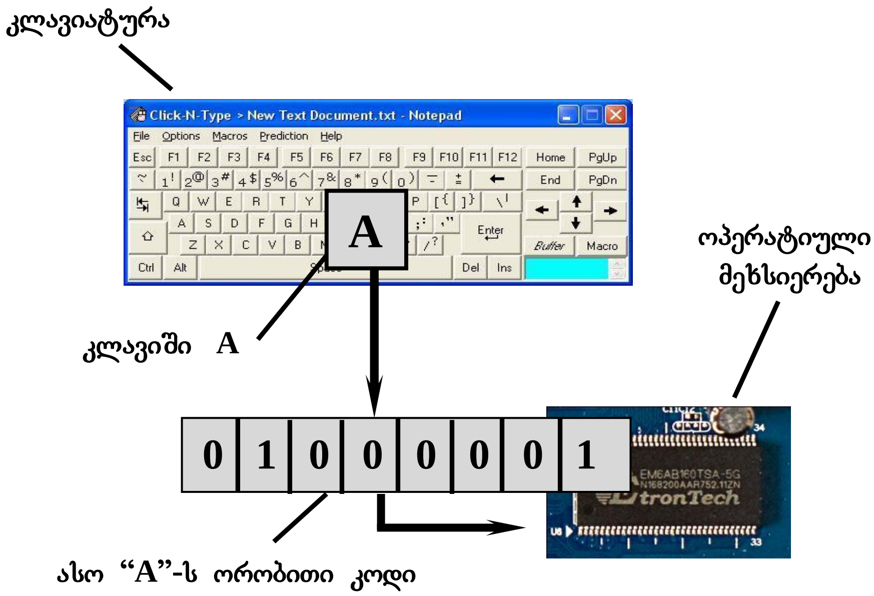
- მოყვანილია კომპიუტერული ალფაბეტისათვის გამოყენებული ყველა სიმბოლო;

● თითოეული სიმბოლოს გვერდით მითითებულია ორობითი რიცხვით გამოსახული მისი რიგითი ნომერი, რომლის მიხედვითაც შედგენილი კოდი (როგორც ეს 2.11 ნახაზზეა ნაჩვენები) გამოიყენება მოცემული სიმბოლოს კოდირებისათვის.

კოდირების გზით სიმბოლოების ორობით რიცხვებად გარდაქმნა იმისათვის არის საჭირო, რომ კომპიუტერული სისტემის ძირითად ელემენტს – **მიკროპროცესორს** – მხოლოდ მათზე შეუძლია გარკვეული ოპერაციების ჩატარება.

2 განვიხილოთ კომპიუტერის ოპერატიულ მეხსიერებაში ლათინური ასო **A**-ს ჩაწერის მაგალითი (ნახ. **2.26**). აღნიშნული ასო კოდირებულია ორობითი კოდური **01000001** სიტყვით. პერსონალური კომპიუტერი ოპერატიულ მეხსიერებაში ასო **A**-ს შესატანად მაგნიტურ მზიდზე აღნიშნული ასოს გამოსახულებას კი არ ქმნის, არამედ სიმბოლოების სპეციალურ **კოდურ ცხრილში** ჩაწერს მის შესაბამის ორობით **01000001** რიცხვს. მიუხედავად ამისა, მონიტორის ეკრანზე ჩვენ ვხედავთ ასო **A**-ს გამოსახულებას. გამარტივებული სახით ეს შეიძლება ასე ავხსნათ.

კომპიუტერის მეხსიერებაში წინასწარაა შენახული ე.წ. **შრიფტის ფაილი,** რომელშიც ორობითი სიტყვების გვერდით მოთავსებულია ამ სიტყვების შესაბამისი სიმბოლოები (ჩვენს შემთხვევაში ორობით რიცხვ **01000001**-ის გვერდით მოთავსებულია სიმბოლო **A**). პროცესორი გაანალიზებს მიღებულ ორობით სიტყვას და მონიტორის ეკრანზე გამოიტანს ამ ორობითი სიტყვის შესაბამის სიმბოლოს (განხილულ შემთხვევაში ასო **A** -ს). ასევე ხდება პრინტერით ტექსტის ბეჭდვის დროსაც, ოღონდ იმ განსხვავებით, რომ ასო **A**-ს გამოსახულება გამოიტანება არა მონიტორის ეკრანზე (ანუ, დისპლეიზე), არამედ ქაღალდზე.



ნახ. 2.26. ოპერატიულ მეხსიერებაში ლათინური ასო **"A"**-ს შესაბამისი ორობითი კოდის შეტანის ილუსტრაცია

3 სიმბოლოების ორობით რიცხვებად გარდაქმნის ოპერაცია კომპიუტერს აძლევს არა მარტო ტექსტის აკრეფის, არამედ ისეთი ოპერაციების მოქნილად და ეფექტურად შესრულების უნარსაც, რომელთა რეალიზება წარმოუდგენელია მოახდინოს ნებისმიერმა საბეჭდმა მანქანამ. კერძოდ, პერსონალურ კომპიუტერს შეუძლია არა მარტო ქაღალდზე დაბეჭდოს ტექსტი, არამედ დაიმახსოვროს ტექსტური მონაცემები, მოახდინოს მათი მოდიფიცირება, მაღალი სიჩქარით გადაუგზავნოს ტექსტი სხვა მომ-

ხმარებელს ან მიიღოს მისგან გამოგზავნილი ტექსტი, იმუშაოს არა მარტო ერთ კონკრეტულ, არამედ (სათანადო პროგრამული უზრუნველყოფის არსებობის შემთხვევაში) მრავალ სხვადასხვა ენაზე შედგენილ ტექსტებზე. უკანასკნელ შემთხვევაში წარმოშობილ ერთადერთ სირთულეს წარმოადგენს ის გარემოება, რომ დიდი რაოდენობის სიმბოლოების შემცველ ენებზე (მაგალითად იაპონური ენის მსგავსი ენები) შედგენილ ტექსტებზე მუშაობისას დაგვჭირდება სიმბოლოების კოდირებისათვის (დასანომრად) გამოვიყენოთ გრძელი (ბევრი რაოდენობის ბიტების შემცველი) ორობითი კოდები.

ცხრ. 2.11. კოდირების **ASCII** ცხრილის სტრუქტურული აგებულება

ათობითი ნომერი	ორობითი ნომერი (კოდი)	კ ო მ ე ნ ტ ა რ ი
0 - 31	00000000 – - 00011111	ცხრილის საბაზისო ნაწილი. მოიცავს მმართველ სიმბოლოებს, რომლებსაც არ შეესაბამება სიმბოლოები. მათი ფუნქციებია მონიტორზე ან დასაბეჭდად ტექსტის გამოტანა, ბგერითი სიგნალის მიწოდება, ტექსტის მონიშვნა და ა.შ..
32 - 127	00100000 – - 01111111	ცხრილის საერთაშორისო სტანდარტული (ინგლისური) ნაწილი; მოიცავს ლათინური ალფაბეტის მთავრულ და ნუსხურ ასოებს, არაბულ ციფრებს, სასვენ ნიშნებს, სხვადასვა სახის ფრჩხილებს, კომერციულსა და სხვა სახის ნიშნებს. 32-ე ნომრის მქონე სიმბოლოა პრობელი, ე.ი. ტექსტში ცარიელი პოზიცია; დანარჩენი სიმბოლოები გარკვეული ნიშნებით გამოისახება.
127 - 255	10000000- -11111111	ცხრილის გაფართოებული (ნაციონალური) ნაწილი; მას კოდური ფურცელი ეწოდება და მოიცავს 128 კოდს (დაწყებულს 10000000-დან და დამთავრებულს 11111111-ით.) არსებობს სხვადასხვა სახის კოდური ფურცლები და თითოეული დანომრილია. გამოიყენება ლათინურისაგან განსხვავებული ნაციონალური ალფაბეტის, ფსევდოგრაფიკისა და სპეციალური ნიშნების განსათავსებლად.

ინგლისურ ენაზე მუშაობისას გამოიყენება 7-ბიტური **კოდური ცხრილი,** რომლის სახელწოდებაა **ASCII** (**A**merican **S**tandad **C**ode for **I**nformation **I**nter-change – "ამერიკული სტანდარტული კოდი ინფორმაციის გასაცვლელად"); **1963** წელს შექმნილი აღნიშნული ცხრილი **128** სიმბოლოს კოდირების საშუალებას იძლევა (რადგან ბიტს მხოლოდ ორი, კერძოდ, "**1**"-ისა და "**0**"-ის ტოლი, მნიშვნელობა შეიძლება ჰქონდეს და თუ ვიყენებთ **7** ბიტს, მაშინ ამ რაოდენობის ბიტებისაგან $2^7 = 128$ ორობითი რიცხვის ფორმირებაა შესაძლებელი). აღნიშნული კოდური ცხრილი აამოქმედა **აშშ**-ს სტანდარტიზაციის **ANSI** (**American National Standard Institut**) ინსტიტუტმა; დღეს იგი ფართოდ გამოიყენება მინი- და მიკროელექტრულ გამომთვლელ მანქანებში, მათ შორის პერსონალურ კომპიუტერებში. აქვე შევნიშნავთ, რომ დიდი გამომთვლელი მანქანებისათვის გამოიყენება **კოდური** **EBCDIC** (**E**xtended **B**inary **C**oded **D**ecimal **I**nterchnge **C**oide –"ინფორმაციის გაცვლის გაფართოებული ორობით-ათობითი კოდი") **ცხრილი**, რომელსაც არ განვიხილავთ.

ცხრ. 2.12. **ASCII** კოდური ცხრილის **საბაზისო ნაწილი**

კ ო დ ი		აღნი-შვნა	კ ო დ ი		აღნი-შვნა	კ ო დ ი		აღნი-შვნა
00000000	0	**NUL**	00001011	11	**VT**	00010101	22	**SYN**
00000001	1	**SOH**	00001100	12	**FF**	000101010	23	**ETB**
00000010	2	**STX**	00001101	13	**CR**	000101010	24	**CAN**
00000011	3	**ETX**	00001110	14	**SO**	000101010	25	**EM**
00000100	4	**EOT**	00001111	15	**SI**	00010100	26	**SUB**
00000101	5	**ENQ**	00010000	16	**DLE**	00010100	27	**ESC**
00000110	6	**ACK**	00010001	17	**DC1**	00010100	28	**FS**
00000111	7	**BEL**	00010010	18	**DC2**	00010100	29	**GS**
00001000	8	**BS**	00010011	19	**DC3**	00010100	30	**RS**
00001001	9	**HT**	00010100	20	**DC4**	00010100	31	**US**
00001010	10	**LF**	00010101	21	**NAK**			

კოდური ASCII ცხრილში ჩამოწერილია **128** სიმბოლო და თითოეული სიმბოლოს გვერდით მითითებულია მისი რიგითი ნომრის გამომხატველი **7**-ბიტური ორობითი რიცხვი. აღნიშნული **7**-ბიტური ორობითი რიცხვის წინ დარეზერვებულია ერთი (მერვე) ბიტი, რომელიც შეიძლება გამოყენებული იქნეს:

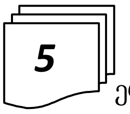
- კოდურ სიტყვებში წყვილი რაოდენობის ერთიანების უზრუნველსაყოფად; ამისათვის, **კენტი** რაოდენობის ერთიანების შემცველი **7**-ბიტური რიცხვის წინ იწერება ბიტი **"1",** ხოლო **ლუწი** რაოდენობის ერთიანების შემცველი **7**-ბიტური რიცხვის წინ – ბიტი **"0";** ეს ამაღლებს კოდების დაცულობას; კერძოდ, კოდური სიტყვის ისეთ დამახინჯებას, რომლის დროსაც მასში შემავალი ერთიანების რაოდენობა კენტი გახდება, კომპიუტერი აღმოაჩენს;

● დანარჩენ 7 ბიტთან ერთად სიმბოლოების დასანომრად; ასეთ შემთხვევაში ორობითი კოდური სიტყვების რაოდენობა იზრდება და ხდება $2^8 = 256$-ის ტოლი; ეს 256 სიმბოლოს კოდირების საშუალებას მოგვცემს.

იაპონურის მსგავს ენებში სიმბოლოების რაოდენობა რამდენიმე ათეულ ათასს აღწევს; ასეთ შემთხვევაში სიმბოლოების დასანომრად გამოიყენება მინიმუმ 16 ბიტი (2 ბაიტის სიგრძის ორობითი რიცხვები, რომლებიც $2^{16} = 65536$ სიმბოლოს კოდირების საშუალებას გვაძლევს, ხილო კოდური ცხრილი 65536 პოზიციისაგან შედგება.

თანამედროვე კომპიუტერები ყველაზე ეფექტურად ამუშავებს 8-ის ჯერადი სიმბოლოებისაგან შემდგარ მონაცემებს. ამას განაპირობებს ის გარემოება, რომ მათი ელექტრონული სქემები სწორედ ასეთი (8, 16, 32, 64 ან 128 და ა.შ.) რაოდენობის ბიტისაგან შემდგარი ორობითი რიცხვების გადაცემა-მიღება-დამუშავებასათვის არის კონსტრუირებული. ამის გამო კოდირების ნებისმიერი სხვა, ვთქვათ 10-ბიტური ან 27-ბიტური, მეთოდი მოუხერხებელი და არაეფექტურია კომპიუტერში გამოსაყენებლად.

მეორე მხრივ, სიმბოლოების კოდირებისათვის (დასანომრად) ძალიან გრძელი ორობითი რიცხვების გამოყენება მნიშვნელოვნად ამცირებს კომპიუტერის მუშაობის ეფექტურობას. მაგალითად, კოდირებისათვის 24 ბიტის ტოლი ორობითი რიცხვების გამოყენება მსოფლიოში არსებული ყველა დამწერლობის ასოების კოდირების საშუალებას მოგვცემდა ($2^{24} = 16777216$), მაგრამ ცალკე აღებული ენების აბსოლიტური უმრავლესობის სიმბოლოების კოდირებისათვის სრულიად საკმარისია 8 ბიტი სიგრძის ორობითი რიცხვის გამოყენება, რადგან მასში არსებული სიმბოლოების რაოდენობა 256-ს არ აღემატება. ამიტომ მათი კოდირებისათვის 24-ბიტური ორობითი რიცხვების გამოყენება კომპიუტერს თითოეულ სიმბოლოზე მუშაობისას იძულებულს გახდიდა დაემუშავებინა სრულიად არასაჭირო 16 ბიტი, რაც გააუარესებდა მის ტექნიკურ მაჩვენებლებს.



კოდირების **ASCII** ცხრილის სტრუქტურა (ცხრ.2.11) შედგება საბაზისო, საერთაშორისო სტანდარტული (ინგლისური) და გაფართოებული (ნაციონალური) ნაწილებისაგან (იხ. ცხრილი 2.11).

საბაზისო ნაწილში (ცხრ. 2.12) განთავსებულია ე.წ. **მმართველი კოდები,** რომლებიც ენის არც ერთ სიმბოლოს არ შეესაბამება; ამის გამო ეს კოდები არ გამოიტანება არც მონიტორის ეკრანზე და არც საბეჭდ მოწყობილობაზე. ისინი მართავს სხვა მონაცემების გამოტანის პროცესს.

2.12 ცხრილში მოცემულია მმართველი ბრძანებების სახელწოდებების აბრევიატურები. მოკლედ განვიხილოთ ეს ბრძანებები.

- **NUL -** **NUL**; “ნული”;
- **SOH -** Start of Heading; “სათაურის დასაწყისი”;
- **STX -** Start of Text; “ტექსტის დასაწყისი”;
- **ETX -** End of Text ”ტექსტის დასასრული”; ამ ბრძანების შესაბამისი **00000011** კოდი გააჩნია Ctrl-C სიმბოლოსაც;
- **EOT -** End of Transmission; “გადაცემის დასასრული”. **UNIX** სისტემაში ანალოგიური კოდი აქვს **Ctrl-D** სიმბოლოსაც;
- **ENQ -** Enquire; “გთხოვთ დადასტურებას”;
- **ACK -** Acknowledgement; “ვადასტურებ”;
- **BEL -** Bell; “ზგერითი სიგნალი” ; გამოიყენება დაპროგრამების **C** და **C++** ენებში და აღინიშნება როგორც **\a**;

- **BS -** Backspace; "ერთი სიმბოლოთი დაბრუნება"; შლის წინა სიმბოლოს;
- **HT -** Horizontal Tabulation; "ჰორიზონტალური ტაბულაცია". დაპროგრამების ბევრ ენაში მას აღნიშნავენ როგორც **\t;**
- **LF -** Line Feed; "სტრიქონის გადაყვანა"; ახლი ტექსტური ფაილის თითოეული სტრიქონის ბოლოში გამოყენებულ ოპერაციულ სისტემაზე დამოკიდე-ბულებით იწერება ან ეს სიმბოლო, ან **CR,** ან ორივე მათგანი (ჯერ **CR**, ხოლო შემდეგ - **LF**);
- **VT -** Vertical Tab; "ვერტიკალური ტაბულაცია";
- **FF -** Form Feed; "ახალი გვერდი";
- **CR -** carriage return "დგიმთამწის დაბრუნება"; დაპროგრამების ბევრ ენაში ამ სიმბოლოს აღნიშნავენ როგორც **\r;** ზოგიერთ ოპერაციულ სისტემაში იგი აღინიშნება როგორც Ctrl-M და იწერება ტექსტური ფაილის თითო-ეული სტრიქონის ბოლოში **LF**–ის წინ;
- **SO -** Shift Out; "შეცვალე ლენტის ფერი". გამოიყენება ინგლისური კოდირები-დან ნაციონალურ კოდირებაზე გადასასვლელად;
- **SI -** Shift In; "დაბალ რეგისტრზე გადართვა";
- **DLE -** Data Link Escape; "მომდევნო სიმბოლოები სპეციალური შინაარსისაა";
- **DC1 -** Device Control 1 "მოწყობილობის მართვის **1**-ლი სიმბოლო";
- **DC2 -** Device Control 2 "მოწყობილობის მართვის მე-**2** სიმბოლო";
- **DC3 -** Device Control 3 "მოწყობილობის მართვის მე-**3** სიმბოლო";
- **DC4 -** Device Control 4 "მოწყობილობის მართვის მე-**4** სიმბოლო";
- **NAK -** Negative AcKnowledgment; "არ ვადასტურებ";
- **SYN -** Synchronization; "სინქრონიზაცია";
- **ETB -** End of Text Block; "ტექსტური ბლოკის დასასრული";
- **CAN -** Cancel; "გაუქმება";
- **EM -** End of Medium; "მზიდის დასასრული";
- **SUB** SUBstitute; "სტრიქონქვეშა ინდექსი";
- **ESC -** ESCape; გამოსასვლელი;
- **FS -** File Separator; "ფაილების დამყოფი (დამაცალკევებელი)";
- **GS -** Group Separator; "ჯგუფების დამყოფი (დამაცალკევებელი)";
- **RS -** Record Separator**;** "ჩანაწერის დამყოფი (დამაცალკევებელი");
- **US -** Unit Separator**;** "მოდულის დამყოფი (დამაცალკევებელი");

ცხრ. 2.13. ASCII კოდური ცხრილის **საერთაშორისო სტანდარტული ნაწილი** (დასაწყისი)

კოდი		სიმბოლო	კოდი		სიმბოლო
00100000	32	პრობელი	00111001	57	9
00100001	33	!	00111010	58	:
00100010	34	"	00111011	59	;
00100011	35	#	00111100	60	<
00100100	36	$	00111101	61	=
00100101	37	%	00111110	62	>
00100110	38	&	00111111	63	?
00100111	39	'	01000000	64	@
00101000	40	(	01000001	65	A
00101001	41	)	01000010	66	B
00101010	42	*	01000011	67	C (ჩ)
00101011	43	+	01000100	68	D
00101100	44	,	01000101	69	E
00101101	45	-	01000110	70	F
00101110	46	.	01000111	71	G
00101111	47	/	01001000	72	H
00110000	48	0	01001001	73	I
00110001	49	1	01001010	74	J (ჟ)
00110010	50	2	01001011	75	K
00110011	51	3	01001100	76	L
00110100	52	4	01001101	77	M
00110101	53	5	01001110	78	N
00110110	54	6	01001111	79	O
00110111	55	7	01010000	80	P
00111000	56	8	01010001	81	Q

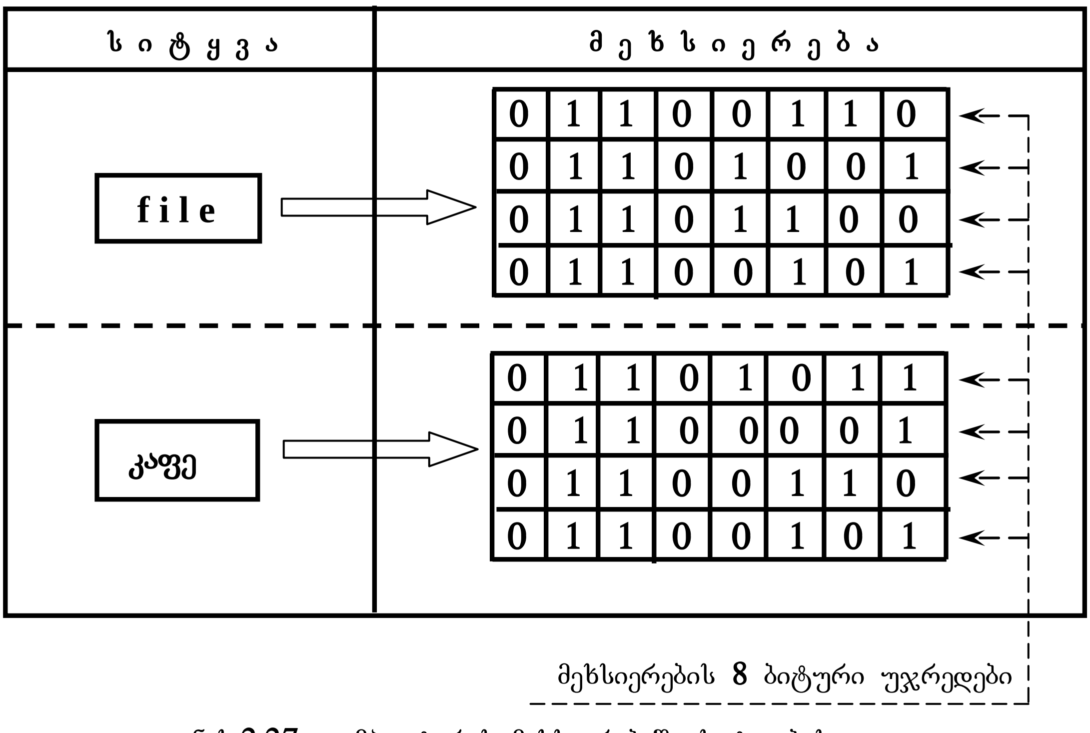
ცხრ. 2.13. ASCII კოდური ცხრილის **საერთაშორისო სტანდარტული ნაწილი** (დასასრული)

კოდი		სიმბოლო	კოდი		სიმბოლო
01010010	82	**R** (ღ)	01101011	107	**k** (კ)
01010011	83	**S** (შ)	01101100	108	***l*** (ლ)
01010100	84	**T** (თ)	01101101	109	**m** (მ)
01010101	85	**U**	01101110	110	**n** (ნ)
01010110	86	**V**	01101111	111	**o** (ო)
01010111	87	**W** (ჭ)	01110000	112	**p**
01011000	88	**X**	01110001	113	**q**
01011001	89	**Y**	01110010	114	**@**
01011010	90	**Z** (ძ)	01110011	115	**r** (რ)
01011011	91	**[**	01110100	116	**s** (ს)
01011100	92	\	01110101	117	**t** (ტ)
01011101	93	**]**	01110110	118	**v** (ვ)
01011110	94	^	01110111	119	**w** (წ)
01011111	95	**_**	01111000	120	**z**
01100000	96	'	01111001	121	**y** (ყ)
01100001	97	**a** (ა)	01111010	122	**z** (ზ)
01100010	98	**b** (ბ)	01111011	123	**{**
01100011	99	**c** (ც)	01111100	124	|
01100100	100	**d** (დ)	01111101	125	**}**
01100101	101	**e** (ე)	01111110	126	~
01100110	102	**f** (ფ)	01111111	127	**DEL**
01100111	103	**g** (გ)			
01101000	104	**h** (ჰ)			
01101001	105	**i** (ი)			
01101010	106	**j** (ჯ)			

6 კოდირების **ASCII** ცხრილის იმ ნაწილში, რომელშიც განთავსებულია 32-დან დაწყებული 127-ის ჩათვლით დამთავრებული რიგითი ნომრის მქონე სიმბოლოები, შეტანილია ინგლისური ენის ალფაბეტი, ამიტომ მას აღნიშნული ცხრილის **ინგლისურ ნაწილი** ეწოდება და **საერთაშორისო სტანდარტად ითვლება**; ამავე ნაწილშია განთავსებული სასვენი ნიშნები, სხვადასხვა სახის ფრჩხილები, არაბული ციფრები, არითმეტიკული ოპერაციების ნიშნები და სხვა დამხმარე სიმბოლოები (ცხრ.2.13). იგი საშუალებას გვაძლევს ვაჩვენოთ თუ რა სახით წარმოიდგინება ინგლისური ენაზე დაწერილი ნებისმიერი სიტყვა კომპიუტერის მეხსიერებაში. მაგალითად, 2.25 ნახაზზე ნაჩვენებია სიტყვების “**file**”და **“ბადე”** კომპიუტერის მეხსიერებაში წარმოდგენის მაგალითი.

7 კოდირების **ASCII** ცხრილის გაფართოებულ (ნაციონალურ) ნაწილში (იხ. ცხრ.2.11) მოთავსებულია ფსევდოგრაფიკის სიმბოლოები, სხვადასხვა ევროპული ქვეყნების ნაციონალური ენების ალფაბეტები და სპეციალური ნიშნები. სიმბოლოების მასში განთავსების გათვალისწინებით ხდება საზღვარგარეთული წარმოების პროგრამების აბსოლუტური უმრავლესობა.

ბოლო პერიოდში ვრცელდება ტექსტური მონაცემების კოდირების უნივერსალური ცხრილი **UNICOD.** მასში სიმბოლოები კოდირდება არა **8-**, არამედ **16**-თანრიგიანი ორობითი რიცხვებით. იგი 2^{16}=**65536** სიმბოლოს კოდირების შესაძლებლობას იძლევა.



ნახ.2.27. კომპიუტერის მეხსიერებაში სიტყვების წარმოდგენის მაილუსტრირებელი სქემა

8 **ASCII** კოდი მსოფლიოში არსებული ენებიდან ყველაზე მეტად ინგლისურ ენასთან არის მისადაგებული. მაგალითად, გერმანულ ენაში არსებობს ე.წ. უმლაუტები, ხოლო ფრანგულ ენაში არსებობს სპეციალური სტრიქონზედა (დიაკრატიკული) ნიშნები. ევროპის ზოგიერთი ქვეყნის ენებში არსებობს რამდენიმე ასო, რომელიც არ შედის **ASCII**–სიმბოლოთა ნაკრებში, სლავურ ან არაბულ ენებს კი სრუ-

ლიად სხვანაირი ალფაბეტი აქვს, ხოლო ჩინურ ენაში ალფაბეტის ნაცვლად იერო-გრაფებია გამოყენებული.

კომპიუტერები მთელ მსოფლიოში ვრცელდება და ამიტომ პროგრამული უზრუნველ-ყოფის დამმუშავებლები დაინტერესებულები არიან საკუთარი პროდუქცია იმ ქვეყნებშიც გაავრცელონ, სადაც მომხმარებელთა უმრავლესობა არ ლაპარაკობს ინგლისურად და დამწერლობაში სიმბოლოთა სულ სხვა ნაკრებებს გამოიყენებს.

ზემოთ აღნიშნულიდან გამომდინარე, ბუნებრივად წარმოიშვა **ASCII** კოდის გაფართოების პრობლემა. მისი გადაწყვეტის **პირველი მცდელობისას ASCII** –სიმ-ბოლოებს დაუმატეს კიდევ **128** სიმბოლო, რის შედეგადაც მიიღეს **Latin-1**-ის სახელწო-დების მქონე **8-**თანრიგიანი ნაკრები**.** ამ ახალმა სტანდარტმა მიიღო სახელწოდება **IS 646.** კოდურ ცხრილს დაემატა შტრიხებისა და დიაკრიტიულ ნიშნებიანი ლათინური ასოები.

პრობლემის გადაწყვეტის **შემდგომი მცდელობის დროს** დამუშავებული იქნა **IS 8859** სტანდარტი და მისი სამი, კერძოდ **IS 8859-1, IS 8859-2** და **IS 8859-3** ვარიანტი. **IS 8859** სტანდარტში ენის ან ენათა ჯგუფის განსასაზღვრად შემოტანილი იქნა **256** სიმბოლო-საგან შედგენილი ნაკრები, რომელსაც ეწოდა **კოდური გვერდი**. **IS 8859-1** სტანდარტში აღნიშნულ ნაკრებს წარმოადგენს **Latin-1**; **IS 8859-2** სტანდარტი მოიცავს ლათინური ალფაბეტის მქონე სლავურ (ჩეხურ, პოლონურ და უნგრულ) ენებს, ხოლო **IS 8859-3** სტანდარტი აღწერს თურქულ, მალტის, გალურ ენებს, ესპერანტოს და ა.შ. ასეთი მი-დგომის **ძირითადი ნაკლია** ის, რომ პროგრამულმა უზრუნველყოფამ უნდა აკონტროლოს, თუ კერძოდ რომელ კოდურ გვერდთან აქვს მას საქმე და, ამავე დროს, ენების არევა დაუშვებელია. ამასთანავე კოდური ცხრილების გარეთ რჩებოდა მსოფლიოში არსებული ზოგიერთი (მაგ, იაპონური, ჩინური და ა.შ.) ენები.

9

პრობლემა საბოლოოდ გადაწყდა **Unicode**-ს სახელწოდების კოდირების ახალი სისტემის შექმნით, რომელიც საერთაშორისო **IS 10646** სტანდარ-ტად იქნა აღიარებული. მოცემული სტანდარტის დროს სიმბოლოების კოდირება ხდება არა **8**-თანრიგიანი ორობითი რიცხვებით, არამედ **16**-თანრიგიანი ორობი-თი რიცხვებით. თექვსმეტი თანრიგი საშუალებას იძლევა საკუთარი უნიკალური კოდით უზრუნველყოფილი იქნას $\mathbf{2^{16} = 65536}$ სხვადასხვა სიმბოლო.

სისტემა **Unicode**-მ გადაწყვიტა მრავალი, მაგრამ არა აბსოლუტურად ყველა პრო-ბლემა. მაგალითად, ლათინური და მრავალი სხვა ენების ალფაბეტები მოწესრიგებულია, ხოლო იეროგრაფები კი არა; ამიტომ ინგლისური, გერმანული, ფრანგული, რუსული და ა.შ. ენების პროგრამებს შეუძლიათ სიტყვები ალფაბეტის მიხედვით განალაგოს ამ სიტყ-ვებში შემავალი ასოების კოდების შედარების გზით, ხოლო იაპონური ან ჩინური ენების პროგრამებს ლექსიკონში სიმბოლოების თანამიმდევრობათა გამოსათვლელად დამატებითი ცხრილები სჭირდებათ.

კიდევ ახალ პრობლემას წარმოადგენს სალაპარაკო ენაში ახალი ტერმინების გაჩენა. მათი გამოჩენა ანბანისაგან შედგენილი სიტყვების შემთხვევაში ახალ კოდებს არ საჭი-როებს, მაგრამ იეროგრიფების შემთხვევაში ასეთი კოდების შემოღება აუცილებელია. გარდა ახალი ტერმინებისა, აუცილებელია სულ მცირე **20000** ახალი საკუთარი და გე-ოგრაფიული დასახელებების (ძირითადად ჩინურ ენაზე) დამატება, ასამოქმედებელია **ბრა-ილის შრიფტიც** და ა.შ.

ამ და მსგავსი სხვა პრობლემების გამო საბოლოოდ აღმოჩნდა, რომ **65536** კოდური სიტყვა არაა საკმარისი ყველა არსებული მოთხოვნის დასაკმაყოფილებლად; ამიტომ **1996** წელს განსაზღვრული იქნა თექვსმეტი დამატებითი **16**-თანრიგიანი **სიბრტყეები** და კოდური სიტყვების საერთო რაოდენობა **65536 + 16 • 65536** = **1 114 112**-მდე იქნა გა-ზრდილი.

2.10. გრაფიკული ინფორმაციის წარმოდგენა ორობითი კოდებით

1 გრაფიკულ ინფორმაციას მიეკუთვნება სხვადასხვა სახის გამოსახულებები (ნახატები, სურათები, ვიდეოგამოსახულებები, ნახაზები და ა.შ.). არსებობს მათი წარმოდგენის შემდეგი ორი ხერხი:

● **რასტრული გრაფიკის ხერხი**, რომლის დროსაც გრაფიკული ობიექტი ფორმირდება გამოსახულების გარკვეული რაოდენობის ელემენტების ერთობლიობის სახით. აღნიშნულ **გამოსახულების ელემენტებს** ეწოდება **პიქსელები** (ინგლ. **pixsel** წარმოადგენს გამოსახულება **picture element**-ის აბრევიატურას, რაც სიტყვა-სიტყვით ნიშნავს გამოსახულების, ნახატის ელემენტს). **პიქსელი** ეწოდება რასტრულ გრაფიკაში უმცირეს ორგანზომილებიან გამოსახულებას, რომელთა მეშვეობითაც ფორმირდება გამოსახულება. პიქსელი წარმოადგენს გარკვეული ფერის მქონე განუყოფად ოთკუთხოვანი ან მრგვალი ფორმის ობიექტს. რაც უფრო მეტია გამოსახულების ფართობის ერთეულში არსებული პიქსელების რაოდენობა, მით უფრო დეტალურია აღნიშნული გამოსახულება. **რასტრული გამოსახულება** წარმოადგენს კომპიუტერის მონიტორზე, ქაღალდზე ან სხვა ამსახ მოწყობილობებსა და მასალებზე პიქსელების ან ფერადი (ჩვეულებრივ, ოთკუთხედოვანი) წერტილების ბადეს.

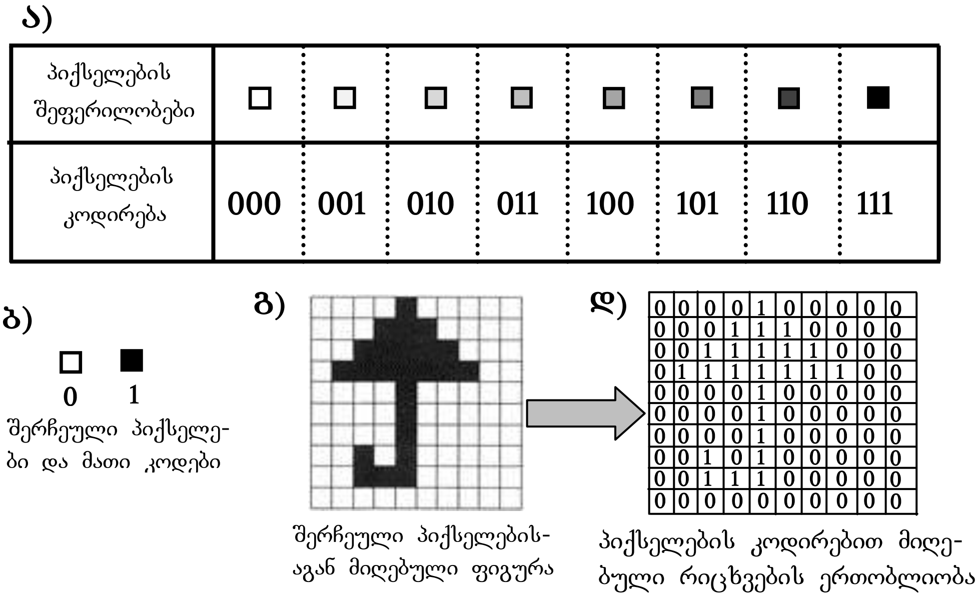
● **ვექტორული გრაფიკის ხერხი,** რომლის დროსაც გრაფიკული გამოსახულება ფორმირდება ხაზების, ვექტორებისა და წრეტილებისაგან.

2 განვიხილოთ შავ-თეთრი გამოსახულების რასტრული ფორმირების ხერხი. ასეთ შემთხვევაში გამოიყენება **რუხი ფერის** პიქსელები. **2.26,ა** ნახაზზე ნაჩვენებია **8** სხვადასხვა გრადაციის მქონე რუხი ფერის პიქსელი; ისინი მარცხნიდან მარჯვნივ განლაგებულია მათი რუხი ფერის ინტენსიურობის ზრდადობის მიხედვით. მოვახდინოთ **3**-თანრიგიანი ორობითი რიცხვებით მოცემული პიქსელების ფერების ინტენსეურობების (სიკაშკაშის) **შემდეგნაირად კოდირება**: რაც უფრო მაღალია მოცემული პიქსელის რუხი ფერის ინტენსიურობა, მით უფრო დიდი **3**-ნიშნა ორობითი რიცხვი მივანიჭოთ მას. ასეთ კოდირებას ვუწოდოთ **ფერის ინტენსიურობათა კოდირება ინტენსიურობების ზრდადობის მიხედვით.** ასეთი კოდირების დროს პიქსელისადმი მინიჭებული ორობითი რიცხვი გვიჩვენებს მისი ფერის ინტენსიურობას: რაც უფრო დიდი ორობითი რიცხვია პიქსელისათვის მინიჭებული, მით უფრო მაღალია მისი ფერის ინტენსიურობა.

აღნიშნული პიქსელებით გამოსახულების აგების პროცესის გასამარტივებლად შევარჩიოთ მაქსიმალურად კონტრასტული ფერის მხოლოდ ორი პიქსელი. ასეთებია ორობითი **000** და **111** რიცხვებით კოდირებული პიქსელები (ნახ.**2.28,ბ**). მათი დახმარებით აგებული ფიგურა **2.28,გ** ნახაზზეა ნაჩვენები. ამ უკანასკნელის ''გაციფროვნებისათვის'', ე.ი ორობითი რიცხვების ერთობლიობად გადასაქცევად, საჭიროა შევასრულოთ შემდეგი ალგორითმი:

1. **2.28** ნახაზზე არსებული თითოეული პიქსელი შევცვალოთ მისთვის მინიჭებული ორობითი რიცხვით (**000**-ით ან **111**-ით);
2. წინა პუნქტში მიღებული ორობითი რიცხვები ისე განვათავსოთ, როგორც მათი შესაბამისი პიქსელებია განთავსებული **2.28,გ** ნახაზზე;
3. ალგორითმის დასასრული.

მოცემული ალგორითმის შესრულების შედეგად მიღებული გრაფიკული გამოსახულება **2.28,**გ ნახაზზეა ნაჩვენები; იგი შეიძლება გარდავქმნათ ორობითი რიცხვების ერთობლიობად. გამოსახულებასა და ორობითი რიცხვების მიღებულ ერთობლიობას შორის არსებობს ურთიერთცალსახა დამოკიდებულება; ამის გამო ერთ-ერთის არსებობის შემთხვევაში შეიძლება აღვადგინოთ მეორე.



ნახ.2.28. შავ-თეთრი სისტემის დროს პიქსელების შეფერილობების **3**-თანრიგიანი კოდებით კოდირებისა და პიქსელების გამოყენებით გამოსახულების აგების ილუსტრირება

გამოსახულების დამახსოვრების უნარი არ აქვს კომპიუტერის მეხსიერებას, მაგრამ მასში შეიძლება ჩავწეროთ ორობითი რიცხვების მიღებული ერთობლიობა, რომელსაც იგი დაიმახსოვრებს; საჭიროების შემთხვევაში მეხსიერების მიერ დამახსოვრებული ორობითი რიცხვების მიმდევრობა შეიძლება მონიტორის ეკრანზე გარდაიქმნას სათანადო გამოსახულებად ან პრინტერის საშუალებით ეს გამოსახულება ამოიბეჭდოს ქაღალდზე.

რვა განსხვავებული ინტენსიურობის მქონე რუხი პიქსელის გამოყენებით აგებული გრაფიკული გამოსახულების (ნახატია ან სურათის) ხარისხი დაბალია. მაღალი ხარისხის შავ-თეთრი გამოსახულების მისაღებად საჭიროა აგებისათვის განსხვავებული ინტენსიურობების რაოდენობა **256**-მდე გავზარდოთ, ე.ი გამოვიყენოთ **256** გრადაციის მქონე რუხი პიქსელები. მათი კოდირებისათვის აუცილებელია **8**-თანრიგიანი ორობითი რიცხვების გამოყენება ($2^8 = 256$). კოდირება უნდა მოვახდინოთ **ფერის ინტენსიურობის ზრდადობის** მიხედვით.

3

ფერადი გამოსახულების რასტრული წარმოდგენა ემყარება ფერების შერევის შესახებ გერმანელი მათემატიკოსისა და ფიზიკოსის **ჰერმან გრასმანის** (Hermann Grassmann) მიერ ფორმულირებულ შემდეგ სამ კანონს:

ჰერმან გრასმანი
(1809 – 1877)

1. სამგანზომილებადობის კანონი: ნებისმიერი ფერი შეიძლება ფორმირდეს სამი **ძირითადი ფერის** კომბინირებით;

2. უწყვეტობის კანონი: შესაძლებელია ნებისმიერ ფერთან უსასრულოდ ახლომდებარე ფერის შერჩევა;

3. ადიტიურობის (შეკრების) კანონი: ფერების შერევით მიღებული კომბინირებული ფერი მხოლოდ შემადგენელ ფერებზეა დამოკიდებული.

ზემოთ ფორმულირებული კანონები უდევს საფუძვლად ფერების შესახებ თანამედროვე თეორიას.

ფერადი გამოსახულების რასტრულად წარმოდგენისათვის გრასმანის მოძღვრების თანახმად, უპირველეს ყოვლისა, აუცილებელია შევარჩიოთ სამი ძირითადი ფერი და თითოეული ფერისათვის ჩავატაროთ ისეთივე მანიპულაციები, როგორც ზემოთ ჩავატარეთ რუხი ფერისათვის.

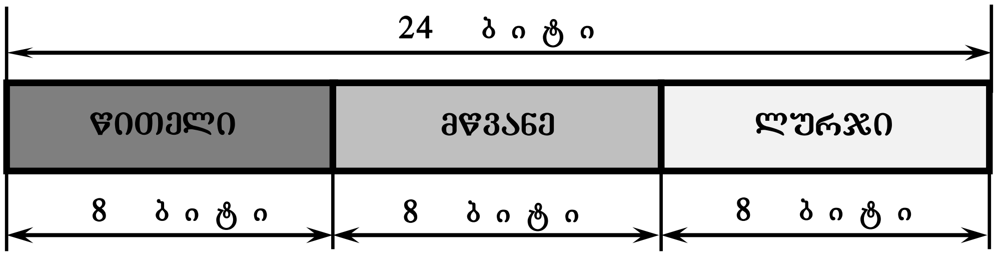
4

ჰერმან გრასმანის თეორიაზე დაყრდნობით შემუშავებული ფერადი გამოსახულების ფორმირების ერთ-ერთი ხერხია **RGB**-მეთოდი. მეთოდის სახელწოდება მიღებულია ძირითად ფერებად შერჩეული წითელი (**R**ed), მწვანე (**G**reen) და ლურჯი (**B**lue) ფერების ინგლისური სახელწოდებების პირველი ასოებისაგან. აღნიშნული მეთოდი ეყრდნობა იმას, რომ ადამიანის თვალი ნებისმიერ ფერს აღიქვამს როგორც ზემოთ აღნიშნული ძირითადი (წითელი, მწვანე და ლურჯი) ფერების შეკრების შედეგად მიღებულ ფერს; მაგალითად:

- ყვითელი ფერი მიიღება წითელი და მწვანე ფერების შეკრებით;
- ცისფერი მიიღება მწვანე და ლურჯი ფერების შეკრებით;
- მეწამული ფერი მიიღება წითელი და ლურჯი ფერების შეკრებით და ა.შ.

შავ-თეთრი გამოსახულების მისაღებად მონიტორის ერთსა და იგივე წერტილისაკენ მიმართული იყო ერთი, კერძოდ რუხი ფერის სხივი და აღნიშნულ სხივს გააჩნდა **256** გრადაცია (ინტენსიურობა); ფერადი პიქსელის მისაღებად მონიტორის ეკრანის ერთსა და იგივე წერტილისკენ მიმართული უნდა იყოს არა ერთი, არამედ ერთდროულად სამი სხვადასხვა, კერძოდ წითელი, მწვანე და ლურჯი ფერის შუქი.

თითოეული ფერის კოდირებისათვის თუ გამოვიყენებთ ერთ ბიტს, მაშინ ნულოვანი ბიტი აღნიშნული ფერის არარსებობის, ხოლო ერთის ტოლი ბიტი პირიქით – მისი არსებობის გამომხატველი იქნება. მაშასადამე, პიქსელის ერთი ფერის კოდირებისათვის სულ **3**-ბიტი (თითოეულ ძირითად ფერზე თითო) იქნება საკმარისი. დავუშვათ, რომ პირველი ბიტი შეესაბამება წითელ, მეორე ბიტი – მწვანე, ხოლო მესამე ბიტი – ლურჯ ფერს. მაშინ ყვითელ ფერს აღნიშნავს კოდი $\mathbf{110}_{(2)}$ (წითელი და მწვანე ფერები არის, ხოლო ლურჯი ფერი – არ არის), ცისფერ ფერს – კოდი $\mathbf{011}_{(2)}$ (წითელი ფერი არ არის, არის მწვანე და ლურჯი ფერები) და ა.შ.



ნახ. 2.29. ფერადი გამოსახულების კოდირება

კოდირების ზემოთ აღწერილი სქემის დროს თითოეულ პიქსელს რვა შესაძლო ფერიდან (2^3=8) შეიძლება ჰქონდეს ერთ-ერთი ფერი. თითოეული ძირითადი ფერის კოდირებისათვის თუ გამოვიყენებთ 8 ბიტს ანუ 1 ბაიტს (**ნახ.**2.29), როგორც ეს დღეს ფართოდ გამოიყენება, შესაძლებელი გახდება თითოეულ ძირითად ფერს მივცეთ 256 გრადაცია; კონკრეტული ძირითადი ფერის ცალკე აღებულ გრადაციას აღნიშნული ფერის გარკვეული ინტენსიურობა (სიკაშკაშე) შეესაბამება; ამიტომ 8-ბიტური (1-ბაიტური) ორობითი რიცხვებით ფაქტობრივად გამოყენებული ძირითადი ფერების გრადაციები (ინტენსიურობები) აღმოჩნდება კოდირებული. სხვადასხვა ინტენსიურობის ძირითადი ფერების შეკრებით განსხვავებული ფერი მიიღება, რომელთაგანაც თითოეული კონკრეტული პიქსელით იქნება წარმოდგენილი. გამოსახული ფერების ანუ პიქსელების საერთო **N** რაოდენობა შემდეგნაირად განისაზღვრება:

$$N = 256 \cdot 256 \cdot 256 = 16.777.216.$$

მოცემული სქემის დროს კომპიუტერის მეხსიერებაში ერთი პიქსელის შესანახად 24 ბიტი ანუ 3 ბაიტი იქნება საჭირო.

წითელი **R,** მწვანე **G** და ლურჯი **B** ფერის ინტენსიურობების კოდირებისათვის გამოყენებულ რიცხვებს თუ შესამამისად $\breve{R}$, $\breve{G}$ და $\breve{B}$ სიმბოლოებით აღვნიშნავთ, მაშინ ცალკე აღებული თითოეული პიქსელის კოდი პირობით სამჯერადი ($\breve{R}$, $\breve{G}$, $\breve{B}$) რიცხვით შეიძლება ავღნიშნოთ; მოცემულ აღნიშვნაში $\breve{R}$, $\breve{G}$, $\breve{B}$ რიცხვებიდან თითოეული მათგანი შესაბამისად წითელი **R**, მწვანე **G** და ლურჯი **B** ფერის ინტენსიურობას (სიკაშკაშეს) გვიჩვენებს.

სამჯერადი ($\breve{R}$, $\breve{G}$, $\breve{B}$) რიცხვის სახელწოდებად გამოიყენება ლათინური ტერმინი **ტრიპლეტი** (ლათ. **triplus** – ''სამჯერადი''). $\breve{R}$, $\breve{G}$, $\breve{B}$ რიცხვები შეიძლება გამოისახოს თვლის როგორც ორობით, ასევე ათობით სისტემაში. მეორე შემთხვევაში მნიშვნელოვნად მცირდება ტრიპლეტის ჩანაწერის სიგრძე.

2.30,ა ნახაზზე მოცემულია სხვადასხვა ფერების შესაბამისი ტრიპლეტები, რომლებშიც გამოყენებულია $\breve{R}$, $\breve{G}$, $\breve{B}$ რიცხვების ექსტრემალური (მინიმალური და მაქსიმალური) მნიშვნელობები. ამ დროს მიიღება **მაქსიმალური ინტენსიურობის** ჯამური ფერები. ცხადია, რომ $\breve{R}$, $\breve{G}$, $\breve{B}$ რიცხვებმა შეიძლება მიიღონ ექსტრემალურ მნიშვნელობებს შორის არსებული ნებისმიერი საშუალედო მნიშვნელობები, რაც ცვლის ჯამურად მიღებული ფერების ინტენსივობებს (შეფერილობებს).

2.30,ა ნახაზიდან ჩანს, თვლის აღნიშნული სისტემის გამოყენების დროს ტრიპლეტების სიგრძე საკმაოდ დიდია, რაც არაეკონომიურს ხდის მათ ჩანაწერებს. $\breve{R}$, $\breve{G}$, $\breve{B}$ რიცხვებს თუ თვლის ათობით სისტემის გამოყენებით გამოვსახავთ, მაშინ ტრიპლეტების ჩანაწერები კომპაქტურები გახდება (ნახ. **2.30,გ**).

2.28,ა,გ ნახაზის თანახმად მაქსიმალური ინტენსიურობის მქონე ჯამურ:

- ცისფერი ფერს შეესაბამება (**00000000, 11111111, 11111111**) ანუ (**0, 256, 256**) ტრიპლეტი; იგი მიიღება მაქსიმალური ინტენსიურობის მქონე მწვანე და ლურჯი ფერე-ბის შეკრებით;

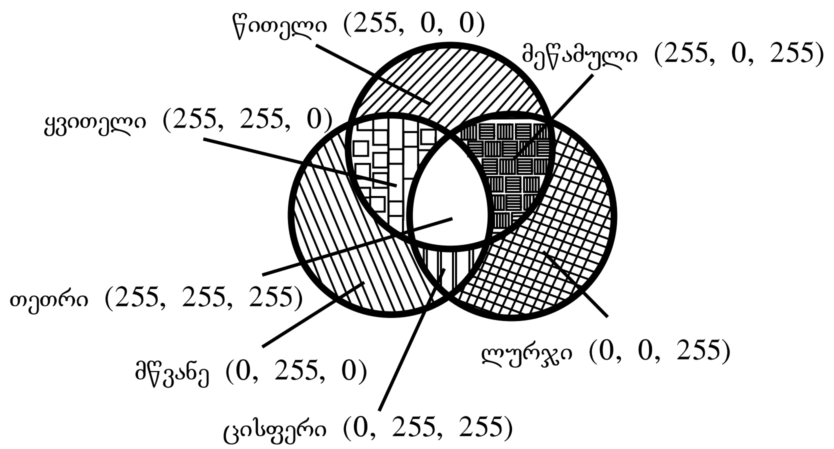
ა)

ფ ე რ ი	წითელი	მწვანე	ლურჯი
წითელი	(11111111	00000000	00000000)
მწვანე	(0000000	11111111	00000000)
ლურჯი	(0000000	00000000	11111111)
ცისფერი	(0000000	11111111	11111111)
მეწამული	(11111111	00000000	11111111)
ყვითელი	(11111111	11111111	00000000)
თეთრი	(11111111	11111111	11111111)
შ ა ვ ი	(0000000	00000000	00000000)

ბ)

00000000 ⟶ 0	
11111111 ⟶ 255	

გ)



ნახ.2.30. ფერითი **RGB** მეთოდი

- მეწამულ ფერს შეესაბამება (**11111111, 00000000, 11111111**) ანუ (**256, 0, 256**) ტრიპლეტი; იგი მიიღება მაქსიმალური ინტენსიურობის მქონე წითელი და ლურჯი ფერების შეკრებით;
- ყვითელ ფერს შეესაბამება (**11111111, 11111111, 00000000**) ანუ (**256, 256, 0**) ტრიპლეტი; იგი მიიღება მაქსიმალური ინტენსიურობის მქონე წითელი და მწვანე ფერების შეკრებით;

სამივე $\breve{R}$, $\breve{G}$, $\breve{B}$ სიდიდეს თუ ერთი და იგივე მნიშვნელობა ექნება, მაგალითად **(53, 53, 53)**, **(148, 148, 148)**, **(195, 195, 195)** და ა.შ., მაშინ ჯამურად გარკვეული ინტენსიურობის (სიკაშკაშის) მქონე რუხი ფერი მიიღება. მინიმალური ინტენსიურობის (სიკაშკაშის) რუხი ფერი გადაიქცევა შავ ფერად, ხოლო მაქსიმალური სიკაშკაშის რუხი ფერი – თეთრ ფერად; მათი ტრიპლეტები იქნება:

- შავი ფერისათვის: (**00000000, 00000000, 00000000**) ანუ **(0, 0, 0)**;
- თეთრი ფერისათვის: (**11111111, 11111111, 11111111**) ანუ (**256, 256, 256**).

6 მონიტორზე გამოტანილი ფერები მონიტორზე მიიღებოდა ძირითადი ფერების შესაბამისი სხივების დაშუქებით. სხვაგვარადაა საქმე პოლიგრაგიულ წარმოებაში ქაღალდზე სხვადასხვა ფერების გამოსახვისას. ამ შემთხვევაში თეთრ ქაღალდზე, რომელზედაც სხვადასხვა ფერები უნდა იქნეს დატანილი, ეცემა თეთრი ფერის შუქი, რომელიც ყველა ძირითადი ფერების შეკრებითაა მიღებული. ჩვენ აღვიქვამთ ქაღალდიდან ანარეკლ შუქს, რომელსაც აქვს გარკვეული ფერი. ამ უკანასკნელის მისაღებად ქაღალდზე დატანილმა საღებარმა ნივთიერებამ დაცემული თეთრი ფერის სხივიდან უნდა შთანთქოს ყველა ფერი, გარდა იმ ფერისა, რომლის მიღებაც ჩვენ გვსურს. ე.ი. მოცემულ შემთხვევაში ფერები მიიღება თეთრი (**255, 255, 255**) ფერიდან მდგენელი (წითელი, მწვანე ან ლურჯი) ფერის გამოკლებით. კერძოდ, წითელი მდგენელის გამოკლების შედეგად მიიღება (**0, 255, 255**) ტრიპლეტი, რომელსაც შეესაბამება **ცისფერი ფერი**; მწვანე მდგენელის გამოკლებით - (**255, 0, 255**) ტრიპლეტი, რომელსაც შეესაბამება **მეწამული ფერი,** ხოლო ლურჯი მდგენელის გამოკლებით – (**255, 255, 0**) ტრიპლეტი, რომელსაც შეესაბამება **ყვითელი ფერი**.

ფერების მიღების მეთოდისთვის, რომლის დროსაც თეთრი ფერიდან მისი მდგენელები გამოირიცხება, ძირითად ფერებად ითვლება ზემოთ ხაზგასმული ცისფერი, მეწამული და ყვითელი ფერები. მეთოდის სახელწოდებად მიღებულია **ცისფერი** (**C**yan), **მეწამული** (**M**agenta) და **ყვითელი** (**Y**ellow) ფერების ინგლისური სახელწოდებების პირველი ასოებისაგან ფორმირებული აბრევიატურა: **CMY**-მეთოდი ეწოდება. მეთოდის ნაკლია ის, რომ მისი გამოყენების დროს ვერ ხერხდება მაღალხარისხოვანი შავი ფერის მიღება; რნიშნული ნაკლის გამოსასწორებლად ძირითად ფერებს **შავი** (blac**K**) ფერი და მეთოდმა მიიღო **CMYK**-მეთოდის სახელწოდება.

7 გამოსახულების **ციფრებად გარდაქმნის**, ანუ, მისი **გაციფრების** დროს ობიექტივის დახმარებით აღნიშნული გამოსახულება პროეცირდება **რასტრად** წოდებულ **m** რაოდენობის სტრიქონებისა და რაოდენობის **n** სვეტების შემცველ შუქმგრძნობიარე მატრიცაზე. მატრიცის თითოეული ელემენტია უმცირესი ზომის წერტილი, რომელიც ფერადი გამოსახულების დროს შედგება წითელი, მწვანე და ლურჯი ფერის სამი შუქმგრძნობიარე გადამწოდისაგან. შემდეგ თვითოეული ფერის შესაბამისი წერტილების სიკიშკაშის ინტენსიურობა მთელი რასტრის მასშტაბით სათანადო ციფრებად გარდაიქმნება.

სამი ძირითადი ფერის შესაბამისი თითოეული წერტილის სიკაშკაშის კოდირებისთვის **8** ბიტის (**1** ბაიტის) გამოყენების დროს მივიღებთ თვალის მიერ აღსაქმელ 2^{24} =**16777216** რაოდენობის სხვადასხვა ფერს, რომელიც უახლოვდება ადამიანის მხედველობის მიერ აღსაქმელი ფერების რაოდენობას. **24** თანრიგისაგან შემდგარი ორობითი კოდით ფერადი გრაფიკის წარმოდგენის რეჟიმს ეწოდება **სრული ფერადოვნების** ანუ

True Color-ის რეჟიმი. ცხადია, რომ ბგერითი მონაცემების ანალოგიურად გრაფიკული მონაცემებიც მეხსიერების მოწყობილობებში ძალიან დიდ ადგილს იკავებს. მაგალითად, თანამედროვე თვალსაზრისით, ძალიან მოკრძალებულ მონიტორს გააჩნია **800 x 600** რაოდენობის მქონე წერტილები; ამ შემთხვევაში **True Color**-ის რეჟიმში წარმოდგენილი გამოსახულება დაიკავებს

$$800 \cdot 600 \cdot 24 = 11.520.000 \text{ ბიტის} = 1.440.000 \text{ ბაიტის}$$

ტოლ მოცულობას.

ფერის უმაღლესი ხარისხით გამოსახვა თუ სავალდებულო არ არის, რასტრის თითოეული წერტილი კოდირდება **16** ბიტით ანუ **2** ბაიტით; ამ დროს მიიღება 2^{16} = **65536** რაოდენობის ერთმანეთისაგან განსხვავებული ფერი. **16** თანრიგისაგან შემდგარი ორობითი კოდით ფერადი გრაფიკის წარმოდგენის რეჟიმს ეწოდება **მაღალი ფერადოვნების** ანუ **High Color**-ის რეჟიმი.

რეჟიმს, რომლის დროსაც რასტრის ერთი წერტილის კოდირებისათვის გამოიყენება **8** ბიტი, ანუ **1** ბაიტი, **კოდირების ინდექსური რეჟიმი** ეწოდება. ამ დროს უზრუნველყოფლია ერთმანეთისაგან განსხვავებული 2^8 = **256** რაოდენობის ფერის არსებობა. ეს არ არის საკმარისი ფერების მთელი გამის გადმოსაცემად. თითოეული წერტილის კოდი ამ დროს გამოხატავს არა უშუალოდ ფერს, არამედ **პალიტრად** წოდებულ ფერების ცხრილში არსებული გარკვეული ფერის ნომერს (ინდექსს). პალიტრა გრაფიკული მონაცემების მქონე ფაილებს უნდა მიმაგრდეს და უნდა მოხდეს მათი გამოყენება გამოსახულების აღწარმოების დროს.

რასტრული მეთოდების ერთ-ერთ ნაკლია გამოსახულების ზომების ნებისმიერ არჩეულ მნიშვნელობამდე პროპორციულად შეცვლის სირთულე. გამოსახულების გადიდების ერთადერთ ხერხს არსებითად თავად პიქსელის ზომების გაზრდა წარმოადგენს; ეს უკანასკნელი კი იწვევს **მარცვლოვანობას** (**პიქსელიზაციას**) - გამოსახულება ''მარცვლოვანი'' ხდება.

8 გრაფიკული ინფორმაციის წარმოდგენის **ვექტორული ხერხები** საშუალებას გვაძლევს თავიდან ავიცილოთ რასტრული მეთოდებისათვის დამახასიათებელი პრობლემა, რომელიც წარმოიშვება გამოსახულების მასშტაბის ცვლილების დროს. ამ შემთხვევაში გამოსახულება წარმოიდგინება წრფეებისა და მრუდეების ერთობლიობათა სახით. მოწყობილობამ გამოსახულების შემადგენელი გარკვეული კონფიგურაციის პიქსელები კი არ უნდა აღაწარმოოს, არამედ დეტალურად უნდა აღწეროს გამოსახულების წარმომქმნელი წრფეებისა და მრუდეების ურთიერთგანლაგება. ამ მონაცემების საფუძველზე მოწყობილობა ქმნის მზა გამოსახულებას.

დაწვრილებითი ტექნოლოგიების დახმარებით არის აღწერილი თანამედროვე პრინტერებისა და მონიტორების მიერ გამოყენებული სხვადასხვა შრიფტები. აღნიშნული ტექნოლოგიები საშუალებას გვაძლევს სიმბოლოების ზომები ფართო ზღვრებში ვცვალოთ და მივიღოთ სხვადასხვა მასშტაბის შრიფტები. ასეთ ტექნოლოგიებს მიეკუთვნება:

- **Microsoft** და **Appl** კომპანიების მიერ დამუშავებული **True Type** სახელწოდების ტექნოლოგია;
- **Adoble Systems** კომპანიის მიერ დამუშავებული **PostScript** სახელწოდების ტექნოლოგია და ა.შ.

ვექტორული მეთოდები ფართოდ გამოიყენება **დაპროექტების ავტომატიზებულ სისტემებში;** მათი საშუალებით შესაძლებელია:

- მონიტორის ეკრანზე გამოვსახოთ რთული სამგანზომილებიანი ობიექტების გამოსახულებები
- მოვახდინოთ ეკრანზე გამოსახული ობიექტების მანიპულირება.

ვექტორული ტექნოლოგიის **ნაკლია** ის, რომ, რასტრული ტექნოლოგიისაგან განსხვავებით, იგი საშუალებას არ გვაძლევს მივიღოთ ობიექტების ფოტოგრაფიული სიზუსტის გამოსახულება.

2.11. ინფორმაცია, როგორც განუსაზღვრელობის მოხსნის საშუალება

ინფორმაციის რაიმე წყაროდან გარკვეული ინფორმაციის მიღებამდე ჩვენთვის უცნობია თუ რა მდგომარეობაში იმყოფება აღნიშნული წყარო, ე.ი. იგი წარმოადგენს თავისებურ შავ ყუთს, რომლის განუსაზღვრელობა მაქსიმალური, ანუ **1**-ის ტოლია.

საინფორმაციო პროცესების რეალიზაციის დროს ინფორმაცია სხვადასხვა ფორმით გადაიტანება ინფორმაციის წყაროდან მიმღებამდე. გადატანისათვის შეიძლება გამოყენებული იქნეს ბუნებრივი ან ხელოვნური (ფორმალური) ენის ნიშნები და სიმბოლოები, რომელთა საშუალებითაც ინფორმაციას ეძლევა **შეტყობინების** სახე.

შეტყობინება ეწოდება გადაცემისათვის გამოყენებული ნიშნების (სიმბოლოების) ერთობლიობის სახით ინფორმაციის წარმოდგენის ფორმას.

გამოსაკვლევ სისტემად განვიხილოთ ინფორმაციის **დისკრეტული წყარო** (დისკრეტული შეტყობინებების წყარო), რომელიც წარმოადგენს სასრული რაოდენობის a_i , $i = 1,2, \ldots, N$ მდგომარეობების მქონე ფიზიკური სისტემას.

სისტემის მდგომარეობების სრულ $\mathbf{A} = \{a_1, a_2, \ldots, a_N\}$ სიმრავლეს ინფორმაციის თეორიაში **აბსტრაქტული ალფაბეტი**, ანუ **შეტყობინებების წყაროს ალფაბეტი**, ხოლო მასში შემავალ ცალკეულ $a_1, a_2, \ldots, a_N$ მდგომარეობას - ამ **ალფაბეტის ასოები** ეწოდება.

ასეთი სისტემა დროის თითოეულ მომენტში შესაძლოა **შემთხვევით გადავიდეს** მდგომარეობათა სასრულ $\mathbf{A}$ სიმრავლეში შემავალ ნებისმიერ a_i მდგომარეობაში. ამ დროს ამბობენ, რომ **სხვადასხვა მდგომარეობები რეალიზდება წყაროს მიერ მათი ამორჩევის შედეგად.**

ვინაიდან რეალობაში წყარო ზოგიერთ მდგომარეობას ხშირად, ხოლო დანარჩენებს - იშვიათად ირჩევს, ამიტომ ზოგადად იგი შეიძლება $\mathbf{A}$ **ანსამბლის** საშუალებით დავახასიათოთ. **ანსამბლი** წარმოადგენს მდგომარეობების ისეთ ერთობლიობას, რომელშიც შემავალი თითოეული a_i მდგომარეობის ქვემოთ მითითებულია სისტემის მიერ ამ მდგომარეობაში გადასვლის p_i ალბათობა:

$$\mathbf{A} = \begin{bmatrix} a_1 & a_2 & \ldots & a_n \\ p_1 & p_2 & \ldots & p_N \end{bmatrix}, \text{ ამასთანავე } \sum_{i=1}^{N} p_i = 1. \qquad (2.45)$$

შემოვიტანოთ ინფორმაციის წყაროს მიერ **მდგომარეობის ამორჩევის განუსაზღვრელობის** ზომა. იგი შეიძლება იმ **ინფორმაციის ზომადაც** განვიხილოთ, რომლის მიღების შემდეგ უტყუარად გვეცოდინება თუ რა მდგომარეობა ამოირჩია ინფორმაციის წყარომ, ე.ი. აღმოიფხვრება წყაროს მიერ მდგომარეობის ამორჩევის განუსაზღვრელობა ინფორმაციის ახლად შემოტანილმა ზომამ უნდა დააკმაყოფილოს შემდეგი სამი პირობა:

1. ინფორმაციის წყაროს მიერ ამოსარჩევი მდგომარეობის რაოდენობის გაზრდისას შესაბამისად უნდა იზრდებოდეს ინფორმაციის ზომა; ამ დროს მხედველობაში არ უნდა მივიღოთ ინფორმაციის წყაროს ისეთი მდგომარეობები, რომლებსაც წყარო არასოდეს ამოირჩევს. ასეთი მდგომარეობის ამორჩევის ალბათობა ნულის ტოლია და მათ **წყაროს დაუშვებელი მდგომარეობა** ეწოდება.

2. ინფორმაციის წყაროს თუ ერთადერთი მდგომარეობის ამორჩევა შეუძლია, მაშინ მის მიერ მდგომარეობის ამორჩევის განუსაზღვრელობა ნულის ტოლია და ამდენად, ასეთ შემთხვევაში მიღებული ინფორმაციის სიდიდე **0**-ის ტოლი უნდა იყოს;

3. ინფორმაციის ზომა უნდა აკმაყოფილებდეს **ადიტიურობის, ანუ შეკრებადობის პირობას**, რომელიც ასე ფორმულირდება: N და M რაოდენობის თანაბარალბათური მდგომარეობების მქონე ორი დამოუკიდებელი წყაროს თუ მდგომარეობათა N_i, M_j წყვილის ერთდროულად მარეალიზებელ **ერთ საინფორმაციო წყაროდ** განვიხილავთ, მაშინ გაერთიანებული წყაროს განუსაზღვრელობა საწყისი წყაროების განუსაზღვრელობების ჯამი უნდა იყოს. ვინაიდან ინფორმაციის გაერთიანებული წყაროს მდგომარეობების საერთო რაოდენობა $N\,M$–ის ტოლია, ამიტომ, საძებნი ფუნქცია უნდა აკმაყოფილებდეს შემდეგ პირობას:

განუსაზღვრელობის ზომად არ შეიძლება ინფორმაციის წყაროს მდგომარეობის N რაოდენობა ავიღოთ, რადგან N=**1**-ის შემთხვევაში, როდესაც განუსაზღვრელობა ნულის ტოლია (არ არსებობს), მივიღებდით ერთის ტოლ განუსაზღვრელობას, რაც ეწინააღმდეგება ზემოთ ფორმულირებულ მე-**2** პირობას.

$$f(NM) = f(N) + f(M). \tag{2.46}$$

ზემოთ ფორმულირებული სამივე პირობა შესრულდება, თუ თანაბარალბათურ მდგომარეობიან ინფორმაციის წყაროსა და მისი მახასიათებელი **A** ანსამბლის განუსაზღვრელობის ზომად მივიღებთ არა მისი მდგომარეობის N რაოდენობას, არამედ ამ რაოდენობის ლოგარითმს:

$$\mathbf{H(A) = \log N.} \tag{2.47}$$

ასეთი ზომა დააკმაყოფილებს ზემოთ მოყვანილ სამივე პირობას. მართლაც:

- **N**-ის ზრდით მონოტონურად იზრდება **H(A)**, ე.ი. სრულდება **1**-ლი პირობა;
- თუ **N** = 1, მაშინ **log 1** = **0**, ე.ი. სრულდება მე-**2** პირობა;
- **log NM = log N + log M,** ე.ი სრულდება მე-**3** პირობა.

აღნიშნული ზომა **1928** წელს შემოგვთავაზა ამერიკელმა მეცნიერმა **რალფ ვინტონ ლაიონ ჰარტლიმ** (Ralph Vinton Lyon Hartley), ამიტომ (**2.47**) ფორმულას **ჰარტლის ფორმულასაც** უწოდებენ.

ჰარტლის ფორმულაში ლოგარითმის ფუძის შერჩევას პრინციპული მნიშვნელობა არა აქვს და მხოლოდ გაზომვის მასშტაბს ან ერთეულს განსაზღვრავს.

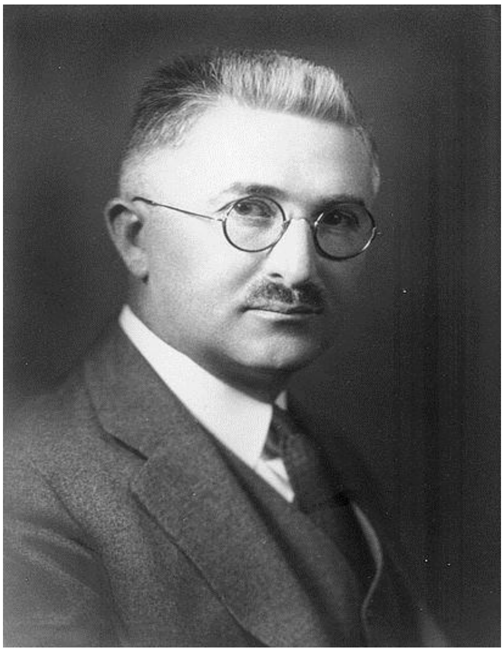
ლოგარითმის ფუძეზე დამოკიდებულებით გამოიყენება გაზომვის შემდეგი ერთეულები:

- **ბიტები;** ამ დროს ლოგარითმის ფუძე **2**-ის ტოლია:

$$\mathbf{H(A) = \log_2 N\ .} \tag{2.48}$$

- **ნატები;** ამ დროს ლოგარითმის ფუძე ნეპერის ანუ ეილერის ***e*** რიცხვის (***e*** = **= 2,71828…)** ტოლია:

$$\mathbf{H(A) = \log_e N = \ln N\ .} \qquad \mathbf{(2.48)}$$



(რ.ვ.ლ. ჰარტლი. 1888-1970)

● **დიტები;** ამ დროს ლოგარითმის ფუძე **10**-ის ტოლია:

$$H(A) = \log_{10}N = \lg N \qquad (2.50)$$

რადგან თანამედროვე კომპიუტერული სისტემის ასაგებად გამოიყენება ორი მდგრადი მდგომარეობის მქონე ელემენტი, ამიტომ ინფორმატიკაში განუსაზღვრელობის ერთეული (**2.48**) ფორმულით განისაზღვრება, მას **ორობითი ერთეული,** ანუ **ბიტი** ეწოდება და ასე განისაზღვრება:

ბიტი წარმოადგენს ორი თანაბარალბათური ხდომილებიდან ერთ-ერთი ხდომილების ამორჩევის დროს არსებულ განუსაზღვრელობას.

3

ჰარტლის ფორმულა შეიძლება ემპირულადაც, ანუ ექსპერიმენტის ჩაუტარებლადაც მივიღოთ. (ბერძ. **empiria** – გამოცდილება. **ემპირული –** ბუნებრივ პირობებში დაკვირვებით და არა ექსპერიმენტით მიღებული).

N=2 შედეგის მქონე ცდას **ორობითი ცდა** ვუწოდოთ. იგი (**2.48**) ფორმულის თანახმად ერთი ბიტის ტოლ ინფორმაციას შეიცავს (**$\log_2 2=1$).** ასეთ შემთხვევაში სამართლიანია შემდეგი მტკიცებულებები:

● **ორი** თანაბარალბათური ხდომილობისაგან შემდგარ სიტუაციაში განუსაზღვრელობის მოსახსნელად ერთი ორობითი ცდის ჩატარება და, მაშასადამე, **ერთი ბიტი** ინფორმაციაა საკმარისი; ამიტომ ორი თანაბარალბათური ხდომილობისაგან შემდგარ განუსაზღვრელობას **ერთი ბიტი** ინფორმაცია შეესაბამება.

● **ოთხი** თანაბარალბათური ხდომილობისაგან შემდგარ სიტუაციაში განუსაზღვრელობის მოსახსნელად ორი ორობითი ცდის ჩატარება და, მაშასადამე, ორი ბიტი ინფორმაციაა საკმარისი; ამიტომ ოთხი თანაბარალბათური ხდომილობისაგან შემდგარ განუსაზღვრელობას **ორი ბიტი** ინფორმაცია შეესაბამება.

● **რვა** თანაბარალბათური ხდომილობისაგან შემდგარ განუსაზღვრელობას **სამი ბიტი** ინფორმაცია შეესაბამება;

● **თექვსმეტი** თანაბარალბათური ხდომილობისაგან შემდგარ განუსაზღვრელობას **ოთხი ბიტი** ინფორმაცია შეესაბამება და ა.შ.

ზემოთ მოყვანილ მსჯელობას თუ გავაგრძელებთ, შეიძლება დავასკვნათ, რომ **32** ბანქოსაგან შემდგარი დასტიდან ამოღებული ერთი ბანქოს გამოსაცნობად **5** ბიტი ინფორმაციაა საკმარისი, ე.ი. საკმარისია **ხუთი** ისეთი კითხვის დასმა, რომელზედაც უნდა გაეცეს პასუხი **"დიახ"** ან **"არა"**.

ამგვარად, შეტყობინება **N** რაოდენობის თანაბარალბათური ვარიანტიდან თუ ერთ-ერთ ვარიანტზე მიგვითითებს, მაშინ მას ჩვენთან **$\log_2 N$** რაოდენობის ინფორმაცია მოაქვს. მართლაც, ზემოთ განხილული მაგალითებისათვის შესაბამისად გვაქვს: **$\log_2 2=1$; $\log_2 4=2$; $\log_2 8=3$; $\log_2 16=4$; $\log_2 32=5$** და ა.შ.

ჰარტლის ფორმულა სიტყვიერად შეიძლება ასე ჩამოვაყალიბოთ:

ინფორმაციის რაოდენობა იმ ხარისხის ტოლია, რომელშიც უნდა ავიყვანოთ **2,** რათა მივიღოთ ამოსარჩევი თანაბარალბათური ვარიანტების საერთო რაოდენობა.

4

შემოთავაზებული მიდგომა ისეთი პრაქტიკული ამოცანების გადაწყვეტის საშუალებას იძლევა, რომელთა დროსაც ინფორმაციის წყაროს ყველა შესაძლო მდგომარეობას ერთნაირი ალბათობა აქვს.

ზოგადად ინფორმაციის წყაროს მიერ მდგომარეობის რეალიზაციის განუსაზღვრელობის ხარისხი არა მარტო ამ წყაროს მდგომარეობების რაოდენობაზე, არამედ მდგომარეობათა ალბათობებზეც არის დამოკიდებული. მაგალითად, ინფორმაციის წყაროს თუ **0,999** და **0,001** ალბათობების მქონე ორი შესაძლო მდგომარეობა გააჩნია, მაშინ მის მიერ მდგომარეობის ამორჩევის განუსაზღვრელობა გაცილებით ნაკლებია, ვიდრე თანაბარ ალბათობებიანი ორი მდგომარეობის მქონე წყაროს მიერ მდგომარეობის ამორჩევის განუსაზღვრელობა; ასეთ შემთხვევაში შედეგი პრაქტიკულად წინასწარ შეიძლება ვიწინასწარმეტყველოთ (რეალიზებული იქნება **0,999** ალბათობის მქონე მდგომარეობა).

ამერიკელმა მეცნიერმა **კლოდ შენონმა** ამორჩევის **H** განუსაზღვრელობის ზომის ცნება განაზოგადა იმ შემთხვევისათვის, როდესაც **H** განუსაზღვრელობა არა მარტო ინფორმაციის წყაროს მდგომარეობათა რაოდენობაზე, არამედ ამ მდგომარეობების ალბათობებზეც (**A** ალფაბეტის a_i სიმბოლოთა ამორჩევის p_i ალბათობებზეც) არის დამოკიდებული. აღნიშნული ზომა გვიჩვენებს **საშუალოდ** ერთ მდგომარეობაზე მოსულ განუსაზღვრელობას, მას **ინფორმაციის დისკრეტული წყაროს ენტროპია** (ბერძ. **ἐντροπία** – "მობრუნება", "გარდაქმნა") ეწოდება და შემდეგი ფორმულით განისაზღვრება:

$$\mathbf{H} = -\sum_{i=1}^{N} p_i \log p_i \qquad (2.51)$$

განუსაზღვრელობა თუ გვინდა ორობით ერთეულებში გავზომოთ, მაშინ ლოგარითმის ფუძედ უნდა ორი ავიღოთ:

$$\mathbf{H} = -\sum_{i=1}^{N} p_i \log_2 p_i \qquad (2.52)$$

თანაბარი ალბათობების მქონე მდგომარეობის ამორჩევის დროს სამართლიანია შემდეგი პირობა:

$$p_1 = p_2 = \ldots = p_N = 1/\mathrm{N}. \qquad (2.53)$$

ამ უკანასკნელს თუ შევიტანთ **(2.52)**-ში, მაშინ მივიღებთ ზემოთ მოყვანილ ჰარტლის **(2.47)** ფორმულას.

5

განუსაზღვრელობის გაზომვისათვის **კლოდ შენონის** მიერ შემოთავაზებულ განზომილებას **ენტროპია** იმიტომ ეწოდა, რომ **(2.52)** გამოსახულება ფორმით ფიზიკური სისტემისათვის ადრე **ბოლცმანის** მიერ განსაზღვრული ენტროპიის გამოსახულების მსგავსია. **თერმოდინამიკის მეორე კანონის** თანახმად, ჩაკეტილი სივრცის **H** ენტროპია განისაზღვრება გამოსახულებით:

$$\mathbf{H} = \frac{1}{\mathbf{M}_n}\sum_{i=1}^{N} m_i \ln\frac{m_i}{\mathbf{M}_n}\quad , \qquad (2.54)$$

სადაც $\mathbf{M}_n$ არის მოცემულ სივრცეში მოლეკულების რაოდენობა, m_i კი – $(v_i + \Delta v)$ სიჩქარით მოძრავი მოლეკულების რაოდენობაა.

რადგან m_i/ M_n იმის ალბათობაა, რომ მოლეკულას აქვს ($v_i + \Delta v$)-ს ტოლი სიჩქარე, ამიტომ **H**-ის გამოსახულება შეიძლება ასე ჩავწეროთ:

$$H = -\sum_{i=1}^{N} p_i \ln p_i. \qquad (2.55)$$

მოცემული ფორმულა ემთხვევა (2.51) ფორმულას: ორივე შემთხვევაში **H** სიდიდე ახასიათებს სისტემის სხვადასხვაგვარობას.

(2.48) და (2.51) ფორმულების გამოყენებით შეგვიძლია განვსაზღვროთ შეტყობინებების **A** წყაროს ალფაბეტის სიჭარბე, რომელიც გვიჩვენებს თუ რამდენად რაციონალურად გამოიყენება მოცემული ალფაბეტი:

$$D = [H_{\max}(A) - H(A)] / [H_{\max}(A)], \qquad (2.56)$$

სადაც $H_{\max}(A)$ არის (2.47) ფორმულის დახმარებით გამოთვლილი მაქსიმალური შესაძლო ენტროპია, ხოლო $H(A)$ – წყაროს ენტროპია, გამოთვლილი (2.52) ფორმულის დახმარებით.

მაგალითისათვის გამოვითვალოთ ინგლისური ტექსტის მაფორმირებელი ინფორმაციის წყაროს **ენტროპია** და **სიჭარბე.**

ცხრილი 2.14. ინგლისურ ალფაბეტში შემავალი ასოების გამოყენების ალბათობები

ა ს ო	ალბათობა	ა ს ო	ალბათობა	ა ს ო	ალბათობა
პრობელი	0,2	**H**	0,047	**W**	0,012
E	0,105	**D**	0,035	**G**	0,011
T	0,072	**L**	0,028	**B**	0,010
O	0,065	**C**	0,023	**V**	0,008
A	0,063	**F**	0,023	**K**	0,003
N	0,058	**U**	0,023	**X**	0,001
I	0,055	**M**	0,021	**J**	0,001
R	0,052	**P**	0,018	**Q**	0,001
S	0,052	**Y**	0,012	**Z**	0,001

დასმული ამოცანის გადასაწყვეტად საჭიროა გამოვიყენოთ ინგლისურ ტექსტში ინგლისური ენის ალფაბეტში შემავალი ასოების გამოყენების ალბათობები. ისინი დადგენილია ექსპერიმენტულად და (2.14) ცხრილშია მოცემული.

(2.51) ფორმულით გამოვითვალოთ ინგლისური **A** ალფაბეტის **H(A)** ენტროპია:

$$
\begin{aligned}
\mathbf{H(A)} = & -\mathbf{0{,}2\log_2 0{,}2} - \mathbf{0{,}105\log_2 0{,}105} - \mathbf{0{,}072\log_2 0{,}072} - \mathbf{0{,}065\log_2 0{,}065} - \\
& -\mathbf{0{,}063\log_2 0{,}063} - \mathbf{0{,}058\log_2 0{,}058} - \mathbf{0{,}055\log_2 0{,}055} - \mathbf{0{,}052\log_2 0{,}052} - \\
& -\mathbf{0{,}052\log_2 0{,}052} - \mathbf{0{,}047\log_2 0{,}047} - \mathbf{0{,}035\log_2 0{,}035} - \mathbf{0{,}028\log_2 0{,}028} - \\
& -\mathbf{0{,}023\log_2 0{,}023} - \mathbf{0{,}023\log_2 0{,}023} - \mathbf{0{,}023\log_2 0{,}023} - \mathbf{0{,}021\log_2 0{,}021} - \\
& -\mathbf{0{,}018\log_2 0{,}018} - \mathbf{0{,}012\log_2 0{,}012} - \mathbf{0{,}012\log_2 0{,}012} - \mathbf{0{,}011\log_2 0{,}011} - \\
& -\mathbf{0{,}010\log_2 0{,}010} - \mathbf{0{,}008\log_2 0{,}008} - \mathbf{0{,}003\log_2 0{,}003} - \mathbf{0{,}001\log_2 0{,}001} - \\
& -\mathbf{0{,}001\log_2 0{,}001} - \mathbf{0{,}001\log_2 0{,}001} - \mathbf{0{,}001\log_2 0{,}001} = \\
& = \mathbf{4{,}03} \text{ (ბიტი/სიმბოლო)}.
\end{aligned}
$$

(2.47) ფორმულით ვიპოვოთ ინგლისური **A** ალფაბეტის მაქსიმალური $\mathbf{H_{max}(A)}$ ენტროპიის მნიშვნელობა (მივიღებთ იმ დაშვებით, რომ ალფაბტში შემავალი ყველა ასოს ალბათობა ერთმანეთის ტოლია და $1/27 \cong \mathbf{0{,}037}$-ს უდრის)

$$\mathbf{H_{max}(A) = \log_2 27 = 4{,}75} \text{ (ბიტი/სიმბოლო)}.$$

ინფორმაციის განხილული წყაროს სიჭარბე ვიპოვოთ (2.56) ფორმულით:

$$\mathbf{D(A) = (4{,}75\text{-}4{,}03)/0{,}75 = 0{,}15.}$$

2.12. ინფორმაციის რაოდენობის გაზომვის საკითხისათვის

“მეცნიერება იწყება იქ, სადაც იწყება გაზომვები.”
დ.ი. მენდელეევი (1834 — 1907)

კომპიუტერულ სისტემებს შეისწავლის ინფორმატიკა. პირველ თავში მოყვანილი განსაზღვრების თანახმად, იგი ფუნდამენტურ საბუნებისმეტყველო მეცნიერებას წარმოადგენს; მეცნიერების ერთ-ერთ დარგად ინფორმატიკის მიჩნევა იმდენად უდავოა, რომ ამერიკაში მას **“კომპიუტერულ მეცნიერებასაც”** (**«computer science»**) უწოდებენ. მეცნიერება კი ცნობილი რუსი მეცნიერის **მენდელეევის** თანახმად, იწყება იქ, სადაც იწყება გაზომვები. უფრო მეტიც, [31]-ის თანახმად გაზომვა წარმოადგენს მიმართულებას, რომლისკენაც მიდის განვითარება, გაფართოება, ევოლუცია და ამდენად, მას არა მარტო წმინდა მეცნიერული, არამედ დიდი პრაქტიკული მნიშვნელობაც აქვს.

გაზომვის მიზანი სპეციალური ტექნიკური საშუალებების დახმარებით ჩატარებული ცდით ფიზიკური სიდიდის რაოდენობითი მნიშვნელობის პოვნაა და ზოგადად გაზომვა ასე განისაზღვრება:

გაზომვა ეწოდება გამზომი საშუალებების დახმარებით შესრულებულ მოქმედებების ერთობლიობას, რომლის მიზანია ნაპოვნი იქნეს **გაზომვის მიღებულ ერთეულებში** გამოსახული **გასაზომი სიდიდის** მნიშვნელობა.

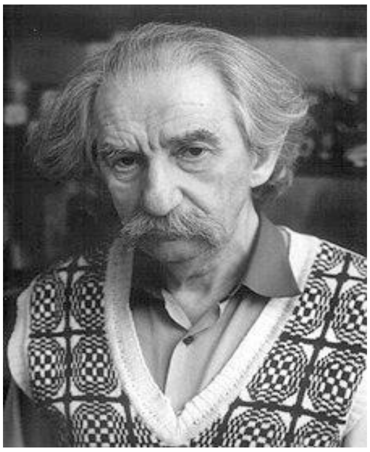
ფორმულირებული განსაზღვრების თანახმად, გაზომვის პროცესის რეალიზებისათვის აუცილებელია წინასწარ იქნეს დადგენილი:

- გასაზომი სიდიდე;
- გაზომვის მიღებული ერთეულები.

2 **ინფორმატიკა** დისციპლინათაშორისი მეცნიერებაა, რომლის კვლევის საგანი არის **ინფორმაცია.** იგი ინფორმაციის ანალიზით, შეკრებით, კლასიფიცირებით, მანიპულირებით, შენახვით, ამოღებით, გადაადგილებითა და სივრცეში გადაცემით არის დაკავებული. აღნიშნულის გამო ზოგიერთი ავტორი მას **ინფორმაციის მეცნიერებასაც** (**«information science»**) უწოდებს და ცხადია, რომ სწორედ ინფორმაციაა ის სპეციფიკური სიდიდე, რომლის გაზომვაც წარმოადგენს ინფორმატიკის, როგორც მეცნიერების ერთ-ერთ ძირითად ამოცანას. რაც შეეხება ამ სიდიდის **გასაზომად მიღებულ ერთეულებს**, მათი დადგენის სამი ასპექტი არსებობს, რომელთა მოკლედ განხილვა შეადგენს მოცემული პარაგრაფის მიზანს.

3 ინფორმაციაზე გარკვეული ოპერაციების ჩასატარებლად მას **შეტყობინების** სახით წარმოადგენენ. შეტყობინება გარკვეული ნიშნებისა და სიმბოლოების ერთობლიობაა, ხოლო ასეთ ერთობლიობას **სემიოტიკად (სემიოლოგიად)** წოდებული მეცნიერება შეისწავლის.

სემიოტიკა, ანუ **სემიოლოგია** (ბერძ. σημειωτική, - ‘‘ნიშანი’’) ეწოდება ნიშნებისა და ამ ნიშნებისაგან შედგენილ სისტემების შემსწავლელ მეცნიერებას.



ი.მ.ლოტმანი (1922-1993)

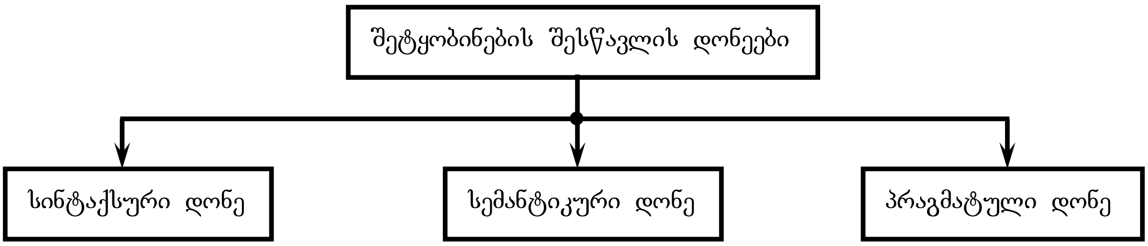
ჩ.უ. მორისი (1901-1979)

სემიოტიკის ძირითადი საკითხები საბჭოთა მეცნიერის **იური მიხეილის ძე ლოტმანისა** და ამერიკელი მეცნიერის **ჩარლზ უილიამს მორისის** ფუნდამენტურ ნაშრომებშია დამუშავებული. **ლოტმანის** აზრით სემიოტიკა წარმოადგენს მეცნიერებას ურთიერთობის პროცესში გამოყენებული საკომუნიკაციო სისტემებისა და ნიშნების შესახებ, ხოლო **მორისი** მას **მეტამეცნიერებად** მიიჩნევს.

მეტამეცნიერება (ძვ. ბერძ. **μετὰ** - ‘‘შორის’’, ‘‘შემდგომ’’) ითვლება უნივერსალურ მეცნიერებად, რომლის მიზანი სხვადასხვა მეცნიერებათა საერთო ენის - ე.წ. **მეტაენის** საშუალებით აღნიშნული მეცნიერებების დასაბუთება და შესწავლაა.

4

შეტყობინება, როგორც ნიშნების ერთობლიობა, **სემიოტიკის** თვალსაზრისით შეისწავლება სინტაქსურ, სემანტიკურ და პრაგმატულ დონეებზე (**ნახ.2.31)**. ამათგან, სინტაქსურ დონეზე შეისწავლება შეტყობინების შინაგანი, ხოლო სემანტიკურ და პრაგმატულ დონეებზე – გარეგანი თვისებები. მოკლედ განვიხილოთ თითოეული მათგანი.



ნახ.2.31. შეტყობინების შესწავლის დონეები

● **სინტაქსურ** (ბერძ. **syntaxis** -"თანადება, თანაწყობა, დაკავშირება"**) დონეზე** განიხილება შეტყობინების შემადგენელ ნიშნებს შორის არსებული თანაფარდობები, რომელთა ძალითაც არის ფორმირებული აღნიშნული შეტყობინების სტრუქტურა;

● **სემანტიკურ** (ბერძ. **sëmantikos** – "აღმნიშვნელი") **დონეზე** განიხილება შეტყობინების აზრობრივი შინაარსი, ინფორმაციის წყაროსთან მისი დამოკიდებულება, კერძოდ ნიშნებსა და მათ მიერ აღნიშნულ საგნებს, მოქმედებებს, თვისებებს შორის არსებული თანაფარდობები;

● **პრაგმატულ** (ბერძ. **pragma**, ნათ. ბრ. **Pragmatos** – "საქმე", "მოქმედება") **დონეზე** განიხილება შეტყობინებასა და მის მიმღებს შორის არსებული დამოკიდებულებები, კერძოდ, განისაზღვრება თუ რა სამომხმარებლო ღირებულებისაა შეტყობინება მისი მიმღებისათვის.

ზემოთ ჩამოთვლილ თითოეულ დონეზე ინფორმაციის ანალიზის დროს წამოიჭრება ამ დონისათვის დამახასიათებელი სპეციფიკური პრობლემები. მოკლედ განვიხილოთ თითოეული მათგანი.

5

<u>სიტაქსურ დონეზე</u> წამოიჭრება **შეტყობინებებისა** და და მათი მატერიალური მზიდების – **სიგნალების** გადაცემასთან დაკავშირებული წმინდა ტექნიკური პრობლემები, რომელთა გადაჭრა აუცილებელია:

● მაქსიმალურად შესაძლებელ მაჩვენებლებთან ახლოს მდებარე მაჩვენებლების მქონე ახალი საინფორმაციო სისტემების ასაგებად;

● არსებული საინფორმაციო სისტემების გამოყენების ეფექტურობის ასამაღლებლად.

მიმღებისათვის შეტყობინებების, როგორც გარკვეული ნიშნების ერთობლიობების, მიტანის პრობლემების განხილვისას სინტაქსურ დონეზე მხედველობაში მიიღება:

- მზიდის სახე;
- ინფორმაციის წარმოდგენის ხერხი;
- შეტყობინებების გადაცემისა და დამუშავების სიჩქარეები;
- ინფორმაციის წარმოდგენისათვის გამოყენებული კოდების ზომები;
- აღნიშნული კოდების საიმედოობა და მათი გარდაქმნების სიზუსტე და ა.შ.

ზემოთ ჩამოთვლილი გასათვალისწინებელი საკითხების შინაარსი გვიჩვენებს, რომ აღნიშნულ დონეზე ხდება შეტყობინებების შინაარსისა და მათი მიზნობრივი დანიშნულებისაგან სრული **აბსტრაჰირება** (ლათ. **abstractio** – "მოცილება", "მოშორება" – რაიმე საგნის ან მოვლენის არაარსებითი ნიშნების უგულებელყოფა და არსებითი ნიშნების გამოყოფა აზროვნებაში). ამ დონეზე მხოლოდ სინტაქსური პოზიციებიდან განხილულ

ინფორმაციას ჩვეულებრივ **მონაცემებს** უწოდებენ, რადგან მისი აზრობრვი შინაარსი სრულიად უგულებელყოფილია.

ინფორმაციის თანამედროვე თეორია ძირითადად სწორედ ამ დონის პრობლემებს იკვლევს. კვლევის პროცესში იგი ეყრდნობა **"ინფორმაციის რაოდენობის"** ცნებას, რომელიც შეტყობინების ფორმირებისათვის ნიშნების გამოყენების სიხშირის ზომაა და არ ასახავს გადასაცემი შეტყობინების არც შინაარსსა და არც მნიშვნელობას. აღნიშნულის გამო ზოგჯერ ამბობენ რომ **ინფორმაციის თანამედროვე თეორია სინტაქსურ დონეზე იმყოფება** [7].

6

სემანტიკური დონის პრობლემები დაკავშირებულია:

- გადასაცემი ინფორმაციის აზრის ფორმალიზებასა და მის გათვალიწინებასთან;
- ობიექტის სახისა და თავად ობიექტის შესაბამისობის ხარისხის განსაზღვრასთან.

ამ დონეზე:

– გაანალიზდება ინფორმაციაში ასასახი ცნობები;
– განიხილება აზრობრივი კავშირები;
– ფორმირდება ცნებები და წარმოდგენები;
– მჟღავნდება ინფორმაციის აზრი, შინაარსი და ხდება მათი განზოგადება.

ამ დონის პრობლემები ძალიან რთულია, რადგან ინფორმაციის აზრობრივი შინაარსი რაიმე ენაზე წარმოდგენილი შეტყობინების სემანტიკაზე უფრო მეტად თავად ინფორმაციის მიმღებ სუბიექტზე (ობიექტზე) არის დამოკიდებული.

7

პრაგმატულ დონეზე დგინდება თუ რა სარგებელი მოუტანა მომხმარებელს ინფორმაციის მიღებამ და გამოყენებამ. ამ დონის პრობლემები განსაზღვრავს საკუთარი მიზნის მიღწევისათვის საჭირო გადაწყვეტილების გამოსამუშავებლად თუ რამდენად სასარგებლო და ფასეულია მომხმარებლისათვის მის მიერ გამოყებული ინფორმაცია.

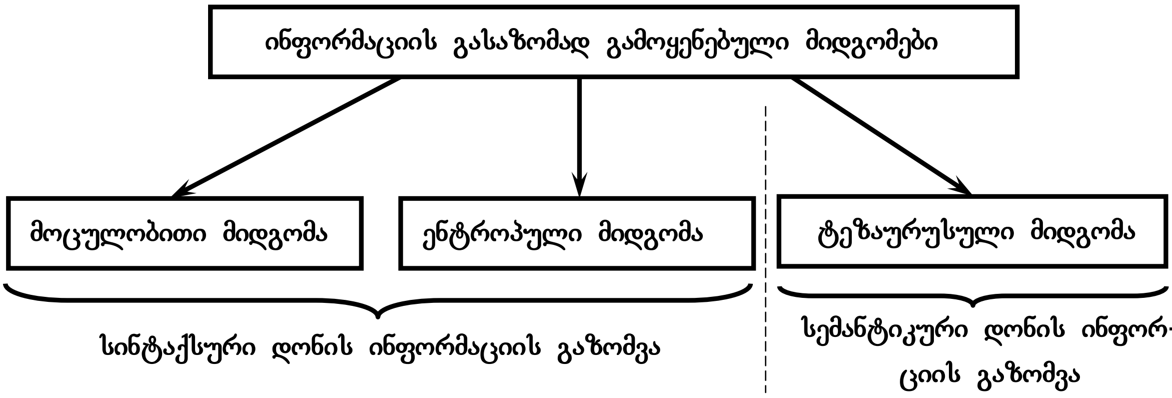
ამ დროს ძირითადი სირთულეა ის რომ, ერთი და იგივე ინფორმაციის სარგებლიანობა, **ჯერ ერთი**, სხვადასხვა მომხმარებლისათვის შეიძლება სხვადასხვაგვარი იყოს, და, **მეორეც**, ეს სარგებლიანობა მთელ რიგ ისეთ ფაქტორებზეა დამოკიდებული, როგორებიცაა, მაგალითად, ინფორმაციის დროულად მიღება და ამ ინფორმაციის გამოყენების შესაძლებ-ლობა.

ინფორმაციის მიტანის სიჩქარის მიმართ წაყენებულ მაღალ მოთხოვნებს ხშირად ის გარემოება განაპირობებს, რომ მმართველი ზემოქმედება **დროის რეალურ მასშტაბში**, უნდა სრულდებოდეს; ეს იმას ნიშნავს, რომ აღნიშნული ზემოქმედება ისეთივე სიჩქარით უნდა ხორციელდებოდეს, რა სიჩქარითაც იცვლება სამართი ობიექტების ან პროცესების მდგომარეობები. ინფორმაციის მიტანის ან მისი გამოყენების დაგვიანებას შეიძლება კატასტროფული შედეგები მოჰყვეს.

8

სინტაქსური დონის ინფორმაციის ზომები. აღნიშნული დონის ინფორმაციის რაოდენობრივი შეფასება ინფორმაციის შინაარსობრივ მხარესთან არაა დაკავშირებული და იგი ახასიათებს გაუპიროვნებელ ინფორმაციას, რომელიც ობიექტისადმი აზრობრივ დამოკიდებულებას არ გამოხატავს. ამიტომ მოცემული ზომა საშუალებას გვაძლევს შევაფასოთ თავისი ბუნებით ისეთ განსხვავებულ ობიექტებში ცირკულირებადი საინფორმაციო ნაკადები, როგორებიცაა კავშირგაბმულობის სისტემები, კომპიუტერები, მართვის სისტემები, ცოცხალი ორგანიზმის ნერვული სისტემები და ა.შ.

სინტაქსური ინფორმაციის გასაზომად, როგორც **2.32** ნახაზზე ჩანს, გამოიყენება მოცულობითი და ენტროპული მიდგომები. მოკლედ განვიხილოთ თითოეული მათგანი.



ნახ. 2.32. სინტაქსური და სემანტიკური დონის ინფორმაციათა შეფასების მიდგომები

მოცულობითი მიდგომის დროს შემოიტანება **ინფორმაციის (მონაცემების) მოცულობის** $V_{\text{მ}}$ პარამეტრი, ხოლო ენტროპული მიდგომის დროს – **ინფორმაციის რაოდენობის** I პარამეტრი. განვიხილოთ ორივე მათგანი.

ცხრ. 2.15. კომპიუტერული ინფორმაციის მოცულობის რაოდენობის ერთეულები

დასახელება	აღნიშ-ვნა	შ ე ნ ი შ ვ ნ ა
1 ბაიტი	**1 ბ**	8 ბიტი
1 კილობაიტი	**1 კბ**	2^{10} ბაიტი=1024 ბიტი ≅ 1 ათასი ბაიტი = 10^3 ბაიტი
1 მეგაბაიტი	1 **მბბ**	2^{20} ბაიტი=1024 კილობაიტი ≅ 1 მილიონი ბაიტი = 10^6 ბაიტი
1 გიგაბაიტი	**1 გბ**	2^{30} ბაიტი=1024 მეგაბაიტი ≅ 1 მილიარდი ბაიტი = 10^9 ბაიტი
1 ტერაბაიტი	**1 ტრბ**	2^{40} ბაიტი=1024 გიგაბაიტი ≅ 10^{12} ბაიტი (ტონაბაიტი)
1 პეტაბაიტი	**1 პტბ**	2^{50} ბაიტი=1024 ტერაბაიტი ≅ 10^{15} ბაიტი
1 ეკსაბაიტი	**1 ეკბ**	2^{60} ბაიტი=1024 პეტაბაიტი ≅ 10^{18} ბაიტი
1 ზეტაბაიტი	**1 ზტბ**	2^{70} ბაიტი=1024 ეკსაბაიტი ≅ 10^{21} ბაიტი
1 იოტაბაიტი	**1 იტბ**	2^{80} ბაიტი=1024 ზეტაბაიტი ≅ 10^{24} ბაიტი

- **ინფორმაციის (მონაცემების) მოცულობა** (**მოცულობითი მიდგომა**). საინფორმაციო პროცესების რეალიზების დროს ინფორმაცია გადაიცემა რაიმე ალფაბეტის ასოების **ერთობლიობისაგან** შემდგარი შეტყობინებების სახით. აღნიშნული **ერთობლიობისათვის** ყოველი ახალი სიმბოლოს (ასოს) მიმატება ზრდის ამ ერთობლიობით წარმოდგენილი ინფორმაციის რაოდენობას. ერთი ასოსაგან შემდგარი ერთობლიობით გამოსახულ შეტყობინებაში შემავალი ინფორმაციის რაოდენობას თუ **ინფორმაციის ერთეულად** მივიღებთ, მაშინ ნებისმიერ სხვა შეტყობინებაში არსებული ინფორმაციის (მონაცემების) $V_{\text{მ}}$ მოცულობა ამ შეტყობინებაში შემავალი სიმბოლოების (თანრიგების) რაოდენობის ტოლი იქნება. რადგან ერთი და იგივე ინფორმაცია მრავალი სხვადასხვა ხერხით (სხვა-

დასხვა ალფაბეტების გამოყენებით) შეიძლება გადაიცეს, ამიტომ შესაბამისად იცვლება ინფორმაციის (შეტყობინების) ზომის ერთეულები.

თვლის ათობით სისტემაში ერთ თანრიგს აქვს **10**-ის ტოლი წონა და შესაბამისად ინფორმაციის ზომის ერთეულს წარმოადგენს ***დიტი*** (ათობითი თანრიგი, იხილეთ წინა პარაგრაფი); ასეთ შემთხვევაში n–თანრიგიანი რიცხვის სახით წარმოდგენილ შეტყობინებაში შემავალი ინფორმაციის (მონაცემების) მოცულობა $V_{მ} = n$ დიტის ტოლი იქნება. მაგალითად, ***ოთხთანრიგიან* 1948** რიცხვი შეიცავს $V_{მ} = 4$ დიტის ტოლი მოცულობის ინფორმაციას.

თვლის ორობით სისტემაში ერთ თანრიგს აქვს **2**-ის ტოლი წონა და შესაბამისად ინფორმაციის ზომის ერთეულს წარმოადგენს ***ბიტი*** (**bit** (binary digit) – ორობითი თანრიგი, იხილეთ წინა პარაგრაფი); ასეთ შემთხვევაში n–თანრიგიანი რიცხვის სახით წარმოდგე-ნილ შეტყობინებაში შემავალი ინფორმაციის (მონაცემების) მოცულობა $V_{მ} = n$ ბიტის ტოლი იქნება. მაგალითად, ***რვა თანრიგიან*** ორობით **10110110** რიცხვს აქვს $V_{მ} = 8$ ბიტის ტოლი მოცულობის ინფორმაცია.

თანამედროვე კომპიუტერულ სისტემებში ბიტებით მონაცემების გაზომვისას ზომის მინიმალურ ერთეულად ფართოდ გამოიყენება ზომის გაფართოებული ერთეული, რომელიც რაოდენობრივად **8** ბიტს უდრის და მას ***ბაიტი*** ეწოდება. უფრო დიდი მოცულობების გამოყენებისას ზომის კიდევ უფრო დიდი ერთეულები (კილობაიტი, მეგაბაიტი და ა.შ.) გამოიყენება (**2.15** ცხრილი). ყურადღება უნდა მივაქციოთ იმ ფაქტს, რომ ორობითი (კომპიუტერული) ინფორმაციის სისტემაში ***"კილო"***, ***"მეგა"*** და ა.შ. წინდებულებიანი ზომის ერთეულების მისაღებად ძირითადი ერთეული მრავლდება არა $10^3 = 1000$-ზე, $10^6 = 1000000$-ზე და ა.შ., არამედ $2^{10} = 1024$-ზე, $10^{20} = 1048576$-ზე და ა.შ.

• <u>ინფორმაციის (მონაცემების) რაოდენობა</u> (ენტროპული მიდგომა).

სინტაქსურ დონეზე ინფორმაციის გაზომვის პრობლემის გადასაწყვეტად ინფორმატიკაში შემოტანილია ე.წ. ***საინფორმაციო ენტროპიის*** ცნება. გავარკვიოთ ამ ახალი ტერმინის არსი.

> საბუნებისმეტყველო მეცნიერებებში ***ენტროპია*** მრავალი ელემენტისაგან შემდგარი სისტემის უწესრიგობის ზომად არის მიჩნეული. ასეთი მიდგომის დროს რაც უფრო მაღალია აღნიშნული უწესრიგობა, მით უფრო მეტია ენტროპია და პირიქით, სისტემის მოწესრიგების კვალობაზე იგი მცირდება. აბსოლუტურად მოწესრიგებული სისტემის ენტროპია ნულის ტოლია.

საინფორმაციო ენტროპია გვიჩვენებს შესასწავლი ობიექტის ან პროცესის შესახებ ცოდნის არასრულყოფილობას, განუსაზღვრელობას. რაც უფრო ნაკლებად ვიცნობთ აღნიშნულ ობიექტს თუ პროცესს, მით უფრო არასრულყოფილი და განუსაზღვრელი ცოდნა გვაქვს მის შესახებ, ე.ი მით უფრო მაღალია ***საინფორმაციო ენტროპია***. ობიექტის ან პროცესის შესასწავლად აუცილებელია მისგან გარკვეული სახის შეტყობინება მივიღოთ და ამ თვალსაზრისით ორივე მათგანი შეიძლება ***შეტყობინების წყაროდ*** განვიხილოთ.

> ***საინფორმაციო ენტროპია*** წარმოადგენს შეტყობინების წყაროს განუსაზღვრელობას, ***განუსაზღვრელობად*** კი ითვლება ის ფაქტი, თუ რამდენად უცნობია ინფორმაციის მიმღებისათვის (დამკვირვებლისთვის) ინფორმაციის მოცემული წყარო (სისტემა, ობიექტი).

შეტყობინების წყაროდან მიღებულ შეტყობინებებს აღნიშნული წყაროს შესახებ ჩვენთან მოაქვს ინფორმაციის რაღაც I რაოდენობა. ამ დროს შეიძლება ადგილი ჰქონდეს შემდეგ ორი შემთხვევას:

1. ***მოტანილი ინფორმაცია ჩვენთვის ნაცნობია***. ასეთ შემთხვევაში წყაროს შესახებ მისგან ვერაფერ ახალს ვერ ვიგებთ; ასეთ შემთხვევაში იგი უცვლელად ტოვებს წყაროს საინფორმაციო ენტროპიას, ანუ ვერ ზრდის (სრულყოფილს ვერ ხდის) წყაროს შესახებ

ჩვენს ცოდნას; ამიტომ ინფორმაციის მიღებული I რაოდენობა ნულის ტოლად მიიჩნევა, ე.ი. $I=0$.

2. **მოტანილი ინფორმაცია ჩვენთვის ახალია.** ასეთ შემთხვევაში რაღაც უცნობს ვიგებთ წყაროს შესახებ. ამით მცირდება წყაროს საინფორმაციო ენტროპია ანუ იზრდება (სრულყოფილი ხდება) წყაროს შესახებ ჩვენი ცოდნა; ამიტომ ინფორმაციის მიღებულ I რაოდენობას აქვს 0-ზე მეტი რაოდენობა, ე.ი. $I > 0$. **რაც უფრო მეტია მოტანილი ინფორმაციის რაოდენობა, მით უფრო მეტად მცირდება შეტყობინების წყაროს საინფორმაციო ენტროპია (განუსაზღვრელობა).** აღნიშნულიდან გამომდინარე, სწორედ ენტროპიის შემცირების ფაქტი შეიძლება გამოვიყენოთ ინფორმაციის რაოდენობრივი შეფასებისათვის და დავასკვნათ შემდეგი:

ინფორმაციის I რაოდენობა წარმოადგენს შესასწავლი ობიექტის (პროცესის) საინფორმაციო ენტროპიის (განუსაზღვრელობის) შემცირების ზომას.

შეტყობინება რაც უფრო მეტად ამცირებს შეტყობინების წყაროს საინფორმაციო ენტროპიას (განუსაზღვრელობას), მით უფრო მეტ I ინფორმაციას შეიცავს იგი და, პირიქით, თუ უცვლელად ტოვებს აღნიშნულ ენტროპიას, მაშინ იგი ფაქტობრივად არ შეიცავს ინფორმაციას, ანუ $I=0$.

მგვარად, **ენტროპიული მიდგომის** დროს ინფორმაციად ითვლება რაიმე პროცესის (ცდის, გაზომვის და ა.შ.) მსვლელობაში გამქრალი განუსაზღვრელობის რაოდენობრივი სიდიდე. ამ დროს განუსაზღვრელობის ზომად შემოიტანება H ენტროპია და ინფორმაციის რაოდენობა გამოითვლება ფორმულით:

$$I = H_{apr} - H_{aps}\,, \qquad (2.57)$$

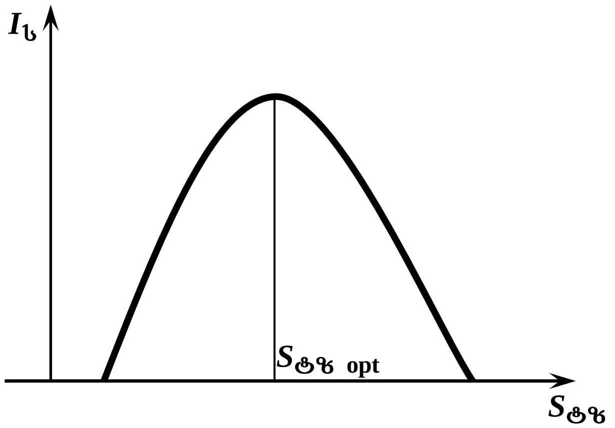
სადაც H_{apr} არის შესასწავლი სისტემის ან პროცესის მდგომარეობის შესახებ **აპრიორული** (შეტყობინების მიღებამდელი) **ენტროპია**, ხოლო H_{aps} – **აპოსტერიორული** (შეტყობინების მიღების შემდგომი) **ენტროპია.**

ცდის მსვლელობის დროს, როდესაც არსებული განუსაზღვრელობა მოიხსნება (ე.ი. როდესაც $H_{aps} = 0$), მაშინ მიღებული ინფორმაციის რაოდენობა საწყის ენტროპიას დაემთხვევა $I = H_{apr}$.-ს ტოლი გახდება.

9 **<u>სემანტიკური დონის ინფორმაციის ზომები.</u>** სემანტიკურ ინფორმაციად ითვლება მომხმარებლის მიერ შეტყობინებიდან გამოტანილი აზრობრივი შინაარსი. მის გასაზომად საყოველთაოდ მიღებული რაოდენობრივი ზომები დღეისათვის არ არსებობს. ინფორმაციის სემანტიკურ შესაფასებლად გამოყენებულ ცნობილ მიდგომებს შორის ყველაზე მეტადაა გავრცელებული **ტეზაურუსული მიდგომა**; ამ დროს მიღებულ შეტყობინებაში არსებული აზრი ფასდება მიმღების ტეზაურუსთან მისი შეთანადების გზით, მიმღების უნარით გაიგოს და აღიქვას მიღებული შეტყობინება. მართლაც, იმისათვის რომ გაიგოს და გამოიყენოს მიღებული ინფორმაცია, მიმღებს უნდა გააჩნდეს ცოდნის გარკვეული მარაგი. საგნის სრული უცოდინარობა მას საშუალებას არ მისცემს საგნის შესახებ მიღებული შეტყობინებიდან გამოიტანოს სასარგებლო ინფორმაცია. რაც უფრო მეტი იცის საგნის შესახებ, მით უფრო მეტი სასარგებლო ინფორმაციის გამოტანა შეუძლია მიმღებს შეტყობინებიდან.

ცნობებსა და ცოდნას, რომლებიც მიმღებს (მომხმარებელს) აქვს ინფორმაციის წყაროს შესახებ, **მომხმარებლის ტეზაურუსი** ეწოდება. შეტყობინებით მომხმარებელი ახალ ცნობებს ან ცოდნას თუ მიიღებს ინფორმაციის წყაროს შესახებ, იგი მათ შეიტანს საკუთარ ტეზაურუსში და გაზრდის მის მოცულობას. თუ შეტყობინებას მომხმარებლისათვის ახალი ცნობები ან ცოდნა არ მოაქვს, მაშინ იგი მათ საკუთარ ტეზაურუსში არ შეიტანს და ტეზაურუსის მოცულობა უცვლელი დარჩება.

შეტყობინებაში არსებული **სემანტიკური ინფორმაციის რაოდენობა** მისი მეშვეობით მომხმარებლის ტეზაურუსის ნაზრდის პროპორციულია.



ნახ.2.33. მომხმარებლის მიერ აღქმული სემანტიკური ინფორმაციის რაოდენობის დამოკიდებულება ტეზაურუსზე $I_ს=f(S_{თზ})$

ამგვარად, მიღებული შეტყობინებებიდან მომხმარებლის მიერ ამოღებული და შემდეგ საკუთარ ტეზაურუსში ჩართული სემანტიკური ინფორმაციის $I_ს$ რაოდენობა დამოკიდებულია ასეთი ინფორმაციის აღსაქმელად $S_{თზ}$ ტეზაურუსის მომზადების (სისრულის) ხარისხზე (ნახაზი **2.33**). ტეზაურუსი თუ საკმარისად არ არის განვითარებული, მაშინ მომხმარებელს შეტყობინებიდან მცირე რაოდენობის სემანტიკური ინფორმაცია ამოაქვს, ხოლო ზღვრულ შემთხვევაში, როდესაც $S_{თზ}=0$-ს (მიმღებს არ გააჩნია საწყისი ცოდნა), იგი საერთოდ ვერ აღიქვამს მოსულ შეტყობინებას (არ შეუძლია გაიგოს იგი). საკმაოდ განვითარებული ტეზაურუსის დროსაც მომხმარებელი მცირე რაოდენობის სემანტიკურ ინფორმაციას აღიქვამს (მოცემული შეტყობინების ზემოქმედებით ტეზაურუსი მცირედ იცვლება); ზღვრულ შემთხვევაში, როდესაც $S_{თზ} \longrightarrow \infty$, მომხმარებელმა ‘‘ყველაფერი იცის’’ და შეტყობინებიდან ამოღებული შეტყობინება მას არ სჭირდება და შეტყობინება შეიცავს ნულის ტოლ სემანტიკურ ინფორმაციას.

მიმღებს შეტყობინებიდან მაქსიმალური რაოდენობის სემანტიკური ინფორმაცია მაშინ ამოაქვს, როდესაც მისი საკუთარი ტეზაურუსი შეთანხმებულია შეტყობინებაში არსებული აზრობრივ შინაარსთან ($S_{თზ} = S_{თზ\ opt}$). ასეთ შემთხვევაში შემომავალი ინფორმაცია გასაგებია მიმღებისათვის, იგი შეიცავს შემდგომში საკუთარ ტეზაურუსში შესატან მისთვის ადრე უცნობ მონაცემებს.

აღნიშნულიდან გამომდინარე, შეტყობინებაში არსებული სემანტიკური ინფორმაციის რაოდენობა ფარდობითი სიდიდეა. ერთსა და იგივე შეტყობინებას აზრობრივი მნიშვნელობა შეიძლება ჰქონდეს უფრო განვითარებული ტეზაურუსის მქონე (**კომპენტენტური**) მიმღებისათვის და უაზრო იყოს არასაკმარისად განვითარებული ტეზაურუსის მქონე (**არაკომპენტენტური**) მომხმარებლისათვის.

სემანტიკური ინფორმაციის რაოდენობის ფარდობით ზომად გამოიყენება **შინაარსიანობის C კოეფიციენტი,** რომელიც განისაზღვრება როგორც სემანტიკური ინფორმაციის $I_ს$ რაოდენობის ფარდობა მის $V_მ$ მოცულობასთან:

$$C = I_ს / V_მ \qquad (2.58)$$

ინფორმაციის სემანტიკური შეფასების კიდევ ერთ მიდგომას, რომელიც მეცნიერებამცოდნეობაში გამოიყენება, წარმოადგენს ის, რომ გასაანალიზებელ დოკუმენტში (შეტყობინებაში, პუბლიკაციაში) არსებული ინფორმაციის სემანტიკურ ღირებულებად მიიღება სხვა დოკუმენტებში მისი დამოწმებების (მასზე მითითებების) რაოდენობა. კონკრეტული მაჩვენებლები ფორმირდება სხვადასხვა ამონაკრებებში დამოწმებათა (მითითებათა) რაოდენობების სტატასტიკური დამუშავების საფუძველზე).

10 **<u>პრაგმატული დონის ინფორმაციის ზომა</u>** განსაზღვრავს თუ რამდენად სასარგებლოა (ღირებულია) ინფორმაცია მომხმარებლისათვის მის წინ დასმული მიზნის მისაღწევად.


ა.ა. ხარკევიჩი (1904-1965)

ერთ-ერთი პირველთაგანი, რომელმაც პრაგმატული დონის ინფორმაციის შეფასების პრობლემაზე დაიწყო მუშაობა, იყო საბჭოთა მეცნიერი **ალექსანდრე ალექსანდრეს ძე ხარკევიჩი.**

მან შემოგვთავაზა **ინფორმაციის ღირებულების ზომად** მიგვეღო დასახული მიზნის მისაღწევად აუცილებელი ინფორმაციის რაოდენობა, ე.ი. გამოგვეთვალა მიზნის მიღწევის ალბათობის ნაზრდი. ასე მაგალითად, თუ ინფორმაციის მიღებამდე მიზნის მიღწევის ალბათობა p_0–ის ტოლი იყო და ინფორმაციის მიღების შემდეგ იგი p_1–ის ტოლი გახდა, მაშინ ინფორმაციის ღირებულებად უნდა მივიღოთ p_1/p_0 ფარდობის ლოგარითმი:

$$I = \log_2 p_1 - \log_2 p_0 = \log_2 \frac{p_1}{p_0} \qquad (2.59)$$

ამგვარად, ამ დროს **ინფორმაციის ღირებულება** იზომება **ინფორმაციის ერთეულებში,** მოცემულ შემთხვევაში – **ბიტებში.**

(2.59) გამოსახულებით გამოთვლილ სიდიდეს შეიძლება ჰქონდეს უარყოფითი ნიშანი, ე.ი. შეიძლება არსებობდეს **უარყოფითი ღირებულების ინფორმაციაც**. ასეთი ინფორმაცია **უარყოფით ინფორმაციად** ითვლება; იგი ზრდის საწყის განუსაზღვრელობას (ენტროპიას) და ამცირებს მიზნის მიღწევის ალბათობას, რის გამოც მას **დეზინფორმაცია** ეწოდება.

აპოსტერიორი *(ლათ.* ***a posteriori*** *– შემდგომიდან) – ცდაზე, ფაქტებზე, ცდიდან გამომდინარე;*

აპრიორი *(ლათ.* ***a priori*** *– წინმდებარედან) – ცდისგან დამოუკიდებელი, იმთავითვე არსებული;*

თეზაურუსი *(ბერძ.* ***θησαυρός*** *– "საგანძური")* ***ფართო გაგებით*** *წარმოადგენს ცოდნის ერთობლიობას, რომელიც გააჩნია მომხმარებელს (სისტემას), ხოლო* ***ვიწრო გაგებით*** *იგი მოცემული ენის სიტყვებსა და სხვა აზრობრივ ელემენტებს შორის არსებული სემანტიკური კავშირების ამსახავი და ლექსიკონის სახით ფორმირებული ცოდნის მარაგია;* ***განმარტებითი ლექსიკონისაგან*** *განსხვავებით, თეზაურუსი საშუალებას გვაძლევს აზრი გავიგოთ არა მარტო განსაზღვრებების დახმარებით, არამედ სხვა ცნებებთან და მათ ჯგუფებთან სიტყვის თანაფარდობითობის დამყარების გზითაც; ამ უკანასკნელის შემწეობით თეზაურასი* ***ხელოვნური ინტელექტის სისტემათა*** *მონაცემების ბაზის შესავსებადაც შეიძლება გამოვიყენოთ.*

III თ ა ვ ი
არითმეტიკა კომპიუტერებისათვის

“მათემატიკა მეცნიერების დედოფალია, ხოლო არითმეტიკა – მათემატიკის დედოფალი”
კ.ფ. გაუსი (1777-1855)

3.1 არითმეტიკული ოპერაციების შესრულება თვლის პოზიციური სისტემების გამოყენების დროს

1 პირველყოფილ საზოგადოებიდან დაწყებული დღევანდელი დღით დამთავრებული ადამიანი განუწყვეტლივ მიისწრაფვის ცივილიზებული სრულყოფილებისაკენ. ამ მისწრაფების დამადასტურებელ დიდებულ ლიტერატურულ ძეგლს წარმოადგენს უკვდავი ლეგენდა **პრომეთეს** შესახებ. პრომეთე განათლების სახეა. იგი წარმოადგენს ადამიანის გონისათვის დამახასიათებელი წინასწარჭვრეტის სიმბოლოს, რომლის თანახმადაც წიგნისა და სხვადასხვა გამოთვლებისათვის აუცილებელი **“რიცხვთა თვლის”** გარეშე კაცობრიობა თავს ვერ დააღწევს სიბნელეს. სწორედ ეს აქვს ხაზგასმული **ესქილეს** (ჩვ. წ.აღრიცხვამდე **525-456**) პოემა “მიჯაჭვულ პრომოთეში” (მთარგმნელი **ა.ქუთათელი**), სადაც პრომეთე აღნიშნავს:

ჰ.ფ.ფიუგერი (1751-1818) “**პრომეთე**”

“კაცნი მხედველნი, - უსინათლოდ
მაინც რჩებოდნენ,
ისმენდნენ, მაგრამ არ ესმოდათ
და მიათრევდნენ
რაღაც ნისლიან, ძილისმომგვრელ
დღეთა გრძელ ჭაპანს.
. . . . .
და მე ვუჩვენე ცის მნათობთა
ჩასვლა, ამოსვლა.
ვასწავლე თანაც ყველა სიბრძნის
სათავე – **<u>წიგნი,</u>**
<u>რიცხვთ ანგარიში</u> და შთავნ-
ერგე შთაგონებული
მეხსიერება – ეგ მუზათა
დედა მშობელი”

ესქილესეული სიბრძნე დიდებულად გამოხატა გერმანელმა **მხატვარმა**, კლასიციზმის მიმდევარმა **ჰენრიხ ფრიდრიხ ფიუგერმა** თავის შედევრში “პრომოთე”, რომელშიც ნაჩვენებია ბნელეთით გარემოცულ ადამიანს თუ როგორ უცისკროვნებს ცხოვრებას წიგნისა და მეცნიერების სიყვარულის სიმბოლო – პრომეთე.

"ანგარიში და გამოთვლა – თავში წესრიგის საფუძველია", გვასწავლის ცნობილი შვეიცარიელი პედაგოგი **ი.ჰ.პესტალოცი (1746-1827).**

სწორედ არითმეტიკული ოპერაციები მიეკუთვნება იმ "ჯადოსნურ" ოპერაციათა რიცხვს, რომელთა შესრულების წყალობითაც დაიმკვიდრა კომპიუტერმა გონიერი ხელოვნური მოწყობილობის სახელი. ამიტომ რამდენადმე დეტალურად განვიხილოთ აღნიშნულ ოპერაციებთან დაკავშირებული ზოგიერთი საკვანძო საკითხი.

კომპიუტერულ ტექნოლოგიაში, აგრეთვე ავტომატიკისა და კავშირგაბმულობის მოწყობილობებში ძირითადად **თვლის ორობითი სისტემები** გამოიყენება; ამას განაპირობებს ის გარემოება, რომ თვლის სხვა სისტემებთან შედარებით ორობით სისტემებს გააჩნია მთელი რიგი უპირატესობები. კერძოდ:

- თვლის ორობითი სისტემის რეალიზებისათვის საკმარისია მხოლოდ ორი მდგრადი მდგომარეობის მქონე მოწყობილობა; ასეთი მდგრადი მდგომარეობის მაგალითებია მასალის (მაგნიტური ლენტის ან დისკის) დამაგნიტებულობა ან განმაგნიტულობა, კონდენსატორის დამუხტულობა ან განმუხტულობა, ძაბვის არსებობა ან არარსებობა, პერფობარათზე ნახვრეტის არსებობა ან არარსებობა და ა.შ.
- თვლის ორობითი სისტემის გამოყენების დროს ინფორმაცია უფრო საიმედოდ და დაბრკოლებამდგრადად შეიძლება იქნეს წარმოდგენილი;
- ინფორმაციის ლოგიკური გარდაქნებისათვის შესაძლებელია გამოყენებული იქნეს ჯორჯ ბულის მიერ დამუშავებული ლოგიკის ალგებრის (ბულის ალგებრის) მეთოდები;
- თვლის ორობითი სისტემის დროს ყველაზე მარტივადაა შესაძლებელი არითმეტიკული ოპერაციების შესრულება.

ცხრილი 3.1. ათობითი, ორობითი, რვაობითი და თექვსმეტობითი რიცხვების მაგალითები

ათობითი	ორობითი	რვაობით	თექვსმეტობითი
0	0 0 0 0 0	0	0
1	0 0 0 0 1	1	1
2	0 0 0 1 0	2	2
3	0 0 0 1 1	3	3
4	0 0 1 0 0	4	4
5	0 0 1 0 1	5	5
6	0 0 1 1 0	6	6
7	0 0 1 1 1	7	7
8	0 1 0 0 0	10	8
9	0 1 0 0 1	11	9
10	0 1 0 1 0	12	A
11	0 1 0 1 1	13	B
12	0 1 1 0 0	14	C
13	0 1 1 0 1	15	D
14	0 1 1 1 0	16	E
15	0 1 1 1 1	17	F
16	1 0 0 0 0	20	10
17	1 0 0 0 1	21	11
18	1 0 0 1 0	22	12
19	1 0 0 1 1	23	13
20	1 0 1 0 0	24	14

ორობითი სისტემის ნაკლია დიდი რიცხვების ჩასაწერად მნიშვნელოვანი რაოდენობის ციფრებიანი რიცხვების გამოყენების აუცილებლობა. კომპიუტერისათვის ეს ნაკლი **არაარსებითია,** მაგრამ არსებითია ორობითი რიცხვების გამომყენებელი ადამიანისათვის. ამიტომ ინფორმაციის "ხელით" კოდირების აუცილებლობის შემთხვევაში (რომელსაც ადგილი აქვს სამანქანო ენაზე პროგრამის შედგენისას), თვლის ორობითი სისტემის ნაცვლად, სასურველია გამოვიყენოთ **თვლის რვაობითი ან თექვსმეტობითი სისტემა**. ამას განაპირობებს შემდეგი ფაქტორები:

- რვაობითი და თექვსმეტობითი რიცხვები თითქმის ისევე ადვილად იკითხება, როგორც ათობითი რიცხვები;
- ორობით რიცხვებთან შედარებით რვაობითი და თექვსმეტობითი რიცხვების ჩასაწერად შესაბამისად **3**-ჯერ და **4**-ჯერ ნაკლები თანრიგებია საჭირო;
- ათობით რიცხვებთან შედარებით რვაობითი და თექვსმეტობითი რიცხვები გაცილებით მარტივად შეიძლება გარდავქმნათ ორობით რიცხვებად.

3.1. ცხრილში მოცემულია თვლის ათობითი, ორობითი, რვაობითი და თექვსმეტობითი სისტემის გამოყენებით ჩაწერილი რიცხვების მაგალითები;

> თვლის ორობითი, რვაობითი და თექვსმეტობითი სისტემების მეშვეობით წარმოდგენილი რიცხვები კომპიუტერული სისტემების გამოყენებით ინფორმაციის დამუშავების პროცესებში გამოსაყენებლადაა მოსახერხებელი, ამიტომ აღნიშნულ რიცხვებზე შესასრულებელი ოპერაციების შემსწავლელ **მათემატიკის ნაწილს** პირობითად **კომპიუტერებზე ორიენტირებული არითმეტიკა** (ძვ. ბერძ. **, ἀριθμός** –**"**რიცხვი"-დან მიღებული ტერმინი **ἀριθμητική**), ანუ "**არითმეტიკა კომპიუტერებისათვის"** ვუწოდოთ. თითოეული ჩვენთაგანისათვის ბავშვობიდანვე ნაცნობი **სასკოლო არითმეტიკა** მხოლოდ ათობით რიცხვებზე ჩასატარებელი ოპერაციების შესწავლითაა დაკავებული.

რვაობითი სისტემიდან ორობით სისტემაში რიცხვების გადასაყვანად რვაობითი რიცხვის თითოეული ციფრი უნდა შევცვალოთ მისი ეკვივალენტური ორობითი ტრიადით (სამი ბიტისაგან შემდგარი ორობითი რიცხვით) (**ნახ3.1,ა**);

ა $\quad 673{,}2_{(8)} = \underbrace{110}_{6}\ \underbrace{111}_{7}\ \underbrace{011}_{3}\ ,\ \underbrace{010}_{2}\ {}_{(2)}$

ბ $\quad 1B5{,}D_{(16)} = \underbrace{1}_{1}\ \underbrace{1011}_{11}\ \underbrace{0101}_{5}\ ,\ \underbrace{1101}_{13}\ {}_{(2)}$

გ $\quad 10110100{,}10111_{(2)} = \underbrace{10}_{2}\ \underbrace{110}_{6}\ \underbrace{100}_{4}\ ,\ \underbrace{101}_{5}\ \underbrace{110}_{6} = 264{,}56_{(8)}$

დ $\quad 10101001\ ,\ 10111_{(2)} = \underbrace{1010}_{A}\ \underbrace{1001}_{9}\ ,\ \underbrace{1011}_{B}\ \underbrace{1000}_{8}\ {}_{(2)} = A9{,}B8_{(16)}$

ნახ. 3.1 თვლის ერთ-ერთი სისტემიდან მეორე სისტემაში რიცხვების გადაყვანის წესები

თექვსმეტობითი სისტემიდან ორობით სისტემაში რიცხვების გადასაყვანად თექვსმეტობითი რიცხვის თითოეული ციფრი უნდა შევცვალოთ მისი ეკვივალენტური ორობითი ტეტრადით (ოთხი ბიტისაგან შემდგარი ორობითი რიცხვით) (**ნახ3.1,ბ**);

ორობითი სისტემიდან რვაობით სისტემაში რიცხვის გადასაყვანად ორობითი რიცხვი მძიმიდან მარცხნივ და მარჯვნივ უნდა დავყოთ ტრეადებად (სამეულებად) და თითოეული ტრეადა შევცვალოთ ეკვივალენტური რვაობითი ციფრით (**ნახ.3.1,გ**);

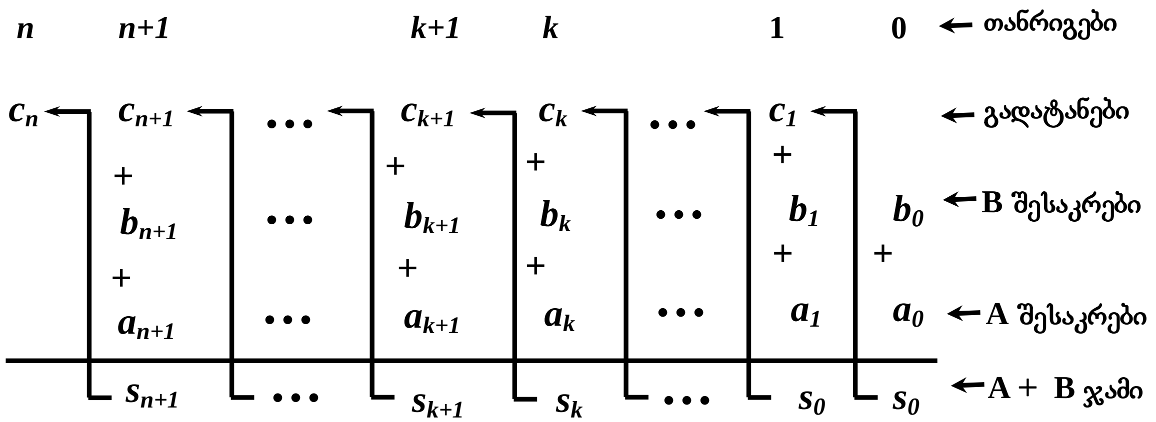
ორობითი სისტემიდან თექვსმეტობით სისტემაში რიცხვის გადასაყვანად ორობითი რიცხვი მძიმიდან მარცხნივ და მარჯვნივ უნდა დავყოთ ტეტრადებად (ოთხეულებად) და თითოეული ტეტრადა შევცვალოთ ეკვივალენტური თექვსმეტობითი ციფრით (**ნახ.3.1,დ**).

3 თვლის ორობით, რვაობით და თექვსმეტობით სიტემებში შეკრების, გამოკლების, გამრავლებისა და გაყოფის ოპერაციები თვლის ათობით სისტემაში შესასრულებელი ანალოგური ოპერაციების მსგავსად სრულდება, ოღონდ შეკრებისა და გამრავლების სპეციფიკური ცხრილის გამოყენებაა აუცილებელი. აღნიშნული ცხრილების განხილვამდე გავეცნოთ $\boldsymbol{n}$-თანრიგიანი დადებითი რიცხვების შეკრების ზოგად პრინციპს.

3.2 ნახაზზე მოყვანილია მთელი დადებითი $\boldsymbol{n}$-თანრიგიანი $\mathbf{A} = a_{(n-1)}\dots a_k\dots a_1\, a_0$ და $\mathbf{B} = b_{(n-1)}\dots b_k\dots b_1\, b_0$ რიცხვების შეკრების პრინციპის მაილუსტრირებელი სქემა; აღნიშნული პრინციპის თანახმად, რიცხვების შესაკრებად საჭიროა მათი თანრიგები შევკრიბოთ შემდეგი წესის დაცვით:

$$c_k + b_k + a_k = c_{k+1}\, s_k\,, \qquad (3.1)$$

c_k არის გადატანა k-1 თანრიგიდან k თანრიგში; b_k, a_k წარმოადგენს **B** და **A** შესაკრებთა k-ური თანრიგებს; s_k არის k-ური თანრიგების ჯამი, ხოლო c_{k+1} არის გადატანა k-1 თანრიგში; $k = 0;\ 1;\ 2\ ;\ \dots\ ;\ (n\text{-}1)$.



ნახ. 3.2. მრავალთანრიგიანი რიცხვების შეკრების წესები

ჯამის თანრიგები უმცროსი თანრიგიდან დაწყებული თანდათან ფორმირდება. ნულოვანი თანრიგისათვის ($k = 0$) გადატანა c_0 არ არსებობს, ამიტომ (3.1) გამოსახულება იღებს სახეს $b_0 + a_0 = c_1\, s_0$.

($\boldsymbol{n}$-1) თანრიგიდან $\boldsymbol{n}$ თანრიგში $c_n = \mathbf{1}$ გადატანის ფორმორებისას არასწორი შედეგი მიიღება, რადგან თანრიგობრივი ბადის გადავსების გამო აღნიშნული გადატანა იკარგება. აღნიშნული გადავსების არარსებობისათვის აუცილებელია სრულდებოდეს პირობა:

$$|\ \mathbf{A} + \mathbf{B}\ | < \mathbf{Q}^n, \qquad (3.2)$$

სადაც არის **Q** თვლის სისტემის ფუძე.

3.2 ცხრილში მოცემულია (3.1) ფორმულის მიხედვით ორობითი რიცხვების k-ური თანრიგების შეკრების ყველა შესაძლო ვარიანტი.

ცხრილი 3.2. ორობით რიცხვთა თანრიგების შეკრების შესაძლო ვარიანტები

შესასვლელი სიდიდეები			გამოსასვლელი სიდიდეები	
k-ური c_k თანრიგის გადატანა	**B** შესაკრების k-ური b_k თანრიგი	**A** შესაკრების k-ური a_k თანრიგი	$k+1$-თანრიგში c_{k+1} გადატანა	k-ური თანრიგის c_{k+1} ჯამი
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

შ ე ჯ ა მ ე ბ ა. თვლის ორობითი, რვაობითი და თექვსმეტობითი სისტემებისათვის აგებული შეკრების ცხრილები **3.2** ნახაზზეა მოცემული. იგი გამოიყენება **3.1** ნახაზზე ნაჩვენები წესების მიხედვით რიცხვების თანრიგობრივი შეკრების დროს.

3.3 ნახაზზე მოყვანილი ცხრილების გამოყენებით განვიხილოთ ორობით, რვაობით და თექვსმეტობით სისტემებში რიცხვების შეკრების მაგალითები (**3.1** და **3.2** მაგალითი).

მ ა გ ა ლ ი თ ი 3.1.

შევკრიბეთ ათობითი რიცხვები 141,5 და 59,75 თვლის ორობით, რვაობით და თექვსმეტობით სისტემებში

პასუხი: $141{,}5+59{,}75 = 201{,}25_{(10)} = 11001001{,}01_{(2)} = 311{,}2_{(8)} = C9{,}4_{(16)}$

ო რ ო ბ ი თ ი ს ი ს ტ ე მ ა

$10001101{,}1_{(2)} + 111011{,}11_{(2)}$

ა

$$\begin{array}{r} 1\,1\,1\,1\,1\,1\,1 \leftarrow \text{გადატანა} \\ +\;1\,0\,0\,0\,1\,1\,0\,1{,}\,1 \\ 1\,1\,1\,0\,1\,1{,}\,1\,1 \\ \hline 1\,1\,0\,0\,1\,0\,0\,1{,}\,0\,1 \end{array}$$

$11001001{,}01_{(2)} = 2^7+2^6+2^3+2^0+2^{-2} = 201{,}25_{(10)}$

რვაობითი სისტემა

$215{,}4_{(8)} + 73{,}6_{(8)}$

ბ

$$\begin{array}{r} 1\;1\;1 \leftarrow \text{გადატანა} \\ +\;2\;1\;5{,}\;4 \\ 7\;3{,}\;6 \\ \hline 3\;1\;1{,}\;2 \end{array}$$

$311{,}2_{(8)} = 3\cdot8^2 + 1\cdot8^1 + 1\cdot8^0 + 2\cdot8^{-1} = 201{,}25_{(10)}$

თექვსმეტობითი სისტემა

$8D{,}8_{(8)} + 3B{,}C_{(16)}$

გ

$$\begin{array}{r} 1\;1 \leftarrow \text{გადატანა} \\ +\;8\;D{,}\;8 \\ 3\;B{,}\;C \\ \hline C\;9{,}\;4 \end{array}$$

$C9{,}4_{(16)} = 12\cdot16^1 + 9\cdot16^0 + 4\cdot16^{-1} = 201{,}25_{(10)}$

ა)

ორობით სისტემაში
შეკრების ცხრილი

+	0	1
0	0	1
1	1	10

ბ)

რვაობით სისტემაში შეკრების ცხრილი

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	10
2	2	3	4	5	6	7	10	11
3	3	4	5	6	7	10	11	12
4	4	5	6	7	10	11	12	13
5	5	6	7	10	11	12	13	14
6	6	7	10	11	12	13	14	15
7	7	10	11	12	13	14	15	16

ბ)

თექვსმეტობით სისტემაში შეკრების ცხრილი

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

ნახ.3.3. შეკრების ცხრილები თვლის 2-ობითი (ა), 8-ობითი (ბ), და 16-ობითი (გ) სისტემებისათვის

მ ა გ ა ლ ი თ ი 3.2.

შეკრიბეთ ათობითი 15 და 6 რიცხვები თვლის ორობით, რვაობით და თექვსმეტობით სისტემებში პასუხი: $15 + 6 = 21_{(10)} = 10101_{(2)} = 25_{(8)} = 15_{(16)}$

ორობითი სისტემა	რვაობითი სისტემა	თექვსმეტობითი სისტემა
$1\,1\,1\,1_{(2)} + 1\,1\,0_{(2)}$	$1\,7_{(8)} + 6_{(8)}$	$F_{(16)} + 6_{(16)}$
ა 1 1 1 გადატანა $\begin{array}{r} 1\,1\,1\,1 \\ +\;0\,1\,1\,0 \\ \hline 1\,0\,1\,0\,1 \end{array}$	ბ 1 ← გადატანა $\begin{array}{r} 17 \\ +\;6 \\ \hline 25 \end{array}$	გ 1 ← გადატანა $\begin{array}{r} F \\ +\;6 \\ \hline 15 \end{array}$
$10101_{(2)} = 2^4+2^2+2^0 =$ $= 16 + 4 + 1 = 21_{(10)}$	$25_{(8)} = 2\bullet 8^1 + 5\bullet 8^0 =$ $= 16 + 5 = 21_{(10)}$	$15_{(16)} = 1\bullet 16^1 + 5\bullet 16^0 =$ $= 16 + 5 = 21_{(10)}$

5

გამოკლება. გამოკლება წარმოადგენს შეკრების შებრუნებულ ოპერაციას, ამიტომ თვლის ორობითი, რვაობითი და თექვსმეტობითი სისტემებისათვის აგებული შეკრების ცხრილები (**იხ.ნახ.3.2**) შეიძლება გამოვიყენოთ გამოკლების ოპერაციის შესრულების დროსაც. აღნიშნულ ცხრილებში მარცხენა განაპირა სვეტში ჩამოწერილია მაკლებები, ზედა განაპირა მწკრივში – სხვაობები, ხოლო აღნიშნულ სვეტსა და მწკრივს შორის არსებულ უჯრედებში ნაჩვენებია საკლებები. **3.2** ნახაზზე მოყვანილი ცხრილების გამოყენებით თვლის ორობით, რვაობით და თექვსმეტობით სისტემებში რიცხვების გამოკლების პრინციპი ილუსტრირებულია **3.3** მაგალითის, ხოლო მრავალთანრიგიანი ორობითი, რვაობითი და თექვსმეტობითი რიცხვების გამოკლების პროცესი ნაჩვენებია **3.4** მაგალითის სახით.

მ ა გ ა ლ ი თ ი 3.3.

$10_{(2)}$, $10_{(8)}$ და $10_{(16)}$ რიცხვებს გამოვაკლოთ ერთიანი

ორობითი სისტემა	რვაობითი სისტემა	თექვსმეტობითი სისტემა
ა სესხი ⟶ 1 $\begin{array}{r} 10 \\ -\;1 \\ \hline 1 \end{array}$	ბ სესხი ⟶ 1 $\begin{array}{r} 10 \\ -\;1 \\ \hline 7 \end{array}$	გ სესხი ⟶ 1 $\begin{array}{r} 10 \\ -\;1 \\ \hline F \end{array}$

მ ა გ ა ლ ი თ ი 3.4.

201,25 –ს გამოვაკლოთ 59,75 თვლის ორობით, რვაობით და თექვსმეტობით სისტემებში

პასუხი: $201,25_{(10)} - 59,75_{(10)} = 141,5_{(10)} = 10001101,1_{(2)} = 215,4_{(8)} = 8D,8_{(16)}$

ო რ ო ბ ი თ ი   ს ი ს ტ ე მ ა

$11001001,01_{(2)} - 111011,11_{(2)}$

ა

$$\begin{array}{r} 1\ \ \ 1\ \ \ 1 \leftarrow \text{სესხები} \\ 1\,1\,0\,0\,1\,0\,0\,1\,,\,0\,1 \\ -\ 0\,0\,1\,1\,1\,0\,1\,1\,,\,1\,1 \\ \hline 1\,0\,0\,0\,1\,1\,0\,1\,,\,1\,0 \end{array}$$

$10001101,1 = 2^7+2^3+2^2+2^0+2^{-1} = 141,5$

რვაობითი სისტემა

$311,2_{(8)} - 73,6_{(8)}$

ბ

$$\begin{array}{r} 1\,1\,1 \leftarrow \text{სესხები} \\ 3\,1\,1\,,\,2 \\ -\ 7\,3\,,\,6 \\ \hline 2\,1\,5\,,\,4 \end{array}$$

$215,4_{(8)} = 2\cdot8^2 + 1\cdot8^1 + 5\cdot8^0 + 4\cdot8^{-1} = 141,5_{(10)}$;

თექვსმეტობითი სისტემა

$C9,4_{(16)} - 3B,C_{(16)}$

გ

$$\begin{array}{r} 1\,1 \leftarrow \text{სესხები} \\ C\,9\,,4 \\ -\ 3\,B\,,C \\ \hline 8\,D\,,8 \end{array}$$

$8D,8_{(16)} = 8\cdot16^1 + D\cdot16^0 + 8\cdot16^{-1} = 141,5_{(10)}$

6 **გამრავლება.** თვლის სხვადასხვა პოზიციურ სისტემებში მრავალნიშნა რიცხვების გადასამრავლებლად შეიძლება გამოვიყენოთ ქვეშმიწერის ჩვეულებრივი ალგორითმი, ოღონდ ერთნიშნა რიცხვების გადამრავლებითა და შეკრებით მიღებული შედეგები თვლის მოცემული სისტემისათვის აგებული გამრავლებისა და შეკრების ცხრილებიდან უნდა ავიღოთ. შეკრების ცხრილები ზემოთ განვიხილეთ (იხ. ნახ. 3.2), ხოლო გამრავლების ცხრილები 3.4 და 3.5 ნახაზებზეა მოცემული.

ა) გამრავლება თვლის ორობით სისტემაში

×	0	1
0	0	0
1	0	1

ბ) გამრავლება თვლის რვაობობით სისტემაში

×	0	1	2	3	4	5	6	7
0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7
2	0	2	4	6	10	12	14	16
3	0	3	6	11	14	17	22	25
4	0	4	10	14	20	24	30	34
5	0	5	12	17	24	31	36	43
6	0	6	14	22	30	36	44	52
7	0	7	16	25	34	43	52	61

ნახ.3.4. გამრავლების ცხრილები თვლის 2-ობითი (ა) და 8-ობობითი (ბ) სისტემებისათვის

გამრავლება თექვსმეტობით სისტემაში

×	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

ნახ.3.5. გამრავლების ცხრილი თვლის 16-ობითი სისტემისათვის

გამრავლების ცხრილის სიმარტივის გამო გამოთვლის **ორობით სისტემაში გამრავლების პროცესი სამრავლის ძვრისა და თანრიგების შეკრების ოპერაციებამდე დაიყვანება. 3.5** და **3.6** მაგალითებში ილუსტრირებულია გამრავლების პროცესი.

მ ა გ ა ლ ი თ ი 3.5.

გადავამრავლოთ რიცხვები 5 და 6 თვლის 2-, 8- და 16-ობით სისტემებში

პასუხი: $5 \cdot 6 = 30_{(10)} = 11110_{(2)} = 36_{(8)} = 1E_{(16)}$

ორობითი სისტემა	რვაობითი სისტემა	თექვსმეტობითი სისტემა
$101_{(2)} \cdot 110_{(2)}$	$5_{(8)} \cdot 6_{(8)}$	$5_{(16)} \cdot 6_{(16)}$
ა $\begin{array}{r} 101 \\ \times\ 110 \\ \hline +\ 101\ \\ 101\ \ \\ \hline 11110 \end{array}$	ბ $\begin{array}{r} \times\ 5 \\ 6 \\ \hline 36 \end{array}$	გ $\begin{array}{r} \times\ 5 \\ 6 \\ \hline 1E \end{array}$
$11110_{(2)} = 2^4+2^3+2^2+2^1 = 16+8+4+2 = 30_{(10)}$	$36_{(8)} = 3 \cdot 8^1+6 \cdot 8^0 = 24+6 = 30_{(10)}$	$36_{(16)} = 1 \cdot 16^1+E \cdot 16^0 = 16+14 = 30_{(10)}$

მ ა გ ა ლ ი თ ი 3.6.

გადავამრავლოთ რიცხვები 115 და 51 თვლის 2-, 8- და 16-ობით სისტემებში

$115_{(10)}=1110011_{(2)}=163_{(8)}=73_{(16)}$;
$51_{(10)}=110011_{(2)}=63_{(8)}=33_{(16)}$;

პასუხი: $115\ \ 51 = 5865_{(10)} = 1011011101001_{(2)} = 13351_{(8)} = 16E9_{(16)}$

ო რ ო ბ ი თ ი   ს ი ს ტ ე მ ა

$1110011_{(2)} \cdot 110011_{(2)}$

ა)

$$\begin{array}{r}
1110011 \\
\times\quad 110011 \\
\hline
+\quad 1110011 \\
1110011 \\
+\quad 1110011 \\
+\quad 1110011 \\
\hline
1011011101001
\end{array}$$

$1011011101001_{(2)}=2^{12}+2^{10}+2^{9}+2^{7}+2^{6}+2^{5}+2^{3}+2^{0} = 5865_{(10)}$

რვაობითი სისტემა

$163_{(8)} \cdot 63_{(8)}$

ბ)

$$\begin{array}{r}
163 \\
\times\quad 63 \\
\hline
+\quad 531 \\
1262 \\
\hline
13351
\end{array}$$

$13351_{(8)}= 1\cdot8^{4} + 3\cdot8^{3} + 3\cdot8^{2} + 5\cdot8^{1} + 1\cdot8^{0} = 5865_{(10)}$

თექვსმეტობითი სისტემა

$73_{(16)} \cdot 33_{(16)}$

გ)

$$\begin{array}{r}
73 \\
\times\quad 33 \\
\hline
+\quad 159 \\
159 \\
\hline
16E9
\end{array}$$

$16E9_{(16)}= 1\cdot16^{3} + 6\cdot16^{2} + E\cdot16^{1} + 9\cdot16^{0} = 5865_{(10)}$

მ ა გ ა ლ ი თ ი 3.7.

35 გავყოთ 14-ზე თვლის 2-ობით, 8-ობით და16-ობით სისტემაში. $35_{(10)}=100011_{(2)}=43_{(8)}=23_{(16)}$; $14_{(10)}=1110_{(2)}=16_{(8)}=E_{(16)}$

პასუხი: $35:14=2{,}5_{(10)}=2{,}4_{(8)}=2{,}8_{(16)}$

ორობითი სისტემა

$100011_{(2)} : 1110_{(2)}$

ა)

$$\begin{array}{r|l}
-\ 100011 & 1110 \\
1110 & 10{,}1 \\
\cline{1-1}
-\ 1110 & \\
1110 & \\
\cline{1-1}
0 &
\end{array}$$

$10{,}1_{(2)}=2^{1}+2^{0}+2^{-1}=2{,}5_{(10)}$

რვაობითი სისტემა

$4\ 3_{(8)} : 1\ 6_{(8)}$

ბ)

$$\begin{array}{r|l}
-\ 43 & 16 \\
34 & 2{,}4 \\
\cline{1-1}
-\ 70 & \\
70 & \\
\cline{1-1}
0 &
\end{array}$$

$2{,}4_{(8)}=2\cdot8^{0}+1\cdot8^{-1}=2{,}5_{(10)}$

თექვსმეტობითი სისტემა

$2\ 3_{(16)} : E_{(16)}$

გ)

$$\begin{array}{r|l}
-\ 23 & E \\
1C & 2{,}8 \\
\cline{1-1}
-\ 70 & \\
70 & \\
\cline{1-1}
0 &
\end{array}$$

$2{,}4_{(8)}=2\cdot8^{0}+1\cdot8^{-1}=2{,}5_{(10)}$

7 **გაყოფა**. თვლის მოცემულ სისტემებში გამოიყენება კუთხური მიწერით გაყოფის წესი. ორობითი რიცხვების გაყოფა ძალიან მარტივია, რადგან განაყოფის თითოეული მომდევნო რიცხვი შეიძლება იყოს მხოლოდ ნული ან ერთი. 3.7 და 3.8 მაგალითებში ილუსტრირებულია გაყოფის პროცესი.

მ ა გ ა ლ ი თ ი 3.8.

5865 გავყოთ 115-ზე თვლის ორობით, რვაობით და თექვსმეტობით სისტემაში;	**ო რ ო ბ ი თ ი ს ი ს ტ ე მ ა**
	$1011011101001_{(2)}$ **:** $1110011_{(2)}$
$5865_{(10)}=13351_{(8)}=16E9_{(16)}= = 1011011101001_{(2)}$; $115_{(10)} = 163_{(8)} = 73_{(16)} = = 1110011_{(2)}$	**ა** $\begin{array}{r\|l} 1\,0\,1\,1\,0\,1\,1\,1\,0\,1\,0\,0\,1 & 1\,1\,1\,0\,0\,1\,1 \\ -\;1\,1\,1\,0\,0\,1\,1 & \overline{1\,1\,0\,0\,1\,1} \\ \hline 1\,0\,0\,0\,1\,0\,0\,0 & \\ -\;1\,1\,1\,0\,0\,1\,1 & \\ \hline 1\,0\,1\,0\,1\,1\,0\,0 & \\ -\;1\,1\,1\,0\,0\,1\,1 & \\ \hline 1\,1\,1\,0\,0\,1\,1 & \\ 1\,1\,1\,0\,0\,1\,1 & \\ \hline 0 & \end{array}$
პასუხი: $5865 : 115 = 51_{(10)} = 110011_{(2)} = 63_{(8)} = = 33(16)$	$110011_{(2)} = 2^5+2^4+2^3+2^1+2^0 = 51_{(10)}$

რვაობითი სისტემა	**თექვსმეტობითი სისტემა**
$1\,3\,3\,5\,1_{(8)}$ **:** $1\,6\,3_{(8)}$	$1\,6\,E\,9_{(16)}$ **:** $1\,7\,3_{(16)}$
ბ $\begin{array}{r\|l} 1\,3\,3\,5\,1 & 1\,6\,3 \\ -\;1\,2\,6\,2 & \overline{6\,3} \\ \hline 5\,3\,1 & \\ -\;5\,3\,1 & \\ \hline 0 & \end{array}$	**გ** $\begin{array}{r\|l} 1\,6\,E\,9 & 7\,3 \\ -\;1\,5\,9 & \overline{3\,3} \\ \hline 1\,5\,9 & \\ -\;1\,5\,9 & \\ \hline 0 & \end{array}$
$6\;3_{(8)} = 6\cdot8^1 + 3\cdot8^0 = 51_{(10)}$	$3\;3_{(16)} = 3\cdot16^1 + 3\cdot16^0 = 51_{(10)}$

3.2. თვლის ორობით-ათობითი სისტემა

“ვინ გიშლის გამოიგონო ულტობი დენთი?”

კოზმა პრუტკოვი

1 ყოველდღიურ ცხოვრებაში ადამიანი იყენებს **თვლის ათობით სისტემას**. ციფრულ მოწყობილობებში ათობითი რიცხვების შესანახად და დასამუშავებლად საჭიროა ისინი ორობითი კოდების სახით იქნეს წარმოდგენილი, ე.ი. გამოყენებული იქნეს **თვლის ორობითი სისტემა.** ზემოთ აღნიშნული (ორობითი და ათობითი) სისტემების ზოგადი თვისებების ურთიერთშერწყმით დამუშავებული იქნა კომპრომისული სისტემა, რომელმაც **ორობით-ათობითი სისტემის** სახელწოდება მიიღო.

ა)

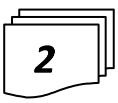
ციფ-რები	კ ო დ ი					
	8 4 2 1	7 4 2 1	2 4 2 1	3-ის სი-ჭარბით	7 5 (-3) 1	53 (-2) 1
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0
1	0 0 0 1	0 0 0 1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 0 1
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 1	0 1 1 0	0 1 1 1
3	0 0 1 1	0 0 1 1	0 0 1 1	0 1 1 0	0 1 1 1	1 0 1 0
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1	1 0 1 0	0 1 0 1
5	0 1 0 1	0 1 0 1	1 0 1 1	1 0 0 0	0 1 0 0	1 0 0 0
6	0 1 1 0	0 1 1 0	1 1 0 0	1 0 0 1	0 1 0 1	1 0 0 1
7	0 1 1 1	1 0 0 0	1 1 0 1	1 0 1 0	1 0 0 0	1 1 1 1
8	1 0 0 0	1 0 0 1	1 1 1 0	1 0 1 1	1 0 0 1	1 1 0 0
9	1 0 0 1	1 0 1 0	1 1 1 1	1 1 0 0	1 1 1 1	1 1 0 1

ბ)

ციფრ-ები	კ ო დ ი	
	$3a+2$	5-დან 2
0	0 0 0 1 0	1 1 0 0 0
1	0 0 1 0 1	0 1 1 0 0
2	0 1 0 0 0	0 0 1 1 0
3	0 1 0 1 1	0 0 0 1 1
4	0 1 1 1 0	1 0 0 0 1
5	1 0 0 0 1	1 0 1 0 0
6	1 0 1 0 0	0 1 0 1 0
7	1 0 1 1 1	0 0 1 0 1
8	1 1 0 1 0	1 0 0 1 0
9	1 1 1 0 1	0 1 0 0 1

ნახ.3.6. ათობითი ციფრების კოდირებისათვის გამოსაყენებელი ზოგიერთი ორობითი კოდი

ათობითი ციფრების გამომსახველ ორობით რიცხვებს **ორობითი კოდური სიტყვები** ეწოდება, ხოლო ამ უკანასკნელებისაგან წარმოქმნილ **M** სიმრავლეს – **ორობითი კოდი.** **M** სიმრავლის ელემენტების რაოდენობა ათობითი ციფრების (10-ის) რაოდენობაზე ნაკლები არ უნდა იყოს, ე.ი. $|M| \geq 10$**;** ვინაიდან n თანრიგიანი ორობითი რიცხვების საერთო რაოდენობაა 2^n, ამიტომ საჭიროა სრულდებოდეს პირობა $2^n \geq 10$, საიდანაც მიიღება, რომ $n \geq 4$; მაშასადამე ორობითი კოდური სიტყვების შემადგენელი ბიტების რაოდენობა ოთხზე ნაკლები არ უნდა იყოს. 4-თანრიგიან ორობით რიცხვებს **ტეტრადები** ან **ნახევარბაიტები** ეწოდება. ტეტრადების დახმარებით შეიძლება შევადგინოთ 0-ისა და 1-ისაგან შემდგარი არა 10, არამედ $2^4 = 16$ სხვადასხვა კომბინაცია. სხვადასხვა 4-თანრიგიანი კოდები (ტეტრადები) 10 ელემენტებად 16 ელემენტის ჯუფდებადობით მიიღება და მათი საერთო რაოდენობა $\mathbf{C_{16}^{10}} = 8008$-ის ტოლია; ე.ი. 4-თანრიგიანი კოდების საერთო რაოდენობაა 8008. ზოგიერთი მათგანი 3.4 ნახაზზე მოყვანილ ცხრილებშია ნაჩვენები.



3.6 ნახაზზე მოყვანილ ცხრილებში თითოეული კოდისათვის მითითებულია ათობითი ციფრების კოდირებისათვის გამოყენებული ათ-ათი კომბინაცია, რომელსაც **ნებადართული კომბინაცია** ეწოდება; დანარჩენი კომბინაციები კოდირებისათვის არ გამოიყენება და მათ **აკრძალული კომბინაციები** ეწოდება. ნებადართული და აკრძალული კომბინაციების არსებობა წარმოადგენს ორობით-ათობითი კო-

დების **ძირითად თავისებურებას,** რომლითაც იგი განსხვავდება თვლის ჩვეულებრივი პოზიციური სისტემებისაგან: ამ უკანასკნელებში ნებადართულია ყველა კომბინაცია.

3.5,ა ნახაზზე მოყვანილია **4-**თანრიგიანი, ხოლო **3.5,ბ** ნახაზზე – **5-**თანრიგიანი კოდები. **4-**თანრიგიანი კოდების ღირსებაა ის, რომ ისინი საშუალებას გვაძლევს ათობითი ციფრების კოდირებისათვის მინიმალური რაოდენობის თანრიგები გამოვიყენოთ. დამატებითი მე-5 თანრიგის შემოტანა საშუალებას გვაძლევს აღმოვაჩინოთ კავშირგაბმულობის ხაზებში რიცხვითი ინფორმაციის გადაცემის პროცესში წარმოშობილი შეცდომები.

5-თანრიგიანი "**5-დან 2**" კოდი მიეკუთვნება ე.წ. **მუდმივწონიანი კოდების** სიმრავლეს. **კოდის წონა** ეწოდება მის კოდურ სიტყვებში არსებული ერთიანების რაოდენობას. მუდმივი წონის მქონე კოდის ყველა კოდური სიტყვა ერთნაირი რაოდენობის ერთიანებს შეიცავს, ე.ი. მუდმივია კოდური სიტყვების წონა. "**5-დან 2**" კოდი ხშირად აღინიშნება როგორც **5C2**-კოდი; მისი თითოეული კოდური სიტყვა შეიცავს ორ ერთიანს, ანუ თითოეული კოდური სიტყვის წონა მუდმივია და იგი **2-ის** ტოლია, აღნიშნული კოდის ნებადართული კოდური სიტყვის ნებისმიერი თანრიგის მნიშვნელობის შეცვლისას წარმოიშობა აკრძალული კომბინაცია. კერძოდ, თუ კოდური სიტყვის რომელიმე ნულოვანი თანრიგი მიიღებს 1-ის ტოლ მნიშვნელობას, მაშინ გაიზრდება ამ სიტყვის წონა, ხოლო თუ რომელიმე ერთის ტოლი თანრიგი მიიღებს 0-ის ტოლ მნიშვნელობას, მაშინ პირიქით – წონა შემცირდება. ორივე შემთხვევაში ნებადართული კოდური სიტყვა გარდაიქმნება აკრძალულ კოდურ სიტყვად და შეცდომა გამომჟღავნდება.

5-თანრიგიანი **3*a*+2** კოდში კოდური ნებისმიერი ორი კოდური კომბინაცია სულ მცირე ორ განსხვავებულ თანრიგს შეიცავს. ამის გამო შეცდომა, რომელიც ცვლის რომელიმე კოდური კომბინაციის ნებისმიერ თანრიგს, წარმოშობს აკრძალულ კომბინაციას.

ერთმანეთისაგან განასხვავებენ **წონით** და **არაწონით კოდებს.**

წონით კოდებს მიეკუთვნება კოდები **8421, 7421, 2421, 75(-3)1** და **53(-2)1** კოდები (**ნახ.3.5**), რომლებისთვისაც ათობითი **D** ციფრის მნიშვნელობა განისაზღვრება შემდეგი გამოსახულებით:

$$\mathbf{D = d_3 \cdot \sigma_3 + d_2 \cdot \sigma_2 + d_1\, \sigma_1 + d_0\, \sigma_0 = (d_3\, d_2\, d_1\, d_0)_{(2)}}, \qquad \mathbf{(3.3)}$$

სადაც $\mathbf{d}_k$ არის ტეტრადის $\boldsymbol{k}$–ური თანრიგის მნიშვნელობა ($\mathbf{d}_k \in \{0;1\}$); $\boldsymbol{\sigma}_k$ - ტეტრადის $\boldsymbol{k}$–ური თანრიგის წონა, გამოხატული თვლის ათობითი სისტემით; $\boldsymbol{k}$=**0;1;2;3;** შევნიშნავთ, რომ **75(-3)1** და **53(-2)1** კოდებისათვის 1-ლი თანრიგის $\boldsymbol{\sigma_1}$ წონა შესაბამისად **-3**-ისა და **-2**-ის ტოლია.

(**3.3**) თანაფარდობა საშუალებას გვაძლევს:

- ათობითი **D** ციფრის ცნობილი მნიშვნელობით განვსაზღვროთ მისი $\mathbf{d_3\, d_2\, d_1\, d_0}$ კოდი, ე.ი. $\mathbf{d_3}$; $\mathbf{d_2}$; $\mathbf{d_1}$; $\mathbf{d_0}$ თანრიგების მნიშვნელობები;
- ცნობილი $(\mathbf{d_3\, d_2\, d_1\, d_0})$ კოდით განვსაზღვროთ **D** ციფრის მნიშვნელობა.

მაგალითი 3.9. **7421** კოდისათვის (**3.3**) გამოსახულება იღებს სახეს:

$$\mathbf{D = d_3 \cdot 7 + d_2 \cdot 4 + d_1 \cdot 2 + d_0 \cdot 1 = (d_3\, d_2\, d_1\, d_0)_{(2)}},$$

მიღებული გამოსახულებიდან ჩანს, რომ **7421** კოდში ათობითი **0;1;2;3; 4;5;6** ციფრები კოდირდება მათი ეკვივალენტური **4**-ნიშნა ორობითი რიცხვებით; დანარჩენი **7, 8** და **9** ციფრების კოდირებისათვის (**3.4**) გამოსახულების შესაბამისად შემდეგი კოდები მიიღება:

$$\mathbf{7 \;\Rightarrow\; D = 1 \cdot 7 + 0 \cdot 4 + 0 \cdot 2 + 0 \cdot 1 = (1\,0\,0\,0)_{(2)}},$$

$$\mathbf{8 \;\Rightarrow\; D = 1 \cdot 7 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = (1\,0\,0\,1)_{(2)}},$$

$$\mathbf{9 \;\Rightarrow\; D = 1 \cdot 7 + 0 \cdot 4 + 1 \cdot 2 + 0 \cdot 1 = (1\,0\,1\,1)_{(2)}}.$$

მაგალითი 3.10. **8421** კოდისათვის (**3.3**) გამოსახულება იღებს სახეს:

$$\mathbf{D = d_3 \cdot 8 + d_2 \cdot 4 + d_1 \cdot 2 + d_0 \cdot 1 = d_3\, 2^3 + d_2\, 2^2 + d_1\, 2^1\ d_0}\ ,$$

სადაც $\mathbf{d}_k \in \{0;1\}$;

იგი შეესაბამება ათობითი რიცხვების ჩანაწერს **2**-ის ფუძის მქონე თვლის სისტემაში.

მაგალითი 3.11. (**3.3**) გამოსახულების გამოყენებით განვსაზღვროთ ორობით-ათობითი **75(-3)1** კოდის თანრიგების მნიშვნელობები (**ცხრ.3.3**).

ცხრ. 3.3. 75(-3)1 კოდის კოდური სიტყვების გამოთვლა

ციფრები	$d_3 \cdot 7 + d_2 \cdot 5 - d_1 \cdot 3 + d_0 \cdot 1$	$(d_3\, d_2\, d_1\, d_0)_{(2)}$
0	$0 \cdot 7 + 0 \cdot 5 - 0 \cdot 3 + 0 \cdot 1$	0000
1	$0 \cdot 7 + 0 \cdot 5 - 0 \cdot 3 + 1 \cdot 1$	0001
2	$0 \cdot 7 + 1 \cdot 5 - 1 \cdot 3 + 0 \cdot 1$	0110
3	$0 \cdot 7 + 1 \cdot 5 - 1 \cdot 3 + 1 \cdot 1$	0111
4	$1 \cdot 7 + 0 \cdot 5 - 1 \cdot 3 + 0 \cdot 1$	1010
5	$0 \cdot 7 + 1 \cdot 5 - 0 \cdot 3 + 0 \cdot 1$	0100
6	$0 \cdot 7 + 1 \cdot 5 - 0 \cdot 3 + 1 \cdot 1$	0101
7	$1 \cdot 7 + 0 \cdot 5 - 0 \cdot 3 + 0 \cdot 1$	1000
8	$1 \cdot 7 + 0 \cdot 5 - 0 \cdot 3 + 1 \cdot 1$	1001
9	$1 \cdot 7 + 1 \cdot 5 - 1 \cdot 3 + 0 \cdot 1$	1110

არაწონითი კოდებისათვის (**3.3**) თანაფარდობა არ სრულდება. ასეთ კოდებს მიეკუთვნება კოდი **3**-ის სიჭარბით (რომელსაც **8421+3-კოდი** ეწოდება) და **5**–თანრიგიანი კოდები: **3*a*+2** და **5-დან 2** კოდი; ამ უკანასკნელ კოდს ხშირად **5C2**–კოდსაც უწოდებენ. ზოგიერთი ორობით-ათობითი კოდისათვის შემდეგი თავისებურებაა დამახასიათებელი. $\mathbf{D_k}$ ციფრის მიმართ $\mathbf{D_k}$ ციფრი წარმოადგენს **9**-მდე დამატებას, თუ სრულდება პირობა $\mathbf{D_i + D_k = 9}$.

9-მდე ერთმანეთის დამატებას წარმოადგენს ათობითი ციფრების შემდეგი წყვილები: (**0** და **9**); (**1** და **8**); (**2** და **7**); (**3** და **6**;) (**4** და **5**).

ორი ათობითი:

$$\mathbf{D_i = (d_{3,i}\, d_{2,i}\, d_{1,i}\, d_{0,i})} \text{ და } \mathbf{D_k = (d_{3,k}\, d_{2,k}\, d_{1,k}\, d_{0,k})}$$

რიცხვისათვის თუ სრულდება პირობა:

$$\mathbf{d_{j,i}} = \begin{cases} \mathbf{0,\ \text{თუ}\ d_{j,i} = 1,} \\ \mathbf{1,\ \text{თუ}\ d_{j,i} = 0,} \end{cases} \quad \mathbf{j = 0;\ 1;\ 2;\ 3,} \qquad (3.4)$$

მაშინ $\mathbf{D_i = (d_{3,i}\, d_{2,i}\, d_{1,i}\, d_{0,i})}$ რიცხვის დამატება შეიძლება მივიღოთ $\mathbf{D_k = (d_{3,k}\, d_{2,k}\, d_{1,k}\, d_{0,k})}$ რიცხვიდან ამ უკანასკნელში ნულების ერთიანებით და ერთიანების – ნულებით შეცვლის გზით.

(**3.4**) თვისება გააჩნია **2421;** **3-ის სიჭარბით** და **3*a* +2** კოდებს (**ნახ.3.4**); მაგალითად, **2421** კოდში ცხრამდე დამატების წარმომქმნელ **3** და **6** რიცხვებს შეესაბამება კომბინაციები **0011** და **1100,** რომელთაგანაც თითოეული მათგანი მიიღება ამ კომბინაციებიდან თანრიგების ინვერსირების გზით.

უარყოფით–თანრიგიანი ორობით-ათობითი $D_i = -D_{n-1}D_{n-2}\dots D_k\dots D_0$ **რიცხვი** შეიძლება წარმოვადგინოთ **პირდაპირი, დამატებითი და შებრუნებული კოდების** სახით:

$$\left.\begin{array}{l} D_{პირ.} = 1\ D_{n-1}D_{n-2}\dots D_k\dots D_0 \ \text{(პირდაპირი კოდი)} \\ D_{შებრ.} = 1\ \tilde{D}_{n-1}\tilde{D}_{n-2}\dots\tilde{D}_k\dots\tilde{D}_0 \ \text{(შებრუნებული კოდი)} \\ D_{დამატ.} = D_{შებრ.} + 1 \ \text{(დამატებითი კოდი)} \end{array}\right\} \quad (3.5)$$

სადაც $D_k = (d_{3,k}\,d_{2,k}\,d_{1,k}\,d_{0,k})_{(2)}$ არის ტეტრადები, (k=0; 1; 2; ... ; n-1); 1 წარმოადგენს უარყოფითი რიცხვების ნიშნის მნიშვნელობას; $\tilde{D}_k$ არის 9-მდე დამატება, რომელსაც აქვს შემდეგი თვისება:

$$D_k + \tilde{D}_k = 1001_{(2)} = 9_{(10)}. \qquad (3.6)$$

(3.5) თანაფარდობიდან გამოდის, რომ უარყოფითი ორობით-ათობითი რიცხვების:

- **პირდაპირი კოდის** სახით წარმოსადგენად, საკმარისია რიცხვს წინ დავუმატოთ უარყოფითი რიცხვის ნიშანი 1; $\overline{D_k}$
- **შებრუნებული კოდის** სახით წარმოსადგენად, საჭიროა რიცხვის ტეტრადები გარდავქმნათ (3.6) პირობის შესაბამისად; ამისათვის (როგორც ზემოთ ავღნიშნეთ), თუ ათობითი ციფრების კოდირებისათვის გამოყენებულია 2421, 3a+2 ან 3-ის სიჭარბის კოდები, რომლებისთვისაც სრულდება (3.4) პირობა, მაშინ შებრუნებული კოდი მიიღება D_k ტეტრადის თანრიგობრივი ინვერსიით, ე.ი. $\tilde{D}_k = \overline{D_k}$;
- **დამატებითი კოდის** სახით წარმოსადგენად (იგულისხმება 10-მდე დამატება), საჭიროა ჯერ მივიღოთ დამატებითი კოდი და მიღებულ შედეგს დავუმატოთ ორობითი ციფრი 1.

მაგალითი 3.12. ათობითი -841 რიცხვი 2421 კოდის გამოყენებით გამოვსახოთ ორობით-ათობით ფორმით და მიღებული რიცხვი წარმოვადგინოთ პირდაპირი, შებრუნებული და დამატებითი კოდის სახით;

ამოხსნა. ა) -841 რიცხვის ორობით-ათობით ფორმას 2421 კოდის გამოყენების დროს ექნება სახე (იხ. ნახ. 3.6): - 1110 0100 0001;

ბ) ზემოთ ფორმულირებული წესების შესაბამისად: $D_{პირ} = 1\ 1110\ 0100\ 0001$;

$D_{შებრ.} = 1\ \underbrace{0001}_{1}\ \underbrace{1011}_{5}\ \underbrace{1110}_{8}$; $D_{დამატ.} = D_{შებრ.} + 1 = 1\ \underbrace{0001}_{1}\ \underbrace{1011}_{5}\ \underbrace{1111}_{9}$.

რიცხვის ორობით-ათობითი სახით წარმოდგენისათვის 8421 კოდის გამოყენების დროს (3.6) პირობა არ სრულდება. ამის გამო ტეტრადების თანრიგობრივი ინვერსირების შედეგად მიიღება არა 9-მდე, არამედ 15-მდე დამატება. მაგალითად, ავიღოთ რიცხვი $5_{(10)} = 0101_{(2)}$;

0101 რიცხვის თანრიგობრივი ინვერსირების შედეგად მიიღება რიცხვი $1010_{(2)} = 10_{(10)}$; ეს უკანასკნელი რიცხვი 10 წარმოადგენს საწყისი რიცხვის 5-ის დამატებას 15-მდე. ასეთ შემთხვევაში ორობით-ათობითი რიცხვის შებრუნებით კოდში წარმოდგენისათვის საჭიროა ჩავატაროთ შემდეგი ოპერაციები:

- ორობით-ათობითი რიცხვის თითოეულ ტეტრადს დავუმტოთ ორობითი რიცხვი 0110 (6);
- მიღებულ ჯამზე შევასრულოთ ინვერსიის ოპერაცია;
- ნიშნის თანრიგში ჩავწეროთ ბიტი 1.

დამატებითი კოდის მისაღებად გამოიყენება გამოსახულება $D_{დამატ.} = D_{შებრ.} + 1$.

მაგალითი 3.13. – ათობითი **-841** რიცხვი **8421** კოდის გამოყენებით გამოვსახოთ ორობით-ათობით ფორმით და მიღებული რიცხვი წარმოვადგინოთ პირდაპირი, შებრუნებული და დამატებითი კოდის სახით;

ამოხსნა. ა) **-841** რიცხვის ორობით-ათობით ფორმას **2421** კოდის გამოყენების დროს ექნება სახე (იხ. ნახ. **3.6**): - **1110 0100 0001;**

ბ) ზემოთ ფორმულირებული წესების შესაბამისად: $\mathbf{D_{პირ}}$ **=1 1110 0100 0001;**

$$\mathbf{D_{შებრ.}} = 1\ \underbrace{(\mathbf{D_2} + 0110)(\mathbf{D_1} + 0110)\ (\mathbf{D_0} + 0110)} = 1\ 0001\ 0101\ 1000$$

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

ინვერსია → 0 0 0 1 0 1 0 1 1 0 0 0

1 5 8

$$\mathbf{D_{დამატ.}} = \mathbf{D_{შებრ.}} +1 = 1\ 0001\ 0101\ 1000 + 1 = 1\ 0001\ 0101\ 1001$$

1 5 9

3.3. ორობით-ათობითი რიცხვების შეკრების თავისებურებები

1

ზოგიერთი კომპიუტერი, მაგალითად ***IBM მეინფრეიმი,*** იყენებს ორობით-ათობით რიცხვებს. არსებობს ორობით-ათობითი რიცხვების შეკრების რამდენიმე ხერხი. გავეცნოთ **8421** კოდით წარმოდგენილი ერთთანრიგიანი დადებითი რიცხვების შეკრების ერთ-ერთ ასეთ ხერხს.

ტეტრადების შეკრებისას გამოიყენება ორობითი არითმეტიკის წესები. მიღებული ჯამი თუ **9**-ზე მეტი აღმოჩნდება, წამოიჭრება **კორექციის ჩატარების** აუცილებლობა. **10**-დან **19**-ერთეულამდე (**1010; 1011; ...; 10011**) ჯამის ტოლობა შემდეგი ნიშნებით შეიძლება გამოვლინდეს:

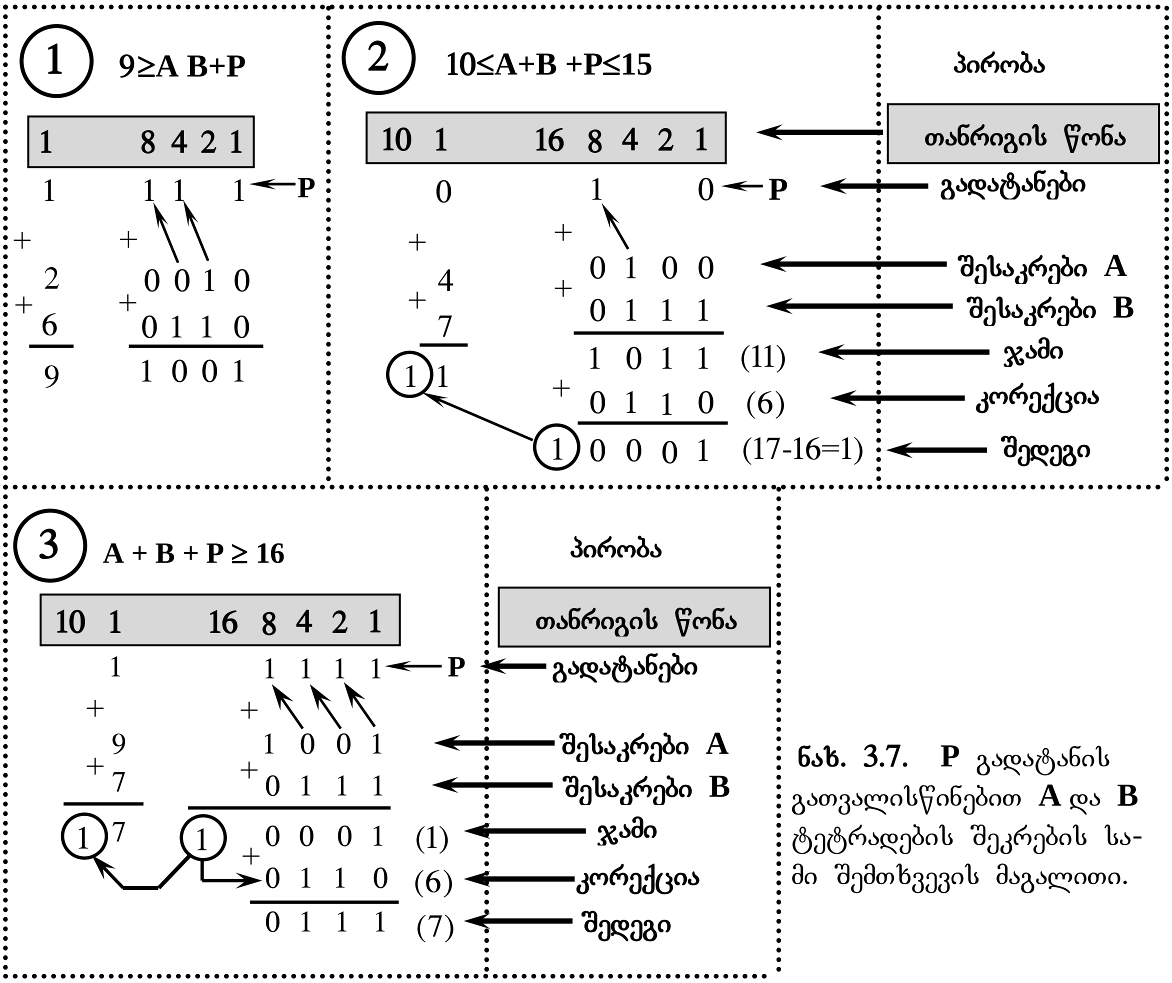
- შეკრების დროს შემდეგ ტეტრადაში გადატანის სახით ჩნდება ორობითი რიცხვის მეხუთე თანრიგი (ასეთებია **10000**-დან **10011-**მდე ორობითი რიცხვები, ანუ **16**-დან **19**-მდე ათობითი რიცხვები);
- ერთიანების არსებობა ა) **8**-ისა და **2**-ის წონის მქონე თანრიგებში (ორობითი **1010** და **1011** ანუ ათობითი **10** და **11** რიცხვები); ბ) **8-**სა და **4-**ის მქონე თანრიგებში (**1100**-დან **1111-**მდე ორობითი ანუ **12**-დან და **15-**მდე ათობითი რიცხვები).

კორექცია ხორციელდება მიღებული შედეგისათვის **0110** ($\mathbf{6_{(10)}}$) რიცხვის მიმატებით. ამ რიცხვს პირობითად **მაკორექტირებელი რიცხვი** ეწოდება. ასეთი ოპერაციის ჩატარების აუცილებლობა იმით საბუთდება, რომ ორობითი რიცხვის მეხუთე თანრიგის წონა **16** ათობით ერთეულს შეადგენს, ხოლო უფროს ტეტრადაში მისი გადატანით აღნიშნული ტეტრადის მნიშვნელობა მხოლოდ **10** ათობითი ერთეულით იზრდება; მათ შორის სხვაობაა **6** ერთეული, რომელიც უკვალოდ იკარგება, რაც ამახინჯებს საერთო შედეგს. აღნიშნული დანაკარგის "ასანაზღაურებლად" ხელოვნურად **6**-ით უნდა გავზარდოთ იმ

ტეტრადის წონა, საიდანაც განხორციელდა გადატანა. კორექციის პრინციპი მაგალითე-ბის საშუალებით ავხსნათ.

მაგალითი 3.14. P გადატანის გათვალისწინებით შევკრიბოთ ორი **A** და **B** ტე-ტრადა სამი შესაძლო შემთხვევისათვის (ნახ. **3.7**).

- **1**-ლი შემთხვევის (ერთთანრიგიანი ათობითი **2; 6** და **9** რიცხვების **2+6+1=9** შეკრების ოპერაციის შესრულების) დროს სრულდება პირობა **A + B + P ≤ 9;** ასეთ შე-მთხვევაში უფროს ტეტრადაში გადატანა არ ხდება, მიღებული ჯამი შეესაბამება ორო-ბით-ათობითი კოდირების ნებადართულ კომბინაციას, ამიტომ კორექციის ჩატარება საჭი-რო არ არის. მიიღება სწორი შედეგი.



ნახ. 3.7. P გადატანის გათვალისწინებით **A** და **B** ტეტრადების შეკრების სა-მი შემთხვევის მაგალითი.

- მე-**2** შემთხვევის (ერთთანრიგიანი ათობითი **4; 7** და **0** რიცხვების **4+7+0=11** შეკრების ოპერაციის შესრულების) დროს სრულდება **10≤A+B+P≤15** პირობა. შეკრების შედეგად მიიღება ჯამი **1011;** მიღებულ რიცხვში ერთიანები არსებობს იმ თანრიგებში, რომელთა წონაა **8** და **2;** ზემოთ ფორმულირებული წესის მიხედვით ეს ნიშნავს, რომ მიღებული შედეგი არასწორია. აღნიშნული შედეგისათვის მაკორექტირებელი **0110** (**6**$_{(10)}$) რიცხვის მიმატებით მივიღებთ **5**-თანრიგიან **<u>1</u>0001** (**17**$_{(10)}$) რიცხვს. ხაზგასმული მეხუთე თანრიგი ‘‘გადადის’’ უფროს ტეტრადაში და მოცემული ტეტრადიდან ‘‘მიაქვს’’ **16** ერთეული, რის შედეგადაც ტოვებს სწორ **0001** (**1**$_{(10)}$) შედეგს. ამგვარად, მოცემულ შემთხვევაში განხორციელებულმა კორექციამ ერთიანი გააჩინა მეხუთე თანრიგში, შემ-დეგ ტეტრადაში, რომლის გადატანამაც საშუალება მოგვცა მიგვეღო სწორი პასუხი.

● მე-**3** შემთხვევის (ერთთანრიგიანი ათობითი **9; 7** და **1** რიცხვების **9+7+1=17** შეკრების ოპერაციის შესრულების) დროს სრულდება **A + B + P ≥ 16** პირობა; შედეგად ვიღებთ ხუთთანრიგიან **<u>1</u>0001** ($17_{(10)}$) რიცხვს, რომლის ხაზგასმული მეხუთე თანრიგი გადადის უფროს ტეტრადაში; მოცემულ ტეტრადას ამით აკლდება **16** ერთეული, ხოლო უფროს ტეტრადას ემატება **10** ერთეული. სწორი პასუხის მისაღებად მოცემულ ტეტრადას უნდა დავუმატოთ "უკვალოდ გამქრალი" **6** ერთეული, რომელიც ჩვენ\ მიერ შერჩეული მაკორექტირებელი რიცხვის ტოლია. კორექციის შედეგად მიიღება სწორი შედეგი: $0111_{(2)}=7_{(10)}$.

ცხრ. 3.4. შებრუნებული კოდების დროს წარმოშობილი **4** ძირითადი შემთხვევა

<u>1-ლი ძირითადი შემთხვევა</u>. **X** და **Y** დადებითია. შეკრებისას ნიშნის თანრიგების ჩათვლით იკრიბება ყველა თანრიგი. რადგან დადებითი შესაკრებების ნიშნის თანრიგები **0**-ის ტოლებია, ამიტომ ჯამის ნიშნის თანრიგიც **0**-ის ტოლია. მაგალითად:

10-ობითი ჩანაწერი	2-ობითი კ ო დ ი
$\begin{array}{r} 3 \\ +\ 7 \\ \hline 10 \end{array}$	$\begin{array}{r} 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1 \\ +\ 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1 \\ \hline 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0 \end{array}$

მიიღება კორექტული შედეგი.

<u>მე-2 ძირითადი შემთხვევა.</u> **X** დადებითია, **Y** – უარყოფითი და აბსოლუტური მნიშვნელობით **X**-ზე მეტია. მაგალითად:

10-ობითი ჩანაწერი	2-ობითი კ ო დ ი
$\begin{array}{r} 3 \\ +\ -10 \\ \hline -7 \end{array}$	$\begin{array}{r} 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1 \\ +\ 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1 \\ \hline 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0 \end{array}$

1 1110101 არის **-10**-ის შებრუნებული კოდი, **1 1111000** კი **-7**–ის შებრუნებული კოდი. მივიღეთ კორექტული შედეგი შებრუნებით კოდში. პირდაპირი კოდი მიიღება ჯამის ინვერტირებით:**1 0000111 = -7**

<u>მე-3 ძირითადი შემთხვევა.</u> **X** დადებითია, **Y** – უარყოფითი და აბსოლუტური მნიშვნელობით **X**-ზე ნაკლებია; მაგალითად:

10-ობითი ჩანაწერი	2-ობითი კ ო დ ი
$\begin{array}{r} 10 \\ +\ -3 \\ \hline 7 \end{array}$	$\begin{array}{r} 0\ 0\ 0\ 0\ 1\ 0\ 1\ 0 \\ +\ 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0 \\ \hline 0\ 0\ 0\ 0\ 0\ 1\ 1\ 0 \\ +1 \\ \hline 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1 \end{array}$

1 1111100 არის **-3**-ის შებრუნებული კოდი. კომპიუტერი ასწორებს დასაწყისში მიღებულ არაკორექტულ (**7**-ის ნაცვლად **6**) შედეგს ნიშნის თანრიგიდან **1**-ის გადატანით ჯამის უმცროს თანრიგში.

მე-4 ძირითადი შემთხვევა. **X** და **Y** უარყოფითია. მაგალითად:

10-ობითი ჩანაწერი	2-ობითი კ ო დ ი
$\begin{array}{r} -3 \\ +\ -7 \\ \hline -10 \end{array}$	$\begin{array}{r} 1\ 1\ 1\ 1\ 1\ 1\ 0\ 0 \\ +\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 0 \\ \hline 1\ 1\ 1\ 1\ 0\ 1\ 0\ 0 \\ +1 \\ \hline 1\ 1\ 1\ 1\ 0\ 1\ 0\ 1 \end{array}$

2-ობით კოდში მოყვანილია შებრუნებული კოდები. მიღებული არაკორექტული შედეგი სწორდება წინა შემთხვევის ანალოგიურად. პირდაპირი კოდი მიიღება ჯამის ინვერსიით: **1 0001010 = -10**$_{(10)}$

3.4. კომპიუტერში არითმეტიკული ოპერაციების შესრულების თავისებურებები

კომპიუტერების უმრავლესობაში მათი კონსტრუქციის გასამარტივებლად გამოკლების ოპერაცია არ გამოიყენება და იგი იცვლება შებრუნებული ან დამატებითი კოდების შეკრებით.

შებრუნებული კოდების შეკრების დროს წარმოიშობა ოთხი ძირითადი და ორი განსაკუთრებული შემთხვევა.

შებრუნებული კოდების შეკრების დროს წარმოშობილი ოთხი ძირითადი შემთხვევა მოცემულია **3.4** ცხრილში, ხოლო ორი განსაკუთრებული შემთხვევა – **3.5** ცხრილში.

ცხრ. 3.5. შებრუნებული კოდების დროს წარმოშობილი **2** განსაკუთრებული შემთხვევა

<u>1-ლი განსაკუთრებული შემთხვევა.</u> **X** და **Y** დადებითია, $X+Y \geq 2^{n-1}$, სადაც n არის რიცხვების ფორმატის თანრიგების რაოდენობა (ერთბაიტური ფორმატის დროს $n = 8$; $2^{n-1} = 2^7 = 128$). მაგალითად:

ათობითი ჩანაწერი	ორობითი კ ო დ ი	
$+\,65$	$+\,0\,1\,0\,0\,0\,0\,0\,1$	
97	$0\,1\,1\,0\,0\,0\,0\,1$	
162	$1\,0\,1\,0\,0\,0\,1\,0$	← **გადავსება**

რიცხვითი ფორმატის ციფრული ნაწილის შვიდი თანრიგი საკმარისი არ არის **8**-თანრიგიანი ჯამის განთავსებისათვის ($162_{(10)} = 10100010_{(2)}$), ამიტომ ჯამის უფროსი თანრიგი აღმოჩნდება ნიშნის თანრიგში, რის გამოც ჯამის ნიშანი არ დაემთხვევა შესაკრებების ნიშნებს; ნიშნების აღნიშნული დაუმთხვევლობა გვატყობინებს, რომ გადაივსო თანრიგობრივი ბადე.

<u>მე-2 განსაკუთრებული შემთხვევა</u>. **X**და**Y** უარყოფითია, $|X| + |Y| \geq 2^{n-1}$ მაგალითად:

ათობითი ჩანაწერი	ორობითი კ ო დ ი	
$+\,-63$	$+\,1\,1\,0\,0\,0\,0\,0\,0$	← -63-ის შებრუნებული კოდი
-95	$1\,0\,1\,0\,0\,0\,0\,0$	← -95-ის შებრუნებული კოდი
-158	$0\,1\,1\,0\,0\,0\,0\,0$	← გ ა დ ა ვ ს ე ბ ა

მოცემულ შემთხვევაშიც ჯამის ნიშანი არ ემთხვევა შესაკრებების ნიშნებს, რაც ადასტურებს თანრიგობრივი ბადის გადავსებას.

დამატებითი კოდების შეკრება. დამატებითი კოდების შეკრების დროსაც ექვსი შემთხვევა წარმოიშობა. **3.6** ცხრილში მოყვანილია **16**-თანრიგიანი კომპიუტერის მიერ მთელი **X** და **Y** რიცხვების შეკრების დროს წარმოშობილი თითოეული აღნიშნული შემთხვევა.

ორ (მეხუთე და მეექვსე) შემთხვევაში შეზღუდული თანრიგიანობის გამო მახინჯდება შედეგის არა მარტო სიდიდე, არამედ ნიშანიც. მდგომარეობის გამოსასწორებლად

შედეგს უნდა დავუმატოთ (მე-**5** შემთხვევის დროს) ან გამოვაკლოთ (მე-**6** შემთხვევის დროს) რიცხვი 2^{16}=**65536**.

გამოთვლების კორექტურობის უზრუნველსაყოფად, შეკრების დაწყებამდე საჭიროა შესაკრებთა ნიშნების გაანალიზებით დავადგინოთ მისაღები ჯამის ნიშანი. არასწორი ნი-

შნის მქონე ჯამის მიღებისას უნდა დავასკვნათ, რომ ოპერაცია არაკორექტურად შე-სრულდა.

არითმეტიკული ოპერაციების შესრულების გასაკონტროლებლად კომპიუტერის პროცესორში (მოწყობილობაში, რომელშიც სრულდება არითმეტიკული და ლოგიკური ოპერაციები) არსებობს:

- გადატანის ინდიკატორი,
- გადავსების ინდიკატორი;

თითოეული ზემოთ აღნიშნული ინდიკატორი შეიცავს 1 ბიტის ტოლ ინფორმაციას და შეუძლია ჰქონდეს მნიშვნელობა:

1 – რაც ნიშნავს, რომ ინდიკატორი დაყენებულია;

0 – რაც ნიშნავს, რომ ინდიკატორი ჩამოყრილია.

ცხრილი 3.6. დამატებითი კოდების კომპიუტერული შეკრების ვარიანტები

შემთხვ-ევები	შესაკრებები და შედეგი	კ ო მ ე ნ ტ ა რ ე ბ ი
1	**X > 0; Y > 0; X + Y < 2^{15}** +0 000 0110 0011 1010 **X = + 1594** 0 100 0100 1001 1011 **X = + 17563** 0 100 1010 1101 0101 **Z = + 19157**	შედეგი კორექტურია
2	**X > 0; Y < 0; |X| < |Y|** +0 000 0110 0011 1010 **X = + 1594** 1 011 1011 0110 1011 **Y = - 17563** 1 100 0001 1001 1111 **Z = - 15969**	შედეგი კორექტურია
3	**X > 0; Y < 0; |X| > |Y|** +0 100 0100 1001 1011 **X=+17563** 1 111 1001 1100 0111 **Y=-1594** 1 0 011 1110 0110 0001 **Z=+1596**	შედეგი კორექტურია; უფროსი თანრიგიდან გა-დატანა არ გაითვალისწ-ინება.
4	**X < 0; Y < 0; |X| + |Y| < 2^{15}** +1 111 1001 1100 0110 **X= -1594** 1 011 1011 0110 0101 **Y= -17563** 11011 0101 0010 1011 **Z= -19157**	შედეგი კორექტულია; უფროსი თანრიგიდან გა-დატანა არ გაითვალისწ-ინება.
5	**X > 0; Y > 0; X + Y ≥ 2^{15}** +0 100 0100 1001 1011 **X= +17563** 0 100 1010 1101 0101 **Y= +19157** 1 000 1111 0111 0000 **Z = - 28816**	დადებითი რიცხვების შე-კრებით მივიღეთ უარყო-ფითი რიცხვი; გ ა დ ა ვ ს ე ბ ა
6	**X < 0; Y < 0; |X| + |Y| > 2^{15}** +1 011 1011 0110 0101 **X= - 17563** 1 011 0101 0010 1011 **Y= - 19157** 1 011 0000 1001 0000 **Z= + 28816**	უარყოფითი რიცხვების შეკრებით მივიღეთ დადე-ბითი რიცხვი; გ ა დ ა ვ ს ე ბ ა

გადატანის ინდიკატორი გვიჩვენებს რომ მოხდა გადატანა ნიშნის ბიტიდან, ხოლო **გადავსების ინდიკატორი** გვიჩვენებს, რომ პირიქით – განხორციელდა ნიშნის ბიტში გადატანა.

ამგვარად, შეკრების ისეთი ოპერაციის შესრულებისას, როდესაც უფროს (ნიშნის) ბიტში ხდება გადატანა, პროცესორი აამოქმედებს გადავსების ინდიკატორს; ასეთი გადატანის არარსებობის შემთხვევაში აღნიშნული ინდიკატორი ჩამოიყრება. ანალოგიურად ფუნქციონირებს გადატანის ინდიკატორიც.

ოპერაციის შესრულების შემდეგ კომპიუტერი იძლევა სიგნალს ინდიკატორების მდგომარეობის შესახებ. ინდიკატორების მდგომარეობები თუ გვიჩვენებს, რომ არითმეტიკული შედეგი არასწორია, მაშინ უნდა მივღოთ მოცემული სიტუაციის გამოსასწორებლად საჭირო ზომები, ე.ი. შეგვიძლია ვაკონტროლოთ არითმეტიკული ოპერაციების შესრულების სისწორე.

შეკრებისა და გამოკლების ოპერაციები წარმოადგენენ კომპიუტერში შესასრულებელ **ძირითად არითმეტიკულ ოპერაციებს.** უფრო რთული ნებისმიერი სხვა ისეთი ოპერაცია, როგორიცაა მაგალითად გამრავლება, გაყოფა, ტრიგონომეტრიული ფუნქციების გამოთვლა და ა.შ., მრავალჯერადად შესასრულებელ შეკრებისა და გამოკლების ოპერაციებამდე დაიყვანება. ასე, მაგალითად, **გამრავლებას** კომპიუტერი ასრულებს სამრავლის მრავალჯერადად ძვრისა და შეკრების ოპერაციების მეშვეობით, ხოლო **გაყოფას** – გასაყოფისათვის გამყოფის შესაბამისი დამატებითი კოდის მრავალჯერადად მიმატების გზით. არითმეტიკული ოპერაციების შესრულების პროცესებს ამგვარი გამარტივება კომპიუტერში თვლის ორობითი სისტემის გამოყენების წყალობითაა მიღწეული.

4 მცურავი წერტილის (მძიმის) ფორმატში წარმოდგენილი რიცხვების არითმეტიკული შეკრებისა და გამოკლების დროს, უპირველეს ყოვლისა, შესაკრებების ხარისხები უნდა გათანაბრდეს. ხარისხების გათანაბრების პროცესში ხდება საკუთარ რეგისტრებში ჩაწერილი მცირე ხარისხების მქონე რიცხვთა მანტისების მარჯვნივ გარკვეული რაოდენობის თანრიგით ძვრა; ძვრის აღნიშნული რაოდენობა ოპერადების ხარისხების სხვაობის ტოლია. თითოეული ძვრის დროს ხარისხი ერთი ერთეულით იზრდება. ხარისხების გათანაბრების შემდეგ რიცხვების ერთანაირი სახელწოდების თანრიგები ორივე რეგისტრის შესაბამის თანრიგებში აღმოჩნდება განთავსებული; ამის შემდეგ ხდება მათი შეკრება თუ გამოკლება. საჭიროების შემთხვევაში მიღებული შედეგი ნორმალიზდება შედეგის მანტისის მარცხნივ ძვრის გზით. მარცხნივ ყოველ მომდევნო დაძვრისას შედეგის ხარისხი მცირდება ერთი ერთეულით. ქვემოთ განხილულ მაგალითებში გამოთვლების გასამარტივებლად **ხარისხი** ჩვეულებრივი **ორობითი ფორმითაა** წარმოდგენილი.

მაგალითი 3.15. შევკრიბოთ ორობითი ნორმალიზებული რიცხვები $\mathbf{0{\bullet}10111 \bullet 10^{-1}}$. და $\mathbf{0{\bullet}1101 \bullet 10^{10}}$. შესაკრებთა ხარისხების სხვაობა მოცემულ შემთხვევაში სამის ტოლია, ამიტომ შეკრების დაწყების წინ პირველი შესაკრების მანტისა მარჯვნივ სამი თანრიგით უნდა იქნეს დაძრული, რაც ხარისხის სამით გაზრდის ტოლფასია:

$$
\begin{array}{rl}
+ & 0\bullet00010111 \bullet 10^{10} \\
 & 0\bullet1101 \qquad\;\; \bullet 10^{10} \\
\hline
 & 0\bullet11101111 \bullet 10^{10}
\end{array}
$$

მაგალითი 3.16. ნორმალიზებულ $0.10101 \cdot 10^{10}$ რიცხვს გამოვაკლოთ რიცხვი $0.11101 \cdot 10^{1}$. მოცემულ შემთხვევაში საკლებისა და მაკლების ხარისხებს შორის სხვაობა ერთის ტოლია, ამიტომ გამოკლების დაწყებამდე მაკლების მანტისა საჭიროა ერთი თანრიგით მარჯვნივ დავძრათ:

$$\begin{array}{r l} + & 0.10101 \quad \cdot 10^{10} \\ & 0.011101 \cdot 10^{10} \\ \hline & 0.001101 \cdot 10^{10} \longleftarrow \text{არანორმალიზებული რიცხვი} \end{array}$$

ჯამად მივიღეთ არანორმალიზებული რიცხვი; მისი ნორმალიზებისათვის მანტისა მარცხნივ ორი თანრიგით უნდა დავძრათ, რაც ხარისხის ორი თანრიგით შემცირების ტოლფასია: $0.1101 \cdot 10^{0}$.

მაგალითი 3.17. შევასრულოთ ორი ორობითი ნორმალიზებული რიცხვის **გამრავლების ოპერაცია:**

$$(0.1110 \cdot 10^{101}) \cdot (0.1001 \cdot 10^{11}) = (0.1110 \cdot 0.1001) \cdot 10^{(101\,+\,11)} =$$
$$= 0.100000101 \cdot 10^{1000}$$

გაყოფის დროს გასაყოფის ხარისხს აკლდება გამყოფის ხარისხი, ხოლო მანტისებზე სრულდება გაყოფის ჩვეულებრივი ოპერაცია. განაყოფად არანორმალიზებული რიცხვის მიღებისას უნდა მოვახდინოთ მისი ნორმალიზება; ნორმალიზებისათვის მანტისა საჭიროა მარცხნივ დავძრათ, რითაც იცვლება ხარისხის მნიშვნელობა; კერძოდ, თითო თანრიგით მარცხნივ დაძვრისას ხარისხი თითო ერთეულით მცირდება. ასევე, მანტისის მარჯვნივ თითო თანრიგით დაძვრა თითო ერთეულით ზრდის ხარისხის მნიშვნელობას. ტერმინი **"მცურავი წერტილი (მძიმე)"** სწორედ იმიტომ იქნა შემოღებული, რომ რიცხვის გამოსახულებაში წერტილის (მძიმის) ფაქტობრივი ადგილმდებარეობის განმსაზღვრელი ორობითი ხარისხის მნიშვნელობა თითოეული არითმეტიკული ოპერაციის ჩატარების შემდეგ კორექტირდება, ე.ი. მოცემული ორობითი სიდიდის ცვლილების შესაბამისად რიცხვის გამოსახულებაში არსებული წერტილი (მძიმე) "ცურავს" (იცვლის ადგილმდებარეობას).

მაგალითი 3.18. შევასრულოთ ორი ნორმალიზებული რიცხვის გაყოფის ოპერაცია:

$$(0.1111 \cdot 10^{100}) : (0.101 \cdot 10^{11}) = (0.1111 : 0.101) \cdot 10^{(100\,-\,11)} =$$
$$= 1.1 \cdot 10^{1} = 0.11 \cdot 10^{10}$$

IV თ ა ვ ი
ინფორმატიკის ლოგიკური საფუძვლები

"ლოგიკა აზროვნების ანატომიაა"

ჯონ ლოკი (1632 - 1704)

"ლოგიკა მოაზროვნეთა ღმერთია"

ლიონ ფეიხტვაინგერი (1884 – 1958)

4.1. ლოგიკა დისკრეტული მოწყობილობის მათემატიკური მოდელის განსაზღვრის სამსახურში

1

ლოგიკა (ძველი ბერძნული **λóγος**-დან, "მეტყველება", "მსჯელობა", "აზრი") ეწოდება მეცნიერებას ინტელექტუალური შემეცნებითი მოღვაწეობის ფორმების, მეთოდებისა და კანონების შესახებ. ვინაიდან აღნიშნული ცოდნა აზროვნების მეშვეობით მიიღება, ამიტომ ლოგიკას ზოგჯერ აზროვნების შესახებ მეცნიერებადაც თვლიან.

აზროვნება მეტყველებაში მსჯელობის სახით განსხეულდება, რომლის კერძო გამოხატულებას რაიმის მტკიცება ან უარყოფა წარმოადგენს; აღნიშნულიდან გამომდინარე, ლოგიკა მსჯელობის (მტკიცებათა და უარყოფათა) მეცნიერებადაც შეიძლება მივიჩნიოთ.

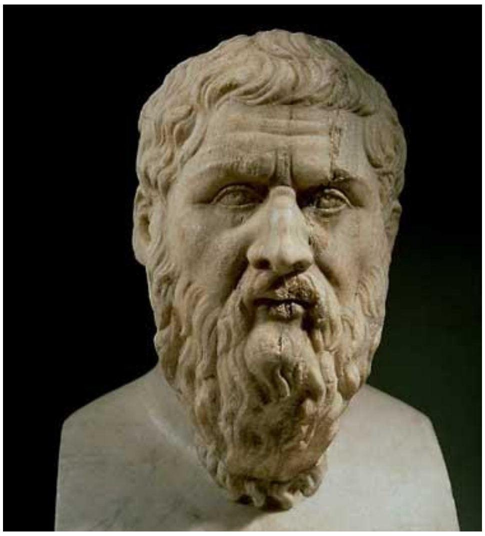
შეხებითი, მხედველობითი, გემოვნებითი, სმენითი და ყნოსვითი შეგრძნებების ერთობლიობას **შეგრძნებითი ცდა** ეწოდება. ცნობილი ავსტრიელი ექიმი, ანტროპოფოსი და რუდოლფ შტაინერის მიმდევარი **კარლ კიონინგი (1902-1966)** შეგრძნებით ცდას **საკუთარი თავის განცდად** განიხილავდა. ლოგიკას, როგორც მეცნიერებას, შემეცნების პროცესში ჭეშმარიტების მიღწევის ხერხები არა შეგრძნებითი ცდიდან, ანუ საკუთარი განცდებიდან, არამედ ადრე ფორმულირებული **ცოდნიდან** გამოჰყავს; ამიტომ **ლოგიკა** საწყისი ცოდნიდან – დასკვნითი ცოდნის მიღების მეცნიერებას წარმოადგენს.

დასკვნების მისაღებად ლოგიკის გამოყენების იდეა ჩაისახა ჩინეთში, ინდოეთსა და ძველ საბერძნეთში ჩამოყალიბებულ სამ ლოკალურ ცივილიზაციაში; აღნიშნულ ცივილიზაციებში ფორმულირებულ მოძღვრებებს შორის თანამედროვე მეცნიერებასა და მათემატიკაში ფართო გამოყენება ჰპოვა ძველი ბერძენი ფილოსოფოსის **არისტოტელეს** მიერ თავის ცნობილ ნაშრომში **"ორგანიონი"** ფორმულირებულმა მოძღვრებამ და დღეს სწორედ **არისტოტელე** ითვლება ლოგიკის მეცნიერების მამამთავრად, თუმცა შეუძლებელია ლოგიკის, როგორც მეცნიერების, ფორმირების პროცესში იგნორირებული იქნეს მეორე დიდი ბერძენი ფილოსოფოსის **პლატონის** ღვაწლი. მის არც ერთ ნაშრომში უშუალოდ ლოგიკის ცნება არ არის ფორმულირებული, მაგრამ მათში წამოჭრილი და გაანალიზებულია ლოგიკისათვის უმნიშვნელოვანესი შემდეგი სამი საკითხი:

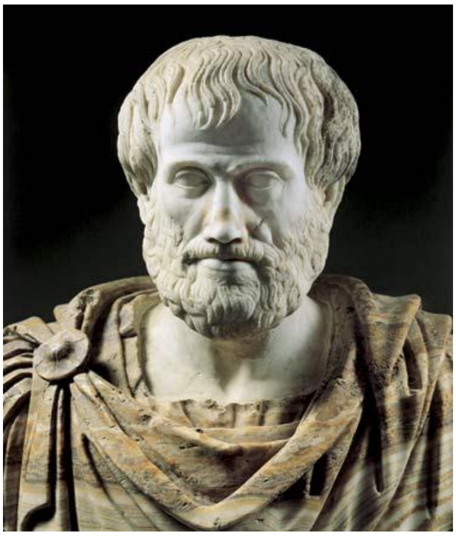
- რა შეიძლება ჩაითვალოს ჭეშმარიტებად და სიყალბედ;
- მსჯელობებსა და დასკვნებში როგორია წინამძღვრებს შორის არსებული კავშირების ბუნება;
- რა წარმოადგენს ცნებების არსს.

2

მსჯელობის დროს ჩვენ ვიყენებთ ცალკეულ გამოთქმებს. **გამოთქმა** წარმოადგენს გარკვეული წინადადების სახით ფორმულირებულ საწყის ცოდნას, რომლის შესახებაც შეიძლება ცალსახად ითქვას ჭეშმარიტია თუ ყალბია. მსჯლობის დროს გამოყენებული გამოთქმები პირობითად აღვნიშნოთ $x_1, x_2, \dots, x_n$ სიმბოლოებით; ამასთანავე ჩავთვალოთ, რომ თუ $x_i \in \{ x_1, x_2, \dots, x_n \}$ გამოთქმა ყალბია, მაშინ $x_i = 0$, ხოლო თუ ჭეშმარიტი – მაშინ $x_i = 1$.



პლატონი (ჩვენს წელთაღ-რიცხვამდე **428 – 347)**



არისტოტელე (ჩვენს წელთაღ-რიცხვამდე **384-322)**

აღნიშნული საწყისი ცოდნის, ანუ x_i გამოთქმების ანალიზის საფუძველზე ფორმირდება ახალი ცოდნა, ანუ **დასკვნითი გამოთქმა**. სიმარტივისათვის მას უბრალოდ **დასკვნა** ვუწოდოთ და პირობითად **y** სიმბოლოთი აღვნიშნოთ. აღნიშნული დასკვნა თუ ნამდვილად $x_1, x_2, \dots, x_n$ გამოთქმებისაგან მიიღება მაშინ ჩავთვალოთ, რომ $y = 1$, ხოლო წინააღმდეგ შემთხვევაში - $y = 1$;

ის ფაქტი, რომ **y** დასკვნა $x_1, x_2, \dots, x_n$ გამოთქმების ფუნქციას წარმოადგენს (ე.ი. $x_1, x_2, \dots, x_n$ გამოთქმები დამოუკიდებელი ცვლადები, ანუ არგუმენტებია), და თითოეული მათგანი იღებს **0**-ის ან **1**-ის ტოლ მნიშვნელობას, მათემატიკურად ჩაიწერება როგორც:

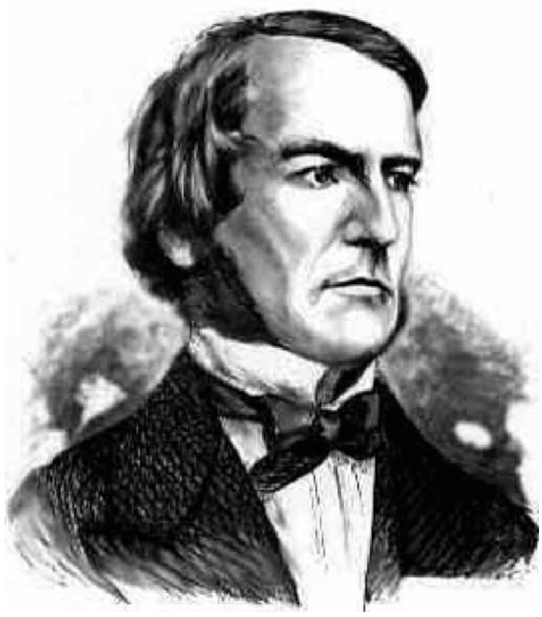
$$\mathbf{y} = f(x_1, x_2, \dots, x_n), \quad x_i\,;\, y \in \{0;1\}. \qquad \mathbf{(4.1)}$$

მიღებული (**4.1**) სახის ფუნქციას, რომელშიც როგორც დამოუკიდებელი ცვლადები (ანუ არგუმენტები) ისევე თავად ფუნქცია მხოლოდ **0**-ისა და **1**-ის მნიშვნელობებს იღებს, **ლოგიკური ფუნქცია** ეწოდება, ვინაიდან ისინი $x_1, x_2, \dots, x_n$ ცვლადებით აღნიშნული საწყისი გამოთქმებიდან **y** დასკვნის ლოგიკურად გამოყვანის საშუალებას იძლევა. ლოგიკური ფუნქციების შემსწავლელ ალგებრას **ლოგიკის ალგებრა** ეწოდება; სასკოლო ალგებრაში ცვლადები ნებისმიერ მნიშვნელობას იღებს; მისგან განსხვავებით ლოგიკის ალგებრაში მათ მხოლოდ ორი, კერძოდ **0**-ის ან **1**-ის ტოლი, მნიშვნელობის მიღება შეუძლია და ამიტომ აღნიშნულ ალგებრას ხშირად **ორობით ცვლადებს,** ხოლო მასში შესასწავლ ფუნქციებს – **ორობით (ლოგიკურ) ფუნქციებს** უწოდებენ.

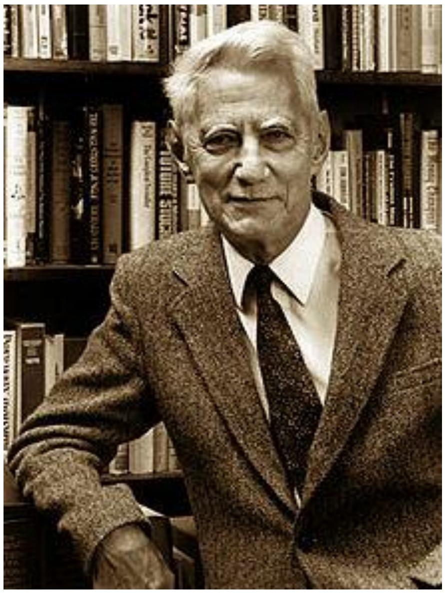
ლოგიკის ალგებრის ფუძემდებელია ცნობილი ინგლისელი (ირლანდიელი) მათემატიკოსი **ჯორჯ ბული**. ავტორის პატივსაცემად ლოგიკის ანუ ორობით ალგებრას **ბულის ალგებრის** სახელითაც მოიხსენიებენ.

ჯორჯ ბულის მიერ დამუშავებული ლოგიკის ალგებრა დიდი ხნის განმავლობაში წმინდა თეორიული ხასიათის ისეთ მოძღვრებად ითვლებოდა, რომელიც ტექნიკურ მეცნიერებებში არ გამოიყენებოდა.

ასეთი შეხედულება შეცვალა **დისკრეტული მოწყობილობების** გამოჩენამ, რომლის წყალობითაც გაფართოვდა ლოგიკის ალგებრის გამოყენების არეალი და მან წმინდა თეორიული მნიშვნელობის გარდა უდიდესი პრაქტიკული (გამოყენებითი) მნიშვნელობის მათემატიკური მოძღვრების სახელიც შეიძინა.



ჯორჯ ბული
(1815-1864)



კლოდ შენონი
(1916-2001)

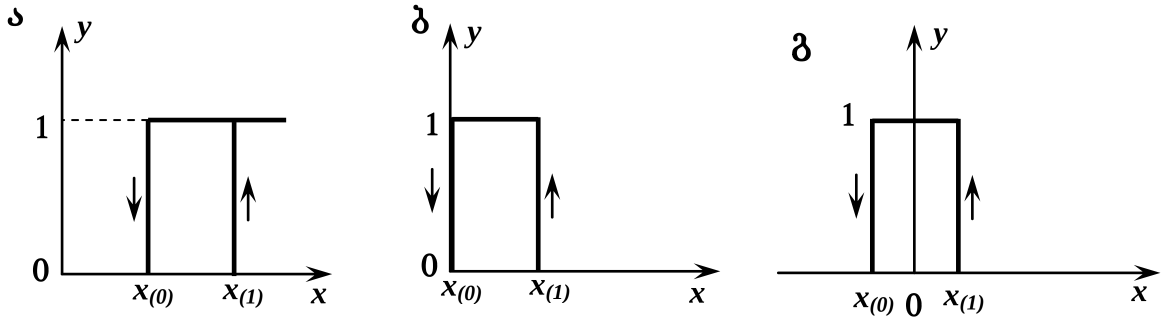
3

ავტომატური მართვის **სისტემის ელემენტი** ეწოდება გარკვეული სიგნალის გარდამქმნელ მის უმარტივეს ნაწილს. დასახული ამოცანების შესრულების მიზნით ფუნქციონურად გაერთიანებული ელემენტების ერთობლიობა წარმოქმნის **მოწყობილობას**. მის სტრუქტურაში არსებული ნებისმიერი ელემენტი წარმოადგენს შესასვლელი x სიგნალის y სიგნალად გარდამქმნელს. $y=f(x)$ თანაფარდობის სახეზე დამოკიდებულებით განასხვავებენ ანალოგურ და დისკრეტულ ელემენტებს.

ანალოგურ ელემენტებში შესასვლელი x სიგნალის მდოვრედ ცვლილებას თან ახლავს გამოსასვლელი y სიგნალის ასეთივე მდოვრე, უწყვეტი ცვლილება. **დისკრეტულ ელემენტებში** შესასვლელი სიგნალის მდოვრედ ცვლილების დროს გამოსასვლელი სიგნალი დისკრეტულად, ნახტომისებურად იცვლება და ერთ-ერთ გარკვეულ მნიშვნელობას მეორე მნიშვნელობით ცვლის. დისკრეტული ელემენტის განმასხვავებელ ნიშანს წარმოადგენს ის გარემოება, რომ მას აქვს სასრული რაოდენობის მდგომარეობა და ამ მდგომარეობას შესასვლელი სიგნალის სხვადასხვა დონეები შეესაბამება.

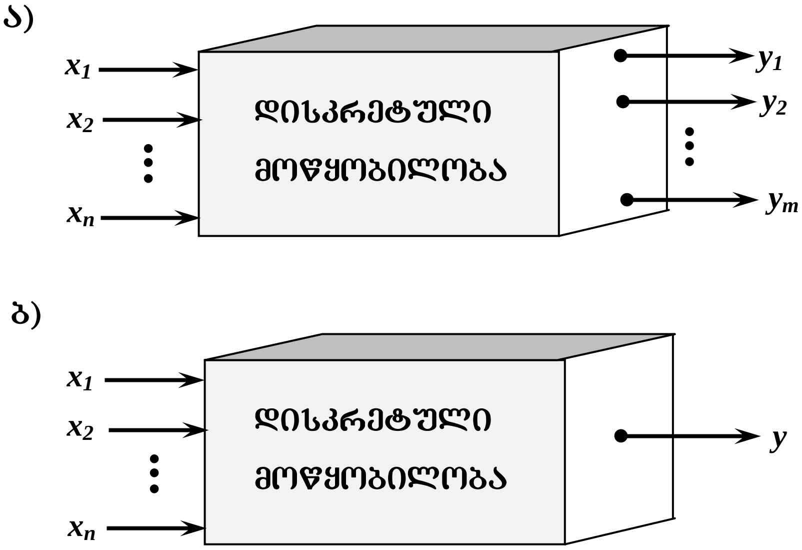
დისკრეტულ ელემენტებს, რომელებშიც შესასვლელი სიგნალის მდოვრედ ცვლილება იწვევს გამოსასვლელი სიგნალის ნახტომისებურ ცვლილებას, **რელეური მოქმედების დისკრეტული ელემენტები** ანუ, შემოკლებით, **რელეური ელემენტები** ეწოდება. რელეური ელემენტებისათვის დამახასიათებელი რელეური მახასიათებლები **4.1,ა** ნახაზზეა მოცემული. როგორც ნახაზიდან ჩანს, რელეური მახასიათებლის დროს შესასვლელი x სიდიდის ცვლილება იწვევს გამოსასვლელი y სიდიდის ნახტომისებურად ცვლილებას, რომელმაც შეიძლება მიიღოს მხოლოდ ორი მნიშვნელობა; ამ მნიშვნელობებიდან ერთ-ერთს, მაგალითად სიგნალის ამპლიტუდის დაბალ დონეს შეიძლება შევუთანადოთ სიმბოლო **0**, ხოლო მაღალ დონეს – სიმბოლო **1** (როგორც ეს **4.1,ა** ნახაზზეა ნაჩვენები), თუმცა შესაძლებელია პირიქითაც მოვიქცეთ.

რელეური ელემენტის კერძო შემთხვევას წარმოადგენს ე.წ. **მეხსიერების ელემენტი,** რომლის მახასიათებლები **4.1.,ბ,გ** ნახაზზეა მოყვანილი. აღნიშნული მახასიათებლების თანახმად შესასვლელი სიგნალის მოქმედების შეწყვეტის შემდეგ მეხსიერების ელემენტი წინა მდგომარეობაში რჩება; სხვა მდგომარეობაში გადასაყვანად მას ახალი შესასვლელი სიგნალი უნდა მიეწოდოს.



ნახ. 4.1. სხვადასხვა სახის რელეური მახასიათებლები

დისკრეტული მოწყობილობა ეწოდება დისკრეტული ელემენტებისაგან აგებულ მოწყობილობას, რომელიც დროის დისკრეტულ მომენტებში ფუნქციონირებს და ახდენს დისკრეტული სიგნალების დამუშავებას.



ნახ.4.2. დისკრეტული მოწყობილობების ბლოკ-სქემები

რელეური მახასიათებლების მქონე ელემენტებისაგან აგებული n რაოდენობის შესასვლელისა და m რაოდენობის გამოსასვლელის მქონე დისკრეტული მოწყობილობა შეიძლება წარმოვადგინოთ (n x m)-პოლუსას სახით (ნახ. **4.2.ა**). კერძო შემთხვევაში, როდესაც $m = 1$, იგი იღებს (n x 1)-პოლუსას სახეს (ნახ. **4.2,ბ**).

სიმარტივისათვის განვიხილოთ **4.2,ბ** ნახაზზე ნაჩვენები დისკრეტული მოწყობილობა. ვინაიდან იგი რელეური მახასიათებლების მქონე დისკრეტული ელემენტებისაგან არის აგებული, ამიტომ შესასვლელი x_i და გამოსასვლელი y სიგნალი იღებს ორ განსხვა-

ვებულ მნიშვნელობას, რომელთაგანაც ერთ-ერთს შეეთანადება სიმბოლო **0**, ხოლო მეორეს – სიმბოლო **1**, ე.ი. სამართლიანია გამოსახულება: $x_i\,, y \in \{0; 1\}$.

მოცემული დისკრეტული მოწყობილობის დანიშნულებაა შესასვლელი $x_1, x_2, \ldots, x_n$ სიგნალების გადამუშავების შედეგად მოახდინოს გამოსასვლელი y სიგნალის ფორმირება; მაშასადამე, იგი ახდენს შემდეგი მათემატიკული ფუნქციის რეალიზებას:

$$y = f\,(x_1, x_2, \ldots, x_n), \quad x_i\,; y \in \{\,0;1\}. \qquad (4.2)$$

მიღებული (**4.2**) გამოსახულება ემთხვევა ზემოთ განხილულ ლოგიკურ (**4.1**) გამოსახულებას, რაც საშუალებას გვაძლევს დავასკვნათ:

- რელეური ელემენტებისაგან აგებული დისკრეტული მოწყობილობა ახდენს გარკვეული სახის ლოგიკური ფუნქციის რეალიზებას;
- ნებისმიერი დისკრეტული მოწყობილობისათვის შეიძლება განისაზღვროს მის მიერ რეალიზებული ლოგიკური ფუნქცია და პირიქით, ნებისმიერი ლოგიკური ფუნქციისათვის შეიძლება აიგოს აღნიშნული ფუნქციის მარეალიზებელი დისკრეტული მოწყობილობა. მაშასადამე, დისკრეტულ მოწყობილობებსა და ლოგიკურ ფუნქციებს შორის შეიძლება დამყარდეს ურთიერთცალსახა დამოკიდებულება და ამიტომ **ლოგიკური ფუნქცია დისკრეტული მოწყობილობის მათემატიკური მოდელია**;

ასეთი გენიალური დასკვნების ფორმულირება მოახდინა ცნობილმა ამერიკელმა ინჟინერმა და მათემატიკოსმა **კლოდ შენონმა 1938** წელს გამოქვეყნებულ სტატიაში **"A Symbolic Analysis Relay and Switching Circuits"** ("რელეური და გადამრთველი სქემების სიმბოლური ანალიზი"). აღნიშნული სტატიის გამოქვეყნების შემდეგ ლოგიკის ალგებრა წმინდა თეორიული სახის მოძღვრებიდან უაღრესად დიდი პრაქტიკული მნიშვნელობის მოძღვრებად გადაიქცა.

5

კლოდ შენონის ზემოთ აღნიშნული ნაშრომის გამოქვეყნებამდე არ არსებობდა დისკრეტული მოწყობილობის აგების ფორმალური მეთოდი და მისი კონსტრუირება მთლიანად იმ ცოდნით, გამოცდილებითა და ოსტატობით იყო შესაძლებელი, რომლებიც დისკრეტული მოწყობილობის შემქნელებს თითქმის ხელოვნების დონემდე ჰქონდათ აყვანილი.

კლოდ შენონის გენიალურული მიგნების შედეგად ნაპოვნი იქნა დისკრეტული მოწყობილობის მათემატიკური მოდელი და აღნიშნული მოწყობილობის აგების ამოცანა დაყვანილი იქნა ლოგიკური ფუნქციის ტექნიკური რეალიზაციის ფორმალური მეთოდებით გადასაწყვეტ ამოცანამდე. მან შესაძლებელი გახადა დამუშავებულიყო დისკრეტული **მოწყობილობის აგების ფორმალური მეთოდი**, რომლის ზოგადი სახე ასეთია:

- სიტყვიერად იქნეს ფორმულირებული დისკრეტული მოწყობილობის ფუნქციონირების ალგორითმი;
- ფორმულირებული ალგორითმის მიხედვით განისაზღვროს დისკრეტული მოწყობილობის მიერ სარეალიზებელი ლოგიკური ფუნქციები;
- მოხდეს მიღებული ლოგიკური ფუნქციების მინიმიზირება;
- მინიმიზირებული ლოგიკური ფუნქციების გამოყენებით მოხდეს დისკრეტული მოწყობილობის სინთეზი.

6

მასაჩუსეტის უნივერსიტეტის პროფესორის **ს.კოლდუელის** აზრით პირველ რელეურ (დისკრეტულ) მოწყობილობებს წარმოადგენდა მე-**19** საუკუნის პირველ ნახევარში სარკინიგზო ტრანსპორტის მართვისათვის შექმნილი **სიგნალიზაციისა და ბლოკირების მოწყობილობები** [17], რომლებსაც დღეს **სარკინიგზო ავტომატიკისა და**

ტელემექანიკის მოწყობილობები ეწოდება. შემდეგში ანალოგიური მოწყობილობები შეიქმნა საჰაერო ტრანსპორტის მოძრაობის მართვისათვისაც.

თავდაპირველად დისკრეტული მოწყობილობების ასაგებად გამოყენებული საელემენტო ბაზა შედგებოდა წმინდა მექანიკური მოქმედების ელემენტებისაგან, რაც ზღუდავდა აღნიშნული მოწყობილობების ფუნქციონალურ შესაძლებლობებს. საელემენტო ბაზის სრულყოფილების კვალდაკვალ ფართოვდებოდა დისკრეტული მოწყობილობების ფუნქციონალური შესაძლებლობები და მათი ტექნიკურ-ეკონომიკური მახასიათებლები;

დისკრეტული მოწყობილობების ასაგებად გამოყენებულ საეტაპო მნიშვნელობის ელემენტებს, რომლებმაც რევოლუციური როლი ითამაშეს აღნიშნული მოწყობილობების განვითარების პროცესში, წარმოადგეს ელექტრომაგნიტური რელეები, ტრანზისტორები და ინტეგრალური სქემები; აღნიშნულიდან გამომდინარე რამდენიმე სიტყვით შევეხებით მათ.

- **1935** წელს ამერიკელი ფიზიკოსის **ჯოსეფ ჰენრის** მიერ **ელექტრომექანიკური რელეს** გამოგონებამ ისეთი დისკრეტული მოწყობილობების წინაპირობები შექმნა, რომელთა გამოყენებითაც კონსტრუირებული იქნა ე.წ. **რელეური, ანუ პირველი თაობის კომპიუტერები;** ისინი რელეური დისკრეტული მოწყობილობებისაგან აგებულ უნივერსალურ დისკრეტულ სისტემებს წარმოადგენდა. მაგალითად, **1939** წელს გერმანელი მეცნიერის **კ. ცუზეს** მიერ კონსტრუირებული დაპროგრამებადი უნივერსალური ციფრული კომპიუტერი **Z3** შეიცავდა დაახლოებით **2600** რელეს, რომელთაგანაც **1400** რელე გამოიყენებოდა მეხსიერების, **600** რელე – არითმეტიკული მოდულის, ხოლო დანარჩენი რელეები – მართვის მოწყობილობის ასაგებად. რელეური საელემენტო ბაზის გამოყენებით შექმნილი პირველი თაობის კომპიუტერებია: **ENIAC** (**აშშ**), **БЭСМ-1, «Урал»** (**საბჭოთა კავშირი**) და ა.შ.
- **1947** წლის **16** დეკემბერს ამერიკელი მეცნიერების **უ. ბრატეინის, ჯ. ბარდინისა** და **უ.შოკლის** მიერ გამოგონებული იქნა ტრანზისტორი; **ტრანზისტორი** წარმოადგენს მცირე ზომის ელექტრონულ მოწყობილობას, რომელიც ელექტრომექანიკური რელეს კონტაქტის მსგავსად ელექტრულ წრედში ელექტრული დენის ჩართვა-გამორთვის მიზნით შეიძლება იქნეს გამოყენებული.

ზემოთ აღნიშნულ ფუნქციებს რელეს კონტაქტი ახდენს ელექტრული წრედის მექანიკური შერთვისა და გაწყვეტის, ხოლო **ტრანზისტორი** – ელექტრონული შერთვისა და გაწყვეტის გზით.

წრედის მექანიკური გაწყვეტის მოვლენა განმარტებას არ საჭიროებს; რაც შეეხება ელექტრონულ გაწყვეტას, მისი არსი ასეთია: წრედში ჩართულ ტრანზისტორს შეუძლია მიიღოს ნულის ან უსასრულობის ტოლი წინაღობა. პირველ შემთხვევაში წრედი დენს დაუბრკოლებლად ატარებს, ე.ი. წრედი დენისათვის ‘‘შერთულია’’, ხოლო მეორე შემთხვევაში წრედში გამავალი მაქსიმალურად მცირდება და პრაქტიკულად ნულის ტოლი ხდება, ე.ი. წრედი დენისათვის ‘‘გაწყვეტილია’’.

რადგან წრედის კომუტირების თვალსაზრისით, ტრანზისტორი ელექტრომექანიკური რელეს კონტაქტების ანალოგურ ფუნქციებს ასრულებს, ამიტომ სქემების ასაგებად კონტაქტების ნაცვლად შეიძლება ტრანზისტორები გამოვიყენოთ.

გასული საუკუნის **60-70**-იან წლებში ტრანზისტორების გამოყენებით აგებულ კომპიუტერებს **მეორე თაობის კომპიუტერებს** უწოდებენ.

პირველი თაობის ანუ რელეური კომპიუტერებისაგან განსხვავებით, მეორე თაობის ტრანზისტორული კომპიუტერები გამოირჩევა შემცირებული ზომებით, გაფართოებული ფუნქციონალური შესაძლებლობებითა და უკეთესი ტექნიკურ-ეკონომიკური მახასიათებლებით. აღნიშნული თაობის კომპიუტერებია **IBM 7090, LARC, Stretch** (**აშშ**), **Atlas** (ინგლისი), **МЭСМ, «Раздан», «Минск-32** (საბჭოთა კავშირი) და ა.შ. ტიპის კომპიუტერები.

● **1958** წლის **12** სექტემბერს ამერიკელი მეცნიერების **ჯ. კილბისა** და **რ. ნოისის** მიერ გამოგონებული იქნა **ინტეგრალური სქემები**. ინტეგრალური სქემა წარმოადგენს (**1,3x1,3**)-დან (**13x13**)-მდე ზომის კრისტალს, რომელშიც "შეყურსულია" რამდენიმე ათეულიდან დაწყებული რამდენიმე ასეულ მილიონამდე დამთავრებული ისეთი სხვადასხვა ელექტრონული ელემენტები, როგორებიცაა მაგალითად, ტრანზისტორი, დიოდი, რეზისტორი და ა.შ.

განასხვავებენ:

მცირე ინტეგრაციის მქონე ინტეგრალურ სქემებს, რომელთა ერთ კრისტალში **100**-მდე ელემენტია მოთავსებული;

საშუალო ინტეგრაციის მქონე ინტეგრალურ სქემებს, რომელთა ერთ კრისტალში **1000**-მდე ელემენტია მოთავსებული;

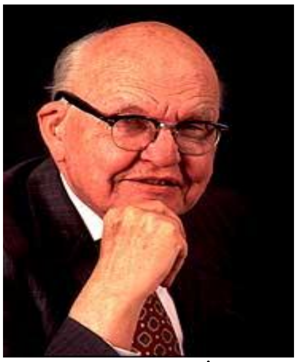
დიდი ინტეგრაციის მქონე ინტეგრალურ სქემებს, რომელთა ერთ კრისტალში **10000**-მდე ელემენტია მოთავსებული;

დისკრეტული მოწყობილობების საელემენტო ბაზის გამომგონებელთა გალერეა

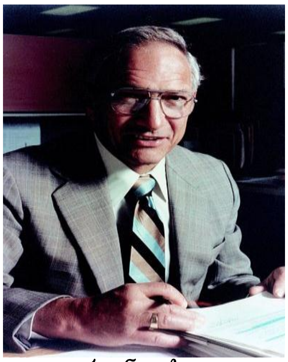
რელეს გამომგონებელი

ჯ. ჰენრი
(1858-1943)

ინტეგრალური სქემის გამომგონებლები



ჯ. კილბი
(1923-2005)

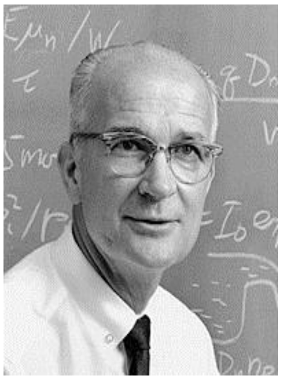


რ. ნოისი
(1927-1990)

ტრანზისტორის გამომგონებლები

უ. ბრატეინი
(1902-1987)

ჯ. ბარდინი
(1908-1991)



უ. შოკლი
(1910-1989)

ზედიდი ინტეგრაციის მქონე ინტეგრალურ სქემებს, რომელთა ერთ კრისტალში **10000**-ზე მეტი ელემენტია მოთავსებული;

მაგალითად, პროცესორი **Pentium 4** წარმოადგენს დაპროგრამებად ინტეგრალურ სქემას, რომელიც **3,1** მილიონზე მეტ ტრანზისტორს შეიცავს.

მცირე და საშუალო ინტეგრაციის მქონე ინტეგრალური სქემებით კომპიუტერების აგება დაიწყო **XX** საუკუნის **70**-იანი წლების დასასრულსა და **80**-იანი წლების დასაწყისში. მათ **მესამე თაობის კომპიუტერები** ეწოდება. ასეთებია **IBM 360** (აშშ), **EC ЭВМ** (საბჭოთა კავშირი) და ა.შ ტიპის კომპიუტერები.

დიდი და ზედიდი ინტეგრაციის მქონე ინტეგრალური სქემებით კომპიუტერების აგება დაიწყო **XX** საუკუნის **80**-იანი წლების დასასრულს და მათ **მეოთხე თაობის კომპიუტერები** ეწოდა.

მეოთხე თაობის კომპიუტერები ორი მიმართულებით განვითარდა. განვიხილოთ ისინი.

პირველი მიმართულება უკავშირდება სუპერკომპიუტერების შექმნას; სუპერკომპიუტერები წარმოადგენს მრავალპროცესორულ მანქანებს, რომელთა სწრაფმოქმედებაა წამში რამდენიმე მილიარდი ოპერაციის შესრულება; ასეთი სუპერკომპიუტერებია **ILLIAS 4, Cray** (აშშ), **Эльбрус-2** (საბჭოთა კავშირი) და ა.შ ტიპის კომპიუტერები.

მეორე მიმართულება დიდი და ზედიდი ინტეგრალური სქემების გამოყენებით მიკრო- და პერსონალური კომპიუტერების შექმნასთანაა დაკავშირებული. ასეთი კომპიუტერების პირველი წარმომადგენლებია **Apple, IBM PC** და ა.შ. ტიპის კომპიუტრები.

4.2. ლოგიკური ფუნქციების ზოგადი დახასიათება. ელემენტალური ლოგიკური ფუნქციები

1 n რაოდენობის ორობით $x_1, x_2, \dots, x_n$ $(x_i \in 0;1))$ ცვლადებზე დამოკიდებულ ორობით

$$y = f(x_1, x_2, \dots, x_n), \qquad x_i\,; y \in \{0;1\}. \qquad (4.3)$$

ფუნქციას **ლოგიკური** (**ორობითი, ან ბულის**) **ფუნქცია** ეწოდება.

n რაოდენობის არგუმენტებზე დამოკიდებული ლოგიკური ფუნქციის არგუმენტების ნებისმიერი ნაკრები წარმოადგენს გარკვეულ n-თანრიგიან ორობითი რიცხვს; რადგან არსებობს 2^n რაოდენობის ასეთი რიცხვი, ამიტომ n რაოდენობის არგუმენტებზე დამოკიდებული არგუმენტების მნიშვნელობათა ნაკრებების რაოდენობაც 2^n-ის ტოლია. მაგალითად:

- $n=2$ რაოდენობის არგუმენტზე დამოკიდებული ლოგიკური $y=f(x_1,x_2)$ ფუნქციის არგუმენტების მნიშვნელობათა ნაკრებების რაოდენობა 2^2=**4**-ის ტოლია (ეს ნაკრებებია **00; 01; 10; 11**)
- $n=3$ რაოდენობის არგუმენტზე დამოკიდებული ლოგიკური $y=f(x_1,x_2, x_3,)$ ფუნქციის არგუმენტების მნიშვნელობათა ნაკრებების რაოდენობა 2^3=**4**-ის ტოლია (ეს ნაკრებებია **000; 001; 010 011; 100, 101, 110, 111**) და ა.შ.

$x_1, x_2, \dots, x_n$ არგუმენტების მნიშვნელობების თითოეულ ნაკრებზე ლოგიკური y ფუნ-ქცია იღებს ორ, კერძოდ **0**-ისა და **1**-ის ტოლ მნიშვნელობებს, ამიტომ:

n რაოდენობის არგუმენტებზე დამოკიდებული ლოგიკური ფუნქციების საერთო **N** რაოდენობა განისაზღვრება ფორმულით:

$$N = 2^{2^n} \qquad (4.4)$$

n=0; 1; 2; 3; 4; 5 არგუმენტებზე დამოკიდებული ლოგიკური $y=f(x_1,x_2, \dots, x_n)$ ფუნქციის **N** რაოდენობები **4.1** ცხრილშია მოყვანილი.

ცხრ.4.1. ფუნქციური **N =** ***f(n)*** დამოკიდებულება

n	0	1	2	3	4	5
N	2	4	16	256	65 536	4 294 967 296

როგორც ზემოთ აღვნიშნეთ, ლოგიკურ ფუნქციებსა და დისკრეტულ მოწყობილობებს შორის ურთიერთცალსახა დამოკიდებულება არსებობს, რაც შემდეგს ნიშნავს:

ნებისმიერი ლოგიკური ფუნქციისათვის შეიძლება ავაგოთ მისი მარეალიზებელი დისკრეტული მოწყობილობა და, პირიქით, ნებისმიერი დისკრეტული მოწყობილობისათვის არსებობს მის მიერ რეალიზებული გარკვეული ლოგიკური ფუნქცია.

ლოგიკური ფუნქციების მიხედვით დისკრეტული მოწყობილობის აგებას **დისკრეტული მოწყობილობის სინთეზი** ეწოდება, ხოლო არსებული დისკრეტული მოწყობილობისათვის მის მიერ რეალიზებული ლოგიკური ფუნქციის განსაზღვრას – **დისკრეტული მოწყობილობის ანალიზი.**

2

სპეციფიკური სახის ლოგიკურ ფუნქციებს წარმოადგენს ***n***=0 რაოდეობის არგუმენტზე დამოკიდებული ლოგიკური ფუნქციები, რომელთა რაოდენობა 2-ის ტოლია (ცხრ.**4.1**). ასეთ ფუნქციებს წარმოადგენს არც ერთ არგუმენტზე დამოკიდებული ლოგიკური ფუნქციები, ანუ **კონსტანტა 0** და **კონსტანტა 1** (მუდმივა **0** და მუდმივა 1). ისინი შესაბამისად აღვნიშნოთ f_1 და f_2 სიმბოლოებით, ე.ი. f_1=**0** და f_2=**1** . კონსტანტა **0**-ს შეესაბამება მუდმივად განრთული ელექტრული სადენი, რომელშიც დენი არ გადის (ცხადია, სადენი შეიძლება განრთული იყოს მექანიკურად ან ელექტრულად), ხოლო კონსტანტა **1**-ს – მუდმივად შერთული ელექტრული სადენი, რომელშიც დენი მუდმივად გადის.

n რაოდენობის შესასვლელების მქონე დისკრეტულ მოწყობილობას (**იხ.ნახ.4.2**) შეესაბამება ***n*** რაოდენობის არგუმენტებზე დამოკიდებული ლოგიკური ფუნქცია. ე.ი. ასეთი მოწყობილობის სინთეზისათვის საჭიროა განვსაზღვროთ ***n*** რაოდენობის არგუმენტებზე დამოკიდებული ლოგიკური ფუნქცია

4.1 ცხრილიდან ჩანს, რომ არგუმენტების რაოდენობის უმნიშვნელო ცვლილება იწვევს ლოგიკური ფუნქციების რაოდენობის მნიშვნელოვან ცვლილებას; კერძოდ, ლოგიკური ფუნქციების რაოდენობა არგუმენტების:

2-იდან 3-მდე გაზრდით იზრდება 16-ჯერ;

3-იდან 4-მდე გაზრდით იზრდება 256-ჯერ;

4-იდან 5-მდე გაზრდით იზრდება 65536-ჯერ და მიიღება იმდენად კოლოსალური რაოდენობის (4 294 967 296) ლოგიკური ფუნქციები, რომ მათი ინდივიდუალური განხილვა ურთულეს ამოცანას წარმოადგენს. აღნიშნულმა გარემოებამ ერთი შეხედვით შეიძლება წარმოშვას დისკრეტული მოწყობილობების სინთეზისათვის ლოგიკური ფუნქციების გამოყენების მეთოდის არაპრაქტიკულობის ილუზია; მაგრამ ეს მხოლოდ ერთი შეხედვით, რადგან აღმოჩნდა, რომ:

ნებისმიერი სირთულის ლოგიკური ფუნქცია შეიძლება გამოვსახოთ **0-**, **1-** და **2-** რაოდენობის არგუმენტებზე დამოკიდებული ლოგიკური ფუნქციებით, რომელთა რაოდენობა 12-ის ტოლია და რომლებსაც **ელემენტალური ლოგიკური ფუნქციები** ეწოდება.

ზემოთ აღნიშნულიდან გამომდინარე, ნებისმიერი სირთულის მქონე დისკრეტული მოწყობილობების როგორც სინთეზის, ასევე ანალიზის ამოცანის გადაწყვეტა შესაძლებელია 12 ელემენტალური ლოგიკური ფუნქციის დახმარებით. გავეცნოთ თითოეულ მათგანს.

0 რაოდენობის არგუმენტებზე დამოკიდებული ლოგიკური ფუნქციები ზემოთ განვიხილეთ.

ცხრილს, რომლის საშუალებითაც ლოგიკური ფუნქციები გამოისახება, **ჭეშმარიტობის ცხრილი** ეწოდება.

n რაოდენობის არგუმენტებზე დამოკიდებული ლოგიკური ფუნქციის გამომსახველი **ჭეშმარიტობის ცხრილი** შედგება 2^n რაოდენობის მწკრივისაგან; მის მარცხენა ნაწილში ჩამოწერილია არგუმენტების მნიშვნელობათა ყველა ნაკრები და თითოეული ნაკრების გასწვრივ მარჯვენა ნაწილში მითითებულია ის მნიშვნელობა, რომელსაც ფუნქცია არგუმენტების მოცემულ ნაკრებზე იღებს.

ერთ არგუმენტზე დამოკიდებული ლოგიკური $y=f(x)$ ფუნქციების ჭეშმარიტობის ცხრილს აქვს **4.2** ცხრილის სახე. როგორც აღნიშნული ცხრილიდან ჩანს, არსებობს ერთ ცვლადზე დამოკიდებული ლოგიკური $y = f'_1(x)$, $y = f'_2(x)$, $y = f_3(x)$ და $y = f_4(x)$ ფუნქციები. განვიხილოთ თითოეული მათგანი.

ცხრ. 4.2. $y=f(x)$ ფუნქციების ჭეშმარიტობის ცხრილი

x	y			
	f'_1	f'_2	f_3	f_4
0	0	1	0	1
1	0	1	1	0

$y = f'_1(x)$ ფუნქცია არგუმენტის ორივე მნიშვნელობაზე იღებს ერთი და იგივე, კერძოდ **0**-ის ტოლ მნიშვნელობას; ე.ი. არგუმენტის მნიშვნელობის ცვლა გავლენას ვერ ახდენს ფუნქციის მნიშვნელობაზე. ასეთ არგუმენტს **ფიქტიური არგუმენტი,** ხოლო ფიქტიური არგუმენტის შემცველ ფუნქციას – **გადაგვარებული ლოგიკური ფუნქცია** ეწოდება.

გადაგვარებული ლოგიკური ფუნქციიდან შეიძლება ფიქტიური არგუმენტი გამოირიცხოს, რის შედეგადაც მიიღება არგუმენტებზე არსებითად დამოკიდებული ფუნქცია. **არგუმენტებზე არსებითად დამოკიდებული ლოგიკური ფუნქცია** ეწოდება ისეთ ლოგიკურ ფუნქციას, რომლის ნებისმიერი არგუმენტის მნიშვნელობის შეცვლა ცვლის ფუნქციის მნიშვნელობას.

არგუმენტებზე არსებითად დამოკიდებულ ფუნქციად გადაგვარებული ფუნქციის გარდაქმნის გზით შესაძლებელია შემცირდეს მასში შემავალი არგუმენტების რაოდენობა, ანუ მოხდეს არგუმენტების რაოდენობის მინიმიზირება. მინიმალური რაოდენობის არგუმენტების შემცველ ლოგიკურ ფუნქციას **მინიმალური ლოგიკური ფუნქცია ეწოდება.**

ზემოთ აღნიშნულიდან გამომდინარე, ლოგიკური $y = f'_1(x)$ ფუნქცია გადაგვარებული ფუნქციაა; იგი შეიცავს ერთადერთ არგუმენტს და იგიც ფიქტიურია; მისი გამორიცხვით მივიღებთ **0** არგუმენტზე დამოკიდებულ $f_1=0$ ფუნქციას, ანუ კონსტანტა **0**-ს.

გადაგვარებულია ლოგიკური $y = f'_2(x)$ ფუნქციაც, რადგან არგუმენტის მნიშვნელობისაგან დამოუკიდებლად იგი **1**-ის ტოლ მნიშვნელობას იღებს. ფიქტიური არგუმენტის გამორიცხვის შედეგად იგი **0** არგუმენტზე დამოკიდებულ $f_1=1$ ფუნქციად, ანუ კონსტანტა 1-ად გადაიქცევა.

ლოგიკური $y = f_3(x)$ ფუნქცია იმეორებს x არგუმენტის მნიშვნელობას, ამიტომ მას **გამეორების ფუნქცია** ეწოდება და შემდეგნაირად აღინიშნება: $y = x$.

ლოგიკური $y = f_4(x)$ ფუნქცია იღებს x არგუმენტის მნიშვნელობის შებრუნებულ მნიშვნელობას, ანუ ახდენს არგუმენტის მნიშვნელობის უარყოფას; ამიტომ მას **უარყოფის**

ფუნქცია ანუ **ინვერსია** (ლათ. **inversion** – "გადასმა") ეწოდება; იგი აღინიშნება, როგორც $y = \overline{x}$ და ასე იკითხება: **"y უდრის არა x –ს".**

ცხრ. 4.3. ორ არგუმენტზე დამოკიდებული ლოგიკური $y = f(x_1, x_2)$ ფუნქციები

x_1	x_2	y															
		f_1'	f_5	f_6	f_3'	f_7	f_3''	f_8	f_9	f_{10}	f_{11}	f_4'	f_{12}	f_4''	f_{13}	f_{14}	f_2'
0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
0	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**1**	**1**	**1**	**1**
1	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	**1**
1	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**

3 ორ არგუმენტზე (**n = 2**) დამოკიდებული $y = f(x_1, x_2)$ ფუნქციების ჭეშმარიტობის ცხრილში (ცხრილი **4.3**) არსებული **16** ლოგიკური ფუნქციიდან **6** ფუნცია გადაგვარებულია; კერძოდ, გადაგვარებულია f_1', f', f_3', f_3'', f_4', f_4'' ფუნქციები.

● გადაგვარებულ f_1' და f_2' ფუნქციებში არსებული ორივე არგუმენტი ფიქტიურია, რადგან მათი ნებისმიერი მნიშვნელობის დროს პირველი იღებს **0**-ის, ხოლო მეორე **1**-ის ტოლ მნიშვნელობას; აღნიშნული ფიქტიური არგუმენტების გამორიცხვით ლოგიკური f_1' ფუნქცია გადაიქცევა **0**-არგუმენტზე დამოკიდებულ f_1 ფუნქციად, ანუ **კონსტანტა 0-ად,** ხოლო ლოგიკური f_2' ფუნქცია – f_2 ფუნქციად, ანუ **კონსტანტა 1-ად.**

● ლოგიკური f_3' ფუნქცია წარმოადგენს x_1 არგუმენტის გამეორებას, ხოლო მასში არსებული x_2 არგუმენტი ფიქტიურია. ასევე, f_3'' ფუნქცია წარმოადგენს x_2 არგუმენტის გამეორებას და მასში არსებული x_1 არგუმენტი ფიქტიურია. ფიქტიური არგუმენტების გამორიცხვის შედეგად პირველი გადაიქცევა x_1 არგუმენტის გამეორებად, ხოლო მეორე – x_2 არგუმენტის **გამეორების** ფუნქციად. მაშასადამე თითოეული მათგანი წარმოადგენს ადრე განხილულ **გამეორების ფუნქციას,** რომელიც ზემოთ განვიხილეთ.

● ლოგიკური f_4' ფუნქცია წარმოადგენს x_2 არგუმენტის უარყოფას, ხოლო მასში არსებული x_1 არგუმენტი ფიქტიურია; ასევე, ლოგიკური f_4'' ფუნქცია წარმოადგენს x_1 არგუმენტის უარყოფას, ხოლო მასში არსებული x_2 არგუმენტი ფიქტიურია. ფიქტიური არგუმეტების გამორიცხვის შედეგად პირველი გადაიქცევა x_2 არგუმენტის, ხოლო მეორე – x_1 არგუმენტის **უარყოფის** ფუნქციად. მაშასადამე, თითოეული მათგანი წარმოადგეს ადრე განხილულ **უარყოფის ფუნქციას,** ანუ **ინვერსიას.**

ლოგიკური ცვლადი ითვლება **ჭეშმარიტად**, თუ მისი მნიშვნელობა **1**-ის ტოლია და **ყალბად,** თუ მისი მნიშვნელობა **0**-ის ტოლია. აღნიშნული ტერმინები გამოვიყენოთ **4.3** ცხრილში არსებული დარჩენილი ლოგიკური ფუნქციების განხილვისას.

● ლოგიკური f_5 ფუნქცია (ცხრ. **4.3**) ჭეშმარიტია მაშინ და მხოლოდ მაშინ, როდესაც ჭეშმარიტია როგორც x_1, ასევე x_2 არგუმენტი; ნებისმიერ დანარჩენ შემთხვევაში იგი ყალბია. მოცემულ ფუნქციას აქვს რამდენიმე სახელწოდება; კერძოდ მას უწოდებენ **კონიუნქციას**, ანუ **ლოგიკურ გამრავლებას**, ანუ **თანხვდენის ფუნქციას;** რადგან აღნიშნული ფუნქციის ჭეშმარიტობის დროს ჭეშმარიტია x_1 **და** x_2 არგუმენტი, ამიტომ ხშირად მას **და** ფუნქციასაც უწოდებენ. არსებობს აღნიშნული ფუნქციის რამდენიმე სიმბოლური აღნიშვნა; კერძოდ, მას აღნიშნავენ შემდეგნაირად: $y = x_1 \& x_2$, ან $y = x_1 \wedge x_2$ ან

$y = x_1 \bullet x_2$, ან $y = x_1 x_2$;

● ლოგიკური f_8 ფუნქცია (ცხრ. **4.3**) მაშინ არის ჭეშმარიტი, როდესაც x_1 და x_2 არგუმენტებიდან ერთ-ერთი არგუმენტი ჭეშმარიტია, ხოლო მეორე ყალბი; მოცემულ ფუნქციას ეწოდება **ორის მოდულით შეკრების ფუნქცია** და აღინიშნება შემდეგნაირად: $y = x_1 \oplus x_2$**.** მოცემული ფუნქცია როდესაც ჭეშმარიტია მაშინ მისი არგუმენტები სხვადასხვა მნიშვნელობებს იღებს, ამიტომ ზოგჯერ მას **არაერთმნიშვნელიანობის ფუნქციასაც** უწოდებენ; იგი ცნობილია "**გამომრიცხავი ან ფუნქციის**" სახელწოდებითაც, რადგან ჭეშმარიტია, მისი ჭეშმარიტობის დროს გამოირიცხება ორივე არგუმენტის ერთდროულად ჭეშმარიტობა (1-ის ტოლობა): ჭეშმარიტი უნდა იყოს **ან** ერთი, **ან** მეორე არგუმენტი.

● ლოგიკური f_9 ფუნქცია (ცხრ. **4.3**) ჭეშმარიტია მაშინ, როდესაც x_1 და x_2 არგუმენტებიდან ერთ-ერთი არგუმენტი მაინც არის ჭეშმარიტი; იგი ყალბი ხდება მაშინ და მხოლოდ მაშინ, როდესაც მისი ყველა არგუმენტი ყალბია. მოცემულ ფუნქციას უწოდებენ **დიზიუნქციას**, ან **ლოგიკურ შეკრების ფუნქციას.** რადგან მოცემული ფუნქციის ჭეშმარიტობის დროს ჭეშმარიტია x_1 **ან** x_2 არგუმენტი, ამიტომ ხშირად მას **ან** ფუნქციასაც უწოდებენ. განხილული ფუნქცია აღინიშნება როგორც $y = x_1 \vee x_2$ ან $y = x_1 + x_2$;

● ლოგიკური f_{10} ფუნქცია (ცხრ. **4.3**) იღებს დიზიუნქციის f_9 ფუნქციის მნიშვნელობათა შებრუნებულ მნიშვნელობებს, ამიტომ მას **დიზიუნქციის უარყოფის ფუნქციას** უწოდებენ. ხშირად მას **ან-არა ფუნქციის** სახელწოდებითაც მოიხსენიებენ, ხოლო მათემატიკურ ლიტერატურაში იგი ცნობილია **ვების ფუნქციის**, აგრეთვე **პირსის ისრის** სახელწოდებითაც. აღინიშნება შემდეგნაირად: $y = \overline{x1 \vee x2}$, ან $y = \overline{x1 + x2}$, ან $y = x_1 \downarrow x_2$.

● ლოგიკური f_{11} ფუნქცია (ცხრ. **4.3**) მაშინ არის ჭეშმარიტი, როდესაც x_1 და x_2 არგუმენტებიდან ორივე ჭეშმარიტია ან ორივე ყალბი. აღნიშნულ ფუნქციას **ეკვივალენტობის ფუნქცია** ეწოდება და აღინიშნება როგორც $y = x_1 \sim x_2$. მოცემული ფუნქცია როდესაც ჭეშმარიტია მაშინ მისი არგუმენტები ერთნაირ მნიშვნელობას იღებენ, ამიტომ ზოგჯერ მას **ერთნაირმნიშვნელიანობის ფუნქციასაც** უწოდებენ;

● ლოგიკური f_{12} და f_{13} ფუნქციებს (ცხრ. **4.3**) **იმპლიკაციები** ეწოდება. პირველი მათგანი აღინიშნება როგორც $y = x_1 \rightarrow x_2$ (იკითხება: "თუ x_1, მაშინ x_2"), ხოლო მეორე მათგანი როგორც $y = x_2 \rightarrow x_1$ (იკითხება: "თუ x_2, მაშინ x_1").

● ლოგიკური f_6 ფუნქცია წარმოადგენს f_{13} იმპლიკაციის შებრუნებულ (ინვერსირებულ), ხოლო f_7 ფუნქცია – f_{12} იმპლიკაციის შებრუნებულ (ინვერსირებულ) ფუნქციას.

● ლოგიკური f_{14} ფუნქცია (ცხრ. **4.3**) იღებს კონიუნქციის f_5 ფუნქციის მნიშვნელობათა შებრუნებულ მნიშვნელობას, ამიტომ მას **კონიუნქციის უარყოფის ფუნქციას** უწოდებენ. ლიტერატურაში იგი **და-არა** ფუნქციის ან **შეფერის ფუნქციის** სახელწოდებითაც არის მოხსენიებული. აღინიშნება შემდეგნაირად: $y = \overline{x_1 \& x_2}$, ან $y = \overline{x_1 \wedge x_2}$, ან $y = \overline{x_1 x_2}$, ან $y = x_1 | x_2$.

4.3 ცხრილში არსებული **14** ფუნქციიდან ორი ფუნქცია წარმოადგენს იმპლიკაციას და ასევე ორი – იმპლიკაციის უარყოფის ფუნქცია; მაშასადამე, სულ არსებობს **12** ელემენტალური ლოგიკური ფუნქცია.

4.3. ელემენტალურ ლოგიკურ ფუნქციათა ფუნქციონალურად სრული სისტემები

1 **4.3** ცხრილში მოყვანილ **14** ელემენტალურ ლოგიკურ ფუნქციას შორის არის იმპლიკაციის ორი და იმპლიკაციის უარყოფის ორი ფუნქცია; თითო ფუნქციით მათი შეცვლით დაგვრჩება **12** ელემენტალური ლოგიკური ფუნქცია, რომლებიც **4.4.** ცხრილშია მოყვანილი.

ლოგიკური ფუნქციების საერთო **N** რაოდენობა არგუმენტების n რაოდენობაზეა დამოკიდებული და (**4.4**) ფორმულით განისაზღვრება. არგუმენტების n რაოდენობის მცირე სიდიდით ზრდით მკვეთრად იზრდება ლოგიკური ფუნქციების საერთო რაოდენობა და, როგორც **4.1** ცხრილიდან ჩანს, n **= 5**-ის შემთხვევაში აღწევს ისეთ სიდიდეს, როგორიცაა **4294967296;** ადვილი წარმოსადგენია არგუმენტების რაოდენობის შემდგომ გაზრდის შემთხვევაში თუ რა კოლოსალურ სიდიდეს შეიძლება მიაღწიოს ფუნქციების საერთო რაოდენობამ. ასეთი რაოდენობის ლოგიკური ფუნქციების ინდივიდუალურად განხილვა პრაქტიკულად მოუხერხებელია. საბედნიეროდ აღმოჩნდა, რომ ეს საჭირო არც არის, რადგან ნებისმიერი ლოგიკური ფუნქცია ელემენტალური ლოგიკური ფუნქციების სუპერპოზციის გზით შეიძლება გამოისახოს.

ცხრ. 4.4. ელემენტალური ლოგიკური ფუნქციები

№	ელემენტალური ლოგიკური ფუნქციების	
	სახელწოდებები	**აღნიშვნები**
1	კონსტანტა 0	$y = 0$
2	კონსტანტა 1	$y = 1$
3	გამეორების ფუნქცია	$y = x$
4	ინვერსია (უარყოფის ფუნქცია)	$y = \bar{x}$
5	კონიუნქცია (და ფუნქცია)	$y = x_1 x_2$
6	დიზიუნქცია (ან ფუნქცია)	$y = x_1 + x_2$
7	კონიუნქციის უარყოფა (და-არა ფუნქცია)	$y = \overline{x_1 x_2}$
8	დიზიუნქციის უარყოფა (ან-არა ფუნქცია)	$y = \overline{x_1 + x_2}$
9	2-ის მოდულით შეკრების ფუნქცია	$y = x_1 \oplus x_2$
10	ეკვივალენტობის ფუნქცია	$y = x_1 \sim x_2$
11	იმპლიკაცია	$y = x_i \rightarrow x_j$
12	იმპლიკაციის უარყოფა	$y = \overline{x_i \rightarrow x_j}$

სუპერპოზიცია წარმოადგენს ლოგიკური ფუნქციების გარკვეული სისტემიდან (ჯგუფიდან) ლოგიკის ალგებრის ახალი სახის ფუნქციების მიღების ხერხს არგუმენტების ნუმერაციის შეცვლის და/ან არგუმენტებად სხვა ლოგიკური ფუნქციების გამოყენების საშუალებით.

მაგალითად, განვიხილოთ ლოგიკური ფუნქციების ჯგუფი: $y_1 = x_1 + x_2$; $y_2 = x_3 x_4$ და $y_3 = x_5 \rightarrow x_6$.

y_1 ფუნქციაში x_1 არგუმენტის y_2 ფუნქციით, ხოლო x_2 არგუმენტის y_3 ფუნქციით შეცვლის შედეგად (ასეთი შეცვლები დასაშვებია, რადგან როგორც ფუნქციების, ასევე არგუმენტების განსაზღვრების არეები ერთმანეთს ემთხვევა და {**0;1**} სიმრავლეს წარმოადგენს) მივიღებთ ახალ ლოგიკურ $y_4 = x_3 x_4 + (x_1 \rightarrow x_2)$ ფუნქციას**;** ასევე, ფუნქციაში არგუმენტების ნუმერაციის შეცვლით მიიღება ლოგიკის ახალი $y_5 = x_6 \rightarrow x_5$ ფუნქცია.

2 განვიხილოთ გარკვეული ელემენტალური ლოგიკური ფუნქციების გაერთიანებით წარმოქმნილი S_i სისტემა. მოცემულ S_i სისტემაში შეიძლება შედიოდეს ყველა (თორმეტივე) ელემენტალური ლოგიკური ფუნქცია, ან მხოლოდ ზოგიერთი მათგანი.

ელემენტალური ლოგიკური ფუნქციებიდან წარმოქმნილ S_i სისტემას ეწოდება **ფუნქციონალურად სრული სისტემა**, თუ იგი არ შეიცავს კონსტანტებს **0**-სა და **1**-ს და თუ შესაძლებელია ნებისმიერი ლოგიკური ფუნქცია მიღებული იქნეს ამ სისტემაში შემავალი ელემენტალური ლოგიკური ფუნქციების სუპერპოზიციის საშუალებით.

ელემენტალური ლოგიკური ფუნქციებიდან წარმოქმნილ S_i სისტემას ეწოდება **ფუნქციონალურად შესუსტებული სრული სისტემა**, თუ იგი შეიცავს კონსტანტებს **0**-სა და **1**-ს და თუ შესაძლებელია ნებისმიერი ლოგიკური ფუნქცია მიღებული იქნეს ამ სისტემაში შემავალი ელემენტალური ლოგიკური ფუნქციების სუპერპოზიციის საშუალებით.

ელემენტალური ლოგიკური ფუნქციებიდან წარმოქმნილი **მინიმალურად სრული სისტემა** ეწოდება ისეთ სრულ სისტემას, რომლისგანაც ნებისმიერი ელემენტალური ფუნქციის გამორიცხვის შედეგად მიღებული სისტემა არ არის ფუნქციონალურად სრული. მინიმალურად სრულ სისტემას **მინიმალური ბაზისი** შეესაბამება.

ელემენტალური ლოგიკური ფუნქციების ფუნქციონალურად სრულ სისტემას **ბაზისი ეწოდება.** მინიმალურ ფუნქციონალურად სრულ სისტემას **მინიმალური ბაზისი** შეესაბამება.

პოლონური წარმოშობის მქონე ამერიკელმა მათემატიკოსმა **ემილ ლეონ პოსტმა** და მისმა საბჭოთა კოლეგამ - **სერგეი ვსევოლოდის ძე იაბლონსკიმ** ერთმანეთისაგან დამოუკიდებლად ჩამოაყალიბეს ლოგიკურ ფუნქციათა სისტემების ფუნქციონალური სისრულის აუცილებელი და საკმარისი პირობების განმსაზღვრელი თეორემა, რომელიც **პოსტ-იაბლონსკის თეორემის** სახელწოდებითაა ცნობილი **და იგი ასე შეიძლება იყოს ფორმულირებული:**

ელემენტალური ლოგიკური ფუნქციების სისტემა ფუნქციონალურად სრული რომ იყოს, ამისათვის აუცილებელია და საკმარისია, რომ იგი შეიცავდეს თუნდაც ერთ ისეთ ლოგიკურ ფუნქციას, რომელიც:

- არ ინარჩუნებს **0**-ს;
- არ ინარჩუნებს **1**-ს;
- არათვითორადია;
- არაწრფივია;
- არამონოტონურია.

2 პოსტ-იაბლონსკის თეორემაში გამოყენებული ტერმინების გაცნობიერებისათვის გავეცნოთ ლოგიკური ფუნქციების ჩაკეტილ კლასებს.

ლოგიკური ფუნქციების ჩაკეტილი კლასი ეწოდება ლოგიკური ფუნქციების ისეთ სიმრავლეს, რომელიც მოიცავს აღნიშნული ფუნქციების სუპერპოზიციით მიღებულ ნებისმიერ ლოგიკურ ფუნქუნციასაც.

არსებობს ლოგიკური ფუნქციების შემდეგი ხუთი ჩაკეტილი კლასი:

- **0-ის შემნარჩუნებელი ლოგიკური ფუნქციების T_0 კლასი**, რომელიც შეიცავს ყველა ისეთ ლოგიკურ $y = f(x_1, x_2, \dots, x_n)$, ფუნქციას, რომელიც **0**-ის ტოლი ხდება მასში შემავალი ყველა არგუმენტის ნულოვანი მნიშვნელობის დროს, ე.ი. რომლისთვისაც სამართლიანია გამოსახულება $y = f(0, 0, \dots, 0) = 0$. **0**-ის შემნარჩუნებელი ლოგიკური ფუნქციებია, მაგალითად, კონსტანტა **0**, გამეორების ფუნქცია, კონიუნქცია, დიზიუნქცია, **2**-ის მოდულით შეკრების ფუნქცია და ა.შ. **0**-ის შემნარჩუნებელი ლოგიკური ფუნქციების რაოდენობა უდრის არსებული ლოგიკური ფუნქციების ნახევარს.

ე. ლ. პოსტი
(1897-1954)

ს. ვ. იაბლონსკი
(1924-1998)

● **1-ის შემნარჩუნებელი ლოგიკური ფუნქციების T_1 კლასი**, რომელიც შეიცავს ყველა ისეთ ლოგიკურ $f(x_1, x_2, \dots, x_n)$, ფუნქციას, რომელიც 1-ის ტოლი ხდება მასში შემავალი ყველა არგუმენტის 1-ის ტოლი მნიშვნელობის დროს, ე.ი. რომლისთვისაც სამართლიანია გამოსახულება $f(1,1,\dots,1) = 1$. 1-ის შემნარჩუნებელი ლოგიკური ფუნქციებია, მაგალითად, კონსტანტა 1, ინვერსია, ეკვივალენტობის ფუნქცია და ა.შ. 0-ის შემნარჩუნებელი ლოგიკური ფუნქციების რაოდენობა უდრის არსებული ლოგიკური ფუნქციების ნახევარს.

● ***თვითორადი ლოგიკური ფუნქციების* S *კლასი***. ლოგიკური $f_1(x_1, x_2, \dots, x_n)$ და $f_2(x_1, x_2, \dots, x_n)$ ფუნქციებს ერთმანეთის მიმართ **ორადი** ფუნქციები ეწოდება, თუ სრულდება ტოლობა: $f_1(x_1, x_2, \dots, x_n) = \overline{f_2(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)}$. საკუთარი თავისადმი ორად ფუნქციას ***თვითორადი ფუნქცია*** ეწოდება. თვითორადი $f(x_1, x_2, \dots, x_n)$ ფუნქციისათვის მართებულია ტოლობა: $f(x_1, x_2, \dots, x_n) = \overline{f(\overline{x}_1, \overline{x}_2, \dots, \overline{x}_n)}$.

ცხრ. 4.5. თვითორადი ფუნქციის მაგალითი

x_1	x_2	x_3	$x_1 x_2$	$x_1 x_3$	$x_2 x_3$	y
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	1	1
1	0	0	0	0	0	0
1	0	1	0	0	0	1
1	1	0	1	0	0	1
1	1	1	1	1	1	1

მოყვანილი განსაზღვრის თანახმად ფუნქცია თვითორადია, თუ არგუმენტების ურთიერთსაწინააღმდეგო მნიშვნელობებზე თვითონაც ურთიერთსაწინააღმდეგო მნიშვნელობებს იღებს. განვიხილოთ ლოგიკური ფუნქცია:

$$y = x_1x_2 + x_1x_3 + x_2x_3\ .$$

მისი ჭეშმარიტობის 4.5 ცხრილიდან ჩანს, რომ იგი არგუმენტების ურთიერთსაწინააღმდეგო (000 და 111), (001 და 110), (010 და 101), (011 და და 100) მნიშვნელობების დროს ურთიერთსაწინააღმდეგო მნიშვნელობებს იღებს; ამიტომ განხილული ფუნქცია თვითორადია. n რაოდენობის არგუმენტებზე დამოკიდებული 2^{2^n} რაოდენობის ლოგიკური ფუნქციებიდან თვითორადი ფუნქციების რაოდენობა $\sqrt{2^{2^n}}$ -ის ტოლია. თვითორადი ლოგიკური ფუნქციების ერთობლიობა წარმოქმნის თვითორადი ლოგიკური ფუნქციების ჩაკეტილ კლასს.

● **წრფივი ლოგიკური ფუნქციების L კლასში** გაერთიანებულია ისეთი ლოგიკური $f(x_1, x_2, \dots, x_n)$ ფუნქციები, რომლებიც შეიძლება შემდეგი სახით იქნეს წარმოდგენილი:

$$f\,(x_1, x_2, \ldots, x_n) \;=\; c_0 \oplus\; c_1x_1 \oplus c_2x_2 \oplus \ldots \oplus\; c_nx_n\;, \qquad (4.1)$$

სადაც $c_i \in \{0;1\}$, $i = 0,1,2, \ldots, n$.

მოცემულ კლასში შედის, მაგალითად, შემდეგი ფუნქციები: კონსტანტა **0**, კონსტანტა **1**, გამეორების ფუნქცია, ინვერსია $\overline{x} = 1 \oplus x$, **2**-ის მოდულით შეკრების $x_1 \oplus x_2$ ფუნქცია, ეკვივალენტობის $x_1 \sim x_2 = x_1 \oplus x_2 \oplus 1$ ფუნქცია.

● **მონოტონური ლოგიკური ფუნქციების M კლასი.** ლოგიკური ფუნქციების მონოტონურობის განმარტებისათვის საჭიროა წინასწარ გავეცნოთ ორობითი ნაკრებების ურთიერთშედარების წესს. განვიხილოთ შემდეგი ორი ორობითი ნაკრები:

$$\widehat{X_1} = \langle x_1^1 x_2^1 \ldots x_n^1 \rangle \quad \text{და} \quad \widehat{X_2} = \langle x_1^2 x_2^2 \ldots x_n^2 \rangle\;,$$

სადაც x_i^1, $x_i^2 \in \{0;1\}$ $(i = 1,2, \ldots, n)$ და მათ ეწოდებათ შესაბამისად ორობითი $\widehat{X_1}$ და $\widehat{X_2}$ ნაკრებების კომპონენტები;

ამბობენ, რომ ორობითი $\widehat{X_1}$ ნაკრები არ არის ორობით $\widehat{X_2}$ ნაკრებზე მეტი, თუ $\widehat{X_1}$ და $\widehat{X_2}$ ნაკრებთა თითოეული i–ური კომპონენტისათვის სრულდება თანაფარდობა:

$$x_i^1 \;\le\; x_i^2; \qquad (4.2)$$

ამ დროს ითვლება, რომ $0 \le 0$, $1 \le 1$ და $0 \le 1$.

ორობითი $\widehat{X_1}$ ნაკრები თუ არ არის მეტი $\widehat{X_2}$ ნაკრებზე, მაშინ წერენ, რომ $\widehat{X_1} \le \widehat{X_2}$. მაგალითად, $1001 \le 1011$.

ამბობენ, რომ ორობითი $\widehat{X_1}$ და $\widehat{X_2}$ ნაკრებების შედარება არ შეიძლება, თუ ამ ნაკრებთა ყველა კომპონენტისათვის არ სრულდება შემდეგი უტოლობები:

$$x_i^1 \;\le\; x_i^2 \;\; \text{ან} \;\; x_i^1 \;\ge\; x_i^2$$

მაგალითად, ორობითი **1010** და **0111** ნაკრებების**.** შედარება არ შეიძლება.

ლოგიკურ $f\,(x_1, x_2, \ldots, x_n)$ ფუნქციას ეწოდება **მონოტონური ფუნქცია,** თუ ნებისმიერი ორი ორობითი $\widehat{X_1} = \langle x_1^1 x_2^1 \ldots x_n^1 \rangle$ და $\widehat{X_2} = \langle x_1^2 x_2^2 \ldots x_n^2 \rangle$ ნაკრებისათვის, რომლებისთვისაც სრულდება $\widehat{X_1} \le \widehat{X_2}$ პირობა, მართებულია უტოლობა:

$$f\,(x_1^1 x_2^1 \ldots x_n^1) \;\le\; f\,(x_1^2 x_2^2 \ldots x_n^2)\,. \qquad (4.3)$$

არამონოტონურია ინვერსია და ინვერსიის შემცველი ნებისმიერი ლოგიკური ფუნქცია.

ლოგიკური ფუნქციების ზემოთ განხილული ჩაკეტილი კლასების გამოყენების მეშვეობით **პოსტ-იაბლონსკის თეორემის** ინტერპრეტირება შეიძლება ასე მოვახდინოთ:

ელემენტალური ლოგიკური ფუნქციების სისტემა ფუნქციონალურად სრული რომ იყოს, ამისათვის აუცილებელი და საკმარისია, რომ იგი შეიცავდეს თუნდაც ერთ ისეთ ლოგიკურ ფუნქციას, რომელიც:

- არ შედის ჩაკეტილ $\mathbf{T_0}$ კლასში;
- არ შედის ჩაკეტილ $\mathbf{T_1}$ კლასში;
- არ შედის ჩაკეტილ **S** კლასში;
- არ შედის ჩაკეტილ **L** კლასში;
- არ შედის ჩაკეტილ **M** კლასში.

ელემენტალური ლოგიკური ფუნქციებისგან შედგენილი კონკრეტული სისტემის ფუნქციონალური სისრულის დასადგენად მოსახერხებელია ვისარგებლოთ **4.6** ცხრილით, რომელშიც ვარსკვლავებით აღნიშნულია მოცემული ელემენტალური ლოგიკური ფუნქციისათვის დამახასიათებელი თვისებები. ამ ცხრილიდან ჩანს, რომ ფუნქციონალურად სრული სისტემებია:

● **დიზიუნქციის**, **კონიუნქციისა** და **ინვერსიისაგან** შემდგარი სისტემა. რადგან $\mathbf{S_1}$ სიმბოლოთი აღნიშნული გვაქვს ყველა ელემენტალური ლოგიკური ფუნქციისაგან შედგე-

ნილი ფუნქციონალურად სრული სისტემა, ამიტომ მოცემული სისტემა აღვნიშნოთ **S_2** სიმბოლოთი;

- **დიზიუნქციისა** და **ინვერსიისაგან** შემდგარი სისტემა; იგი **S_3** სიმბოლოთი აღვნიშნოთ;
- **კონიუნქციისა** და **ინვერსიისაგან** შემდგარი სისტემა; იგი **S_4** სიმბოლოთი აღვნიშნოთ;
- ერთადერთი **ᲓᲐ-ᲐᲠᲐ ფუნქციისაგან** შემდგარი სისტემა; იგი **S_5** სიმბოლოთი აღვნიშნოთ;
- ერთადერთი **ᲐᲜ-ᲐᲠᲐ ფუნქციისაგან** შემდგარი სისტემა; იგი **S_6** სიმბოლოთი აღვნიშნოთ;

ცხრ. 4.6. ელემენტალური ლოგიკური ფუნქციებისაგან შედგენილი სისტემების ფუნქციონალური სისრულის დასადგენი დამხმარე ცხრილი

ელემენტალური ლოგიკური ფუნქციის თვისება	ელემენტალური ლოგიკური ფუნქციები									
	კონსტანტა 0	კონსტანტა 1	ინვერსია	კონიუნქცია	დიზიუნქცია	2-ის მოდულით შეკრება	ეკვივალენტობა	იმპლიკაცია	ᲓᲐ-ᲐᲠᲐ ფუნქცია	ᲐᲜ-ᲐᲠᲐ ფუნქცია
0-ს არ ინარჩუნებს		*	*				*	*	*	*
1-ს არ ინარჩუნებს	*		*			*			*	*
არათვითორადია	*	*		*	*	*	*	*	*	*
არაწრფივია				*	*			*	*	*
არამონოტონურია			*			*	*	*	*	*

განხილული ფუნქციონალურად სრული სისტემიდან **S_2** და **S_3** სისტემები არ არის მინიმალური, ვინაიდან **S_2** სისტემიდან კონიუნქციის გამორიცხვით მიღებული **S_3** და დიზიუნქციის გამორიცხვით მიღებული **S_4** სისტემა მაინც ფუნქციონალურად სრულ სისტემებს წარმოადგენს.

რაც შეეხება **S_3, S_4, S_5** და **S_6** სისტემებს, ისინი მინიმალურ სისტემებს წარმოადგენს, რადგან მათგან ნებისმიერი ელემენტალური ფუნქციის გამორიცხვით მიიღება ფუნქცონალურად არასრული, ან ცარიელი სისტემა.

პოსტ-იაბლონსკის თეორემის თანახმა უსასრულო რაოდენობის მქონე ლოგიკური ფუნქციებიდან ნებისმიერი ლოგიკური ფუნქცია შეიძლება გამოვსახოთ:

- დიზიუნქციის, კონიუნქციისა და ინვერსიის საშუალებით;
- დიზიუნქციისა და ინვერსიის საშუალებით;
- კონიუნქციისა და ინვერსიის საშუალებით;
- **ᲓᲐ-ᲐᲠᲐ** ფუნქციის საშუალებით;
- **ᲐᲜ-ᲐᲠᲐ** ფუნქციის საშუალებით;

4.4. ლოგიკის (ბულის) ალგებრა და მისი კანონები

1

ალგებრა წარმოადგენს მეცნიერებას, რომელიც **იზომორფიზმამდე** სიზუს-ტით შეისწავლის **ალგებრულ სისტემებს**.

ზემოთ მოყვანილ განსაზღვრებაში გამოყენებული ტერმინი **იზომორფიზმი** (ბერ-ძნული სიტყვებისაგან ***isos*** – "ერთნაირი", ***homoios*** – "მსგავსი" და ***morphe*** – "ფორმა") სიტყვა-სიტყვით ითარგმნება როგორც **"ერთნაირი, მსგავსი ფორმები"** და იგი ობიექტების სტრუქტურებს შორის არსებულ შესაბამისობას ახასიათებს.

ნებისმიერი ბუნების ელემენტებისაგან შედგენილი ორი სისტემა ერთმანეთის **იზო-მორფულია**, თუ პირველი სისტემის შემადგენლობაში არსებულ თითოეულ ელემენტსა და კავშირს მეორე სისტემაში არსებული მხოლოდ ერთი ელემენტი და ერთი კავშირი შე-ესაბამება და პირიქით.

იზომორფიზმამდე სიზუსტით შესწავლა ნიშნავს მოცემული სისტემის ყველა იზო-მორფული (ე.ი. იგივური, მსგავსი ფორმის მქონე) სისტემის შესწავლას.

რაც შეეხება **ალგებრულ სისტემებს,** იგი წარმოადგენს სიმრავლეთა მოწესრიგებულ **{R; E}** წყვილს, რომელთაგანაც პირველ **(R)** სიმრავლეში გაერთიანებულია გარკვეული ბუნების ელემენტები (რიცხვები, ცნებები, ასოები), ხოლო მეორე **(E)** სიმრავლეში – ოპერაციები (შეკრება, გამოკლება, ამოფესვა და ა.შ.).

ჯორჯ ბულის მიერ შექმნილი **ლოგიკის** (ან, როგორც იგი უწოდებდა, **გამოთქმების**) **ალგებრა** წარმოადგენს ისეთ ალგებრულ სისტემას, რომლისთვისაც **R** სიმრავლეში გა-ერთიანებულია ლათინური ასოებით აღნიშნული **გამოთქმები**, ხოლო **E** სიმრავლეში – ორი ბინარული (ლოგიკური შეკრება, ანუ **დიზიუნქცია** და ლოგიკური გამრავლება, ანუ **კონიუნქცია**) და ერთი უნარული (**ინვერსია**) ოპერაცია.

R სიმრავლეში შემავალ ლათინურ ასოებს **ლოგიკური ცვლადები** ეწოდებათ; ისინი იღებენ **0**-ის ტოლ მნიშვნელობებს, თუ მათი საშუალებით აღნიშნული გამოთქმები ყალ-ბია და **1**-ის ტოლ მნიშვნელობას – თუ ისინი ჭეშმარიტია.

ვინაიდან ლოგიკური ცვლადების მნიშვნელობები გამოხატავს არა გარკვეულ სიდი-დეებს, არამედ ჭეშმარიტობა-სიყალბის ფაქტს, მათ **ლოგიკური მნიშვნელობები** ეწოდებათ.

ლოგიკურ (გამოთქმების) ალგებრას ხშირად მისი შემქნელის პატივსაცემად, **ბულის ალგებრასაც**) უწოდებენ.

სასკოლო ალგებრაში გამოყენებული ცვლადები იღებს ნებისმიერ (მთელ, ნამდვილ, რაციონალურ, არარაციონალურ) მნიშვნელობებს, ხოლო ლოგიკური ცვლადები მხოლოდ ორ (**0**-ის ან **1**-ის ტოლ) მნიშვნელობებს, ამიტომ მათ **ორობით ცვლადებსაც** უწოდებენ.

2

სასკოლო ალგებრის მსგავსად **ლოგიკის ალგებრაშიც** სამართლიანია გარკ-კვეული კანონები. ზოგიერთ შემთხვევაში სასკოლო და ლოგიკის ალგებრათა კანონები ერთმანეთს ემთხვევა, მაგრამ ლოგიკის ალგებრაში არსებობს ისეთი კა-ნონებიც, რომლებიც სასკოლო ალგებრაში არ სრულდება. განვიხილოთ ლოგიკის ალგებრის **ძირითადი კანონები.**

1) **ასოციურობის** (ლათ. associatio – "შეერთება") **კანონი.** სამართლიანია როგორც ლოგიკის, ასევე სასკოლო ალგებრისთვისაც და ცვლადების სხვადასხვაგვარად დაჯგუფე-ბის საშუალებას იძლევა:

$$x_1(x_2x_3) = (x_1x_2)\,x_3; \qquad (4.4)$$

$$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3. \qquad (4.5)$$

2) **კომუტატურობის** (ლათ. commutatio – "ცვლა") **კანონი.** სამართლიანია როგორც ლოგიკის, ასევე სასკოლო ალგებრისთვისაც და ოპერანდების გადანაცვლების საშუალებას იძლევა:

$$x_1 x_2 = x_2 x_1; \qquad (4.6)$$

$$x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3. \qquad (4.7)$$

3) **დიზიუნქციის მიმართ** (ლათ. distribution – "განაწილება", "განთავსება") **კონიუნქციის დისტრიბუტულობის კანონი.** სამართლიანია როგორც ლოგიკის, ასევე სასკოლო ალგებრისთვისაც და ფრჩხილის გახსნის საშუალებას იძლევა:

$$x_1(x_2 + x_3) = x_1x_2 + x_1x_3; \qquad (4.8)$$

4) **კონიუნქციის მიმართ დიზიუნქციის დისტრიბუტულობის კანონი.** სამართლიანია მხოლოდ ლოგიკის ალგებრისთვის, ხოლო სასკოლო ალგებრაში იგი არ სრულდება:

$$x_1 + x_2 x_3 = (x_1 + x_2)(x_1 + x_3); \qquad (4.9)$$

ქვემოთ მოყვანილი კანოონები მხოლოდ ლოგიკის ალგებრისთვის არის სამართლიანი.

5) **იდემპონტენტურობის (ტავტოლოგიის) კანონი:**

$$x \bullet x = x; \qquad (4.10)$$

$$x + x = x; \qquad (4.11)$$

6) **ორმაგი უარყოფის კანონი:**

$$\overline{\overline{X}} = x; \qquad (4.12)$$

7) **დე მორგანის კანონი:**

$$\overline{x_1 x_2} = \overline{x_1} + \overline{x_2}; \qquad (4.13)$$

$$\overline{x_1 + x_2} = \overline{x_1} \bullet \overline{x_2}; \qquad (4.14)$$

8) **წინააღმდეგობის კანონი:**

$$x \bullet \overline{x} = 0; \qquad (4.15)$$

9) **გამორიცხული მესამის კანონი:**

$$x + \overline{x} = 1. \qquad (4.16)$$

10) **შეწებების კანონი:**

$$x_1 \bullet x_2 + x_1 \overline{x_2} = x_1 \qquad (4.17)$$

$$(x_1 + x_2)(x_1 + \overline{x_2}) = x_1 \qquad (4.18)$$

11) **შთანთქმის (აბსორბციის) კანონი:**

$$x_1 + x_1 x_2 = x_1 \qquad (4.19)$$

$$x_1 (x_1 + x_2) = x_1 \qquad (4.20)$$

12) **უნივერსალური სიმრავლის კანონი:**

$$x + 1 = 1; \qquad (4.21)$$

$$x \bullet 1 = x; \qquad (4.22)$$

13) **ნულოვანი სიმრავლის კანონი:**

$$x + 0 = x; \qquad (4.23)$$

$$x \bullet 0 = x; \qquad (4.24)$$

დე მორგანის კანონი **კლოდ შენონმა** განაზოგადა ნებისმიერი რაოდენობის ლოგიკური (ორობითი) ცვლადებისათვის: დიზიუნქციის და კონიუნქციის ნიშნებით დაკავშირებული კომბინაციის ინვერტირებით მიიღება კომბინაციები, რომლებშიც:

- დიზიუნქციის ოპერაციები შეცვლილია კონიუნქციის ოპერაციებით;

$$\overline{x_1 + x_2 + x_3 + \ldots + x_n} = \overline{x_1} \bullet \overline{x_2} \bullet \overline{x_3} \bullet \ldots \bullet \overline{x_n}\,. \qquad (4.25)$$

- კონიუნქციის ოპერაციები შეცვლილია დიზიუნქციის ოპერაციებით:

$$\overline{x_1\, x_2\, x_3 \ldots x_n} = \overline{x_1} + \overline{x_2} + \overline{x_3} + \ldots + \overline{x_n}\,; \qquad (4.26)$$

3

დიზიუნქციისა და/ან კონიუნქციის ნიშნებით შეერთებული უინვერსიო და/ან ინვერსიულ ლოგიკური ცვლადების ერთობლიობა წარმოქმნის **ლოგიკურ ფორმულებს.** ლოგიკის ალგებრის ზემოთ განხილული კანონების გამოყენებით შესაძლებელია გავამარტივოთ ლოგიკური ფორმულები.

განვიხილოთ ლოგიკის ძირითადი კანონების გამოყენებით ლოგიკური ფორმულების გამარტივების რამდენიმე მაგალითი.

მაგალითი 4.1.

$$\overline{x + y}\,(x\,\overline{y}) = \overline{x}\,\overline{y}\,(x\,\overline{y}) =$$
$$= \overline{x}\,x\,\overline{y}\,\overline{y} = 0\;\overline{y}\,\overline{y} = 0\,\overline{y} = 0.$$

მიმდევრობით გამოყენებულია: დე მორგანის, ასოციაციურობის, გამორიცხული მესამისა და ნულოვანი სიმრავლის კანონები.

მაგალითი 4.2.

$$\overline{x}\,y + \overline{x + y} + x = \overline{x}\,y + \overline{x}\,\overline{y} +$$
$$+ x = \overline{x}\,(y + \overline{y}) + x =$$
$$= \overline{x} + x = 1.$$

პირველად გამოყენებულია დე მორგანის კანონი; შემდეგ ფრჩხილებს გარეთაა გატანილი $\overline{x}$ ცვლადი და, ბოლოს, გამოყენებულია გამორიცხული მესამის კანონი.

მაგალითი 4.3.

$$x\,y + x\,y\,z + x\,z\,p =$$
$$= x\,(y\,(1 + z) + z\,p) =$$
$$= x\,(y + z\,p).$$

პირველად ფრჩხილებს გარეთ გატანილია საერთო მამრავლები, ხოლო შემდეგ გამოყენებულია უნივერსალური სიმრავლის კანონი.

მაგალითი 4.4.

დავამტკიცოთ კონიუნქციის მიმართ დიზიუნქციის დისტრიბუტულობის კანონი:

$$\boxed{(x_1+x_2)\ (x_1+x_3)} = x_1x_1+x_1x_3+x_2x_1+x_2x_3 = x_1+x_1x_3+x_2x_1+x_2x_3 = x_1(1+x_3)+x_2x_1+$$
$$+x_2x_3 = x_1+x_2x_1+x_2x_3 = x_1(1+x_2)+x_2x_3 = \boxed{x_1+x_2x_3}$$

გამოყენებულია იდემპოტენტურობის (ტავტოლოგიის) და უნივერსალური სიმრავლ-ის კანონები.

4.5. ლოგიკური ფუნქციების ანალიზური ფორმები

1 ლოგიკის ალგებრაში განსახილველი ფუნქციები წარმოადგენს ჩვენ მიერ გან-საზღვრულ ლოგიკურ ფუნქციებს. ჩვენ უკვე გავეცანით მათი წარმოდგენის ცხრილურ ფორმას. მოცემული პარაგრაფის მიზანია შევისწავლოთ ლოგიკური ფუნქციების ანალიზური და კოორდინატული ფორმები, რომლებიც მნიშვნელოვან როლს თამაშობენ აღნიშნული ფუნქციების გარდაქმნისა და გამარტივების პროცესებში.

თვალსაჩინოებისათვის შესწავლას დავიწყებთ სამ ცვლადზე დამოკიდებული ლოგიკუ-რი $\boldsymbol{y=f(x_1,x_2,x_3)}$ ფუნქციის განხილვით; მიღებული შედეგები შეიძლება განზოგადებული იყოს ნებისმიერი $\boldsymbol{n}$ რაოდენობის არგუმენტზე დამოკიდებული ნებისმიერი $\boldsymbol{y=f(x_1,x_2,\ldots,x_n)}$ ფუნქციისათვისაც.

2 გამორიცხული მესამის კანონის გამომხატველი (**4.16**) ფორმულის ძალით შეგვიძლია დავწეროთ:

$$1 = (x_1+\overline{x_1})\ (x_2+\overline{x_2})\ldots(x_n+\overline{x_n}). \qquad (4.27)$$

ფრჩხილების გახსნისა და წევრების დაჯგუფების შემდეგ მივიღებთ:

$$1 = x_1\,x_2\ldots x_{n-1}\,x_n + x_1\,x_2\ldots x_{n-1}\,\overline{x_n} + x_1\,x_2\ldots\overline{x_{n-1}}\,x_n +$$
$$= + x_1\,x_2\ldots\overline{x_{n-1}}\,\overline{x_n} + \ldots + \overline{x_1}\,\overline{x_2}\ldots\overline{x_{n-1}}\,\overline{x_n} \qquad (4.28)$$

(**4.28**) გამოსახულების თანახმად **კონსტანტა** 1 შეიძლება დაიშალოს 2^n რაოდენობის კონიუნქციების დიზიუნქციად; თითოეულ ასეთ კონუნქციას **ერთიანის კონსტიტუენტა** ანუ **მინიტერმი** ეწოდება. აღსანიშნავია, რომ:

- (**4.28**) გამოსახულებაში შემავალ ნებისმიერ მინიტერმში თუ უინვერსიო ცვლადებს შევცვლით ერთიანებით, ხოლო ინვერსიულ ცვლადებს – ნულიანებით, მივიღებთ არგუმე-ნტების მნიშვნელობათა იმ ნაკრებს, რომელსაც მოცემული მინიტერმი შეესაბამება;
- ნებისმიერი ორი მინიტერმი, უკიდურეს შემთხვევაში, ერთი ცვლადის მნიშვნელო-ბით მაინც განსხვავდება ერთმანეთისაგან და ამიტომ მათი ლოგიკური ნამრავლი (კონიუნ-ქცია) ნულის ტოლია.

შემოვიღოთ:

- აღნიშვნები $\overline{x} = x^0,\ x = x^1;$
- ორობითი ცვლადი $\boldsymbol{\sigma \in \{0;1\}}$

და (**4.28**) გადავწეროთ შემოკლებული ფორმით:

$$1 = \bigvee_{(\sigma_1\sigma_2\ldots\sigma_n)} x_1^{\sigma_1}\,x_2^{\sigma_2}\ldots x_n^{\sigma_n} \qquad (4.29)$$

სადაც $\bigvee$ დიზიუნქციის ნიშანია და დიზიუნქცია აიღება $\sigma_1, \sigma_2, \dots, \sigma_n$ ცვლადების მნიშვნელობათა ყველა ნაკრებისათვის.

იმისათვის, რომ სრულდებოდეს $x^{\sigma} = 1$ ტოლობა, აუცილებელია x–სა და σ–ს ჰქონდეთ ერთნაირი მნიშვნელობა (იხ. ცხრილი 4.7). მაშასადამე , $x_1^{\sigma_1}\, x_2^{\sigma_2} \dots x_n^{\sigma_n}$ მინიტერმი 1-ის ტოლი ხდება x_i არგუმენტების მნიშვნელობის ერთადერთი ნაკრებისათვის, კერძოდ მაშინ, როდესაც $x_1 = \sigma_1,\ x_2 = \sigma_2,\ \dots,\ x_n = \sigma_n$.

(2.29) გამოსახულებაში შემავალ მინიტერმს, რომელიც შეესაბამება $\sigma_1, \sigma_2, \dots, \sigma_n$ ცვლადების i-ურ ($i = 0, 1, \dots, (2^n - 1)$) ნაკრებს, **$i$-ური მინიტერმი** ვუწოდოთ.

ცხრ. 4.7. x^{σ} სიდიდის მნიშვნელობის განსაზღვრა

x	σ	x^{σ}
0	0	$0^0 = \overline{0} = 1$
0	1	$0^1 = 1$
1	0	$1^0 = \overline{1} = 0$
1	1	$1^1 = 1$

ზემოთ აღნიშნულ i-ურ მინიტერმში შემავალ უინვერსიო ცვლადებს თუ შევცვლით 1-ანებით, ხოლო ინვერსირებულ ცვლადებს – 0-იანებით, მაშინ მივიღებთ $y=f(x_1,x_2,\dots,x_n)$ ფუნქციის **არგუმენტების მნიშვნელობათა i-ურ ნაკრებს.** მაშასადამე:

> არსებობს ურთიერთცალსახა დამოკიდებულება (4.29) გამოსახულებაში შემავალ მინიტერმებსა და ლოგიკის ალგებრის n რაოდენობის ცვლადზე დამოკიდებული ფუნქციის არგუმენტების მნიშნელობათა ნაკრებებს შორის.

$f(x_1,x_2,\dots,x_n)$ თუ ფუნქცია i-ური მინიტერმის შესაბამისი არგუმენტების მნიშვნელობათა i-ურ ნაკრებზე 0-ის ტოლი ხდება, მაშინ ამ მინიტერმს **ნულოვანი მინიტერმი** ვუწოდოთ

ანალოგიურად, i-ურ მინიტერმს ვუწოდოთ **ერთეულოვანი მინიტერმი,** თუ $f(x_1,x_2,\dots,x_n)$ ფუნქცია 1-ის ტოლი ხდება ამ მინიტერმის შესაბამისი არგუმენტების მნიშნელობათა i-ურ ნაკრებზე.

იგივურად ნულის ტოლი ლოგიკური ფუნქციის მნიშვნელობა 0-ის ტოლია არგუმენტების მნიშვნელობათა ნებისმიერი ნაკრებისათვის**;** ასეთ შემთხვევაში (4.29) გამოსახულებაში შემავალი ყველა მინიტერმი ნულოვანი იქნება; ასევე, იგივურად ერთის ტოლი ლოგიკური ფუნქციის მნიშვნელობა 1-ის ტოლია არგუმენტების მნიშვნელობათა ნებისმიერი ნაკრებისათვის; ასეთ შემთხვევაში (4.29) გამოსახულებაში შემავალი ყველა მინიტერმი ერთეულოვანი იქნება.

იგივურად 0-ის და 1-ის არატოლ ნებისმიერ ორ $f_i(x_1,x_2,\dots,x_n)$ და $f_j(x_1,x_2,\dots,x_n)$ ფუნქციას ერთეულოვანი მინიტერმების სხვადასხვა ერთობლიობები შეესაბამება. აღნიშნული მინიტერმების დიზიუნქციის ნიშნით დაკავშირებით მიღებულ გამოსახულებას ლოგიკური ფუნქციის **დიზიუნქციური სრულყოფილი ნორმალური ფორმა** ეწოდება და იგი ლოგიკური ფუნქციის წარმოდგენის ანალიზურ ფორმად გამოიყენება.

მაგალითად, განვიხილოთ 3 არგუმენტზე დამოკიდებული $y=f(x_1,x_2,x_3)$ ფუნქციები. აღნიშნული ფუნქციებისათვის (4.29) გამოსახულებას ექნება სახე:

$$1 = \bigvee_{(\sigma_1\,\sigma_2\,\sigma_3)} x_1^{\sigma_1}\, x_2^{\sigma_2}\, x_3^{\sigma_3} \qquad (4.30)$$

სამ არგუმენტზე დამოკიდებული ლოგიკური ფუნქციების საერთო რაოდენობა $2^{2^3} = 256$-ის ტოლია და ზოგიერთი მათგანი 4.7 ცხრილშია მოყვანილი. იგივურად ნულის ტოლია $y_1 = f_1(x_1,x_2,x_3)$ ფუნქცია. იგი ნულის ტოლ მნიშვნელობას იღებს არგუმენტების მნიშვნელობათა ნებისმიერ ნაკრებზე. იგივურად ერთის ტოლია $y_{256} = f_{256}(x_1,x_2,x_3)$

ფუნქცია, რომელიც 1-ის ტოლ მნიშვნელობას იღებს არგუმენტების მნიშვნელობათა ნებისმიერ ნაკრებზე.

ცხრ. 4.7. სამ ცვლადზე დამოკიდებული ლოგიკური $y=f(x_1,x_2,x_3)$ ფუნქციები

x_1	x_2	x_3	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	•••	y_{255}	y_{255}	y_{256}
0	0	0	0	0	0	0	0	0	0	0	0	0	0	•••	1	1	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	•••	1	1	1
0	1	0	0	0	0	0	0	0	0	0	0	0	0	•••	1	1	1
0	1	1	0	0	0	0	0	0	0	0	0	0	0	•••	1	1	1
1	0	0	0	0	0	0	0	0	0	0	1	1	1	•••	1	1	1
1	0	1	0	0	0	0	1	1	1	1	0	0	0	•••	1	1	1
1	1	0	0	0	1	1	0	0	1	1	0	0	1	•••	0	1	1
1	1	1	0	1	0	1	0	1	0	1	0	1	0	•••	1	0	1

დანარჩენი ფუნქციებიდან ნებისმიერ ორ $f_i(x_1,x_2,\dots,x_n)$, $f_j(x_1,x_2,\dots,x_n)$ ფუნქციას, სადაც $i, j \in (2,3,\dots,255)$, ერთეულოვანი მინიტერმების სხვადასხვა ერთობლიობები შეესაბამება; კერძოდ:

$y_1 = f_1(x_1,x_2,x_3)$ ფუნქციას შეესაბამება ერთეულოვანი მინიტერმების $\{x_1\ x_2\ x_2\}$ ნაკრები;

$y_4 = f_1(x_1,x_2,x_3)$ ფუნქციას შეესაბამება ერთეულოვანი მინიტერმების $\{x_1\ x_2\ x_2;\ \ x_1\ x_2\ \overline{x_2}\}$ ნაკრებები;

$y_8 = f_1(x_1,x_2,x_3)$ ფუნქციას შეესაბამება ერთეულოვანი მინიტერმების $\{x_1\ x_2\ x_2;\ \ x_1\ x_2\ \overline{x_2};\ x_1\ \overline{x_2}\ x_3\}$ ნაკრებები და ა.შ.

$f(x_1,x_2,\dots,x_n)$ ფუნქციის არგუმენტების მნიშვნელობათა ნაკრებს, რომელიც $x_1^{\sigma_1}\, x_1^{\sigma_1} \bullet\bullet\bullet\, x_n^{\sigma_n}$ მინიტერმს შეესაბამება, აქვს შემდეგი სახე: $\sigma_1\ \sigma_1\ \dots\ \sigma_1$.

(4.29) გამოსახულებაში შემავალი თითოეული $x_1^{\sigma_1}\, x_1^{\sigma_1} \bullet\bullet\bullet\, x_n^{\sigma_n}$ მინიტერმი გავამრავლოთ ფუნქციის $f\,(\sigma_1\ \sigma_1\ \dots\ \sigma_1)$ მნიშვნელობაზე. მიღებულ გამოსახულებაში 0-ის ტოლი გახდება ყველა ის დიზიუნქციური წევრი (მინიტერმი), რომლისთვისაც სრულდება ტოლობა: $f\,(\sigma_1\sigma_2\dots\sigma_n)=0$. ვინაიდან $x + 0 = x$, ამიტომ (4.29) გამოსახულებაში დაგვრჩება მხოლოდ ერთეულოვანი მინიტერმები, რომელთა დიზიუნქციაც $f(x_1,x_2,\dots,x_n)$ ფუნქციას გამოსახავს, ე.ი.

$$f(x_1,x_2,\dots,x_n) = \bigvee_{(\sigma_1\,\sigma_2\,\dots\,\sigma_n)} x_1^{\sigma_1}\, x_2^{\sigma_2} \dots x_n^{\sigma_n} \bullet f\,(\sigma_1\sigma_1\dots\sigma_1) \qquad (4.31)$$

(4.31) გამოსახულება შეიცავს მხოლოდ ისეთ $x_1^{\sigma_1}\, x_1^{\sigma_1} \bullet\bullet\bullet\, x_n^{\sigma_n}$ მინიტერმებს, რომლებისთვისაც მართებულია ტოლობა $f\,(\sigma_1\sigma_2\dots\sigma_n)=1$, ამიტომ იგი შეიძლება ასე გადავწეროთ:

$$f(x_1,x_2,\dots,x_n) = \bigvee_{f(\sigma_1\,\sigma_2\,\dots\,\sigma_n)=1} x_1^{\sigma_1}\, x_2^{\sigma_2} \dots x_n^{\sigma_n} \qquad (4.32)$$

(4.32) ფორმით ლოგიკური ფუნქციის წარმოდგენას **დიზიუნქციური სრულყოფილი ნორმალური ფორმა**, ანუ შემოკლებით – **დსნფ** ეწოდება.

მოცემულ ტერმინში სიტყვა ‘‘სრულყოფილი’’ გამოხატავს იმ ფაქტს, რომ ფუნქციის გამოსახულებაში შემავალი თითოეული მინიტერმი (დიზიუნქციური წევრი) შეიცავს ყველა იმ არგუმენტს, რომელზედაც დამოკიდებულია მოცემული ფუნქცია.

4

დიზიუნქციური სრულყოფილი ნორმალური ფორმა წარმოადგენს ლოგიკური ფუნქციის გამოსახვის **ანალიზურ** ფორმას. იგი შესაძლებელია მივიღოთ ლოგიკური ფუნქციის ჭეშმარიტობის ცხრილიდან შემდეგი ალგორითმის გამოყენებით (მოცემულ ალგორითმში 1-ლ და მე-2 პუნქტებით ხდება განსახილველი ლოგიკური ფუნქციის **ერთეულოვანი მინიტერმების** განსაზღვრა).

1. ლოგიკური ფუნქციის ჭეშმარიტობის ცხრილიდან ამოვირჩიოთ მოცემული ფუნქციის არგუმენტების ის ნაკრებები, რომლებზედაც ფუნქცია 1-ის ტოლ მნიშვნელობას იღებს;
2. შევადგინოთ არგუმენტების მნიშვნელობათა ამორჩეული არგუმენტების კონიუნქციები; ამისათვის თუ ნაკრებში არგუმენტის მნიშვნელობა 1-ის ტოლია, იგი კონიუნქციაში უცვლელად გადმოვიწეროთ, ხოლო თუ იგი 0-ის ტოლია, მაშინ იგი კონიუნქციაში ინვერსირებული სახით გადმოვიწეროთ;
3. პუნქტ 2-ში ფორმირებული კონიუნქციები (რომლებიც მოცემული ფუნქციის ერთეულოვან მიმნიტერმებს წარმოადგენენ) შევაერთოთ დიზიუნქციის ნიშნებით.
4. დასასრული.

__მაგალითი 4.5.__ ზემოთ ფორმულირებული ალგორითმის დახმარებით განვსაზღვროთ ჭეშმარიტობის **4.8** ცხრილით მოცემული ლოგიკური ფუნქციის დიზიუნქციური სრულყოფილი ნორმალური ფორმის გამოსახულება.

ცხრ. 4.8. ლოგიკური $y = f(x_1,x_2,x_3,x_4)$ ფ უ ნ ქ ც ი ა

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

გ ა დ ა წ ყ ვ ე ტ ა:

1. 0000; 0010; 0011; 0110; 1001; 1100; 1101; 1110;

2. $\overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4}$; $\overline{x_1}\,\overline{x_2}\,x_3\,\overline{x_4}$; $\overline{x_1}\,\overline{x_2}\,x_3\,x_4$; $\overline{x_1}\,x_2\,x_3\,\overline{x_4}$;
$x_1\,\overline{x_2}\,\overline{x_3}\,x_4$; $x_1\,x_2\,\overline{x_3}\,\overline{x_4}$; $x_1\,x_2\,\overline{x_3}\,x_4$; $x_1\,x_2\,x_3\,\overline{x_4}$.

3.
$$y = \overline{x_1}\,\overline{x_2}\,\overline{x_3}\,\overline{x_4} + \overline{x_1}\,\overline{x_2}\,x_3\,\overline{x_4} + \overline{x_1}\,\overline{x_2}\,x_3\,x_4 + \overline{x_1}\,x_2\,x_3\,\overline{x_4} + x_1\,\overline{x_2}\,\overline{x_3}\,x_4 + x_1\,x_2\,\overline{x_3}\,\overline{x_4} + x_1\,x_2\,\overline{x_3}\,x_4 + x_1\,x_2\,x_3\,\overline{x_4};$$

4. დასასრული.

5

(**4.32**) გამოსახულებაში დიზიუნქცია ავიღოთ ყველა ისეთი ნაკრებისათვის, რომლებისათვისაც $f(\sigma_1\sigma_2...\sigma_n)=0$. მივიღებთ ახალი ფუნქციის გამოსახულებას, რომელიც 1-ის ტოლ მნიშვნელობებს მიიღებს არგუმენტების მნიშვნელობათა ისეთ ნაკრებებზე, რომლებზედაც $f(x_1,x_2,\dots,x_n)$ ფუნქცია იღებს 0-ის ტოლ მნიშვნელობებს. ასეთ ფუნქციას $f(x_1,x_2,\dots,x_n)$ **ინვერსირებული ფუნქცია** ეწოდება და აღინიშნება როგორც $\overline{f}\,(x_1,x_2,\dots,x_n)$. მაშასადამე:

$$\overline{f}\,(x_1,x_2,\dots,x_n) = \bigvee_{f(\sigma_1\,\sigma_2\,\dots\sigma_n)=0} x_1^{\sigma_1}\, x_2^{\sigma_2} \cdots x_n^{\sigma_n} \qquad (4.33)$$

მოვახდინოთ (**4.33**) ტოლობის ინვერსირება:

$$\overline{\overline{f}}\,(x_1,x_2,\dots,x_n) = \overline{\bigvee_{f(\sigma_1\,\sigma_2\,\dots\sigma_n)=0} x_1^{\sigma_1}\, x_2^{\sigma_2} \cdots x_n^{\sigma_n}} \qquad (4.34)$$

უკანასკნელი გამოსახულების მარცხენას ნაწილის ინვერსირებისას თუ გამოვიყენებთ ორმაგი უარყოფის (**4.12**) კანონს, მივიღებთ:

$$\overline{\overline{f}}\,(x_1,x_2,\dots,x_n) = f\,(x_1,x_2,\dots,x_n), \qquad (4.35)$$

ხოლო მარჯვენა ნაწილის ინვერსირებისას თუ გამოვიყენებთ დე მორგანის (**4.26**) კანონს, მივიღებთ დიზიუნქციების კონიუნქციას:

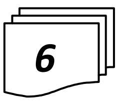
$$\overline{\bigvee_{f(\sigma_1\,\sigma_2\,\dots\sigma_n)=0} x_1^{\sigma_1}\, x_2^{\sigma_2} \cdots x_n^{\sigma_n}} = \bigwedge_{f(\sigma_1\,\sigma_2\,\dots\sigma_n)=0} \left(x_1^{\overline{\sigma_1}} + x_2^{\overline{\sigma_2}} + \cdots + x_n^{\overline{\sigma_n}} \right), \qquad (4.36)$$

სადაც $\bigwedge$ არის კონიუნქციის (ლოგიკური გამრავლების) ნიშანი.

(**4.35**)-სა და (**4.36**)-ს თუ გავითვალისწინებთ (**4.33**)-ში, მივიღებთ:

$$f\,(x_1,x_2,\dots,x_n) = \bigwedge_{f(\sigma_1\,\sigma_2\,\dots\sigma_n)=0} \left(x_1^{\overline{\sigma_1}} + x_2^{\overline{\sigma_2}} + \cdots + x_n^{\overline{\sigma_n}} \right). \qquad \mathbf{(4.37)}$$

(**4.36**) ფორმით ლოგიკური ფუნქციის წარმოდგენას **კონიუნქციური სრულყოფილი ნორმა-ლური ფორმა**, ანუ შემოკლებით – **კსნფ** ეწოდება.



კონიუნქციური სრულყოფილი ნორმალური ფორმაც წარმოადგენს ლოგიკური ფუნქციის გამოსახვის **ანალიზურ** ფორმას. იგი შესაძლებელია მივიღოთ ლოგიკური ფუნქციის ჭეშმარიტობის ცხრილიდან შემდეგი ალგორითმის გა-მოყენებით:

1. ლოგიკური ფუნქციის ჭეშმარიტობის ცხრილიდან ამოვირჩიოთ მოცემული ფუნქცი-ის არგუმენტების ის ნაკრებები, რომლებზედაც ფუნქცია **0**-ის ტოლ მნიშვნელობას იღებს;

2. შევადგინოთ არგუმენტების მნიშვნელობათა ამორჩეული არგუმენტების დიზიუნქცი-ები; ამისათვის თუ ნაკრებში არგუმენტის მნიშვნელობა **0**-ის ტოლია, იგი კონიუნქციაში უცვლელად გადმოვიწეროთ, ხოლო თუ იგი **1**-ის ტოლია, მაშინ იგი კონიუნქციაში ინვე-რსირებული სახით გადმოვიწეროთ;

3. პუნქტ **2**-ში ფორმირებული კონიუნქციები (რომლებიც მოცემული ფუნქციის ნუ-ლოვან მინიტერმებს წარმოადგენს) შევაერთოთ კონიუნქციის ნიშნებით.

4. დასასრული

<u>**მაგალითი 4.6**</u>. ზემოთ ფორმულირებული ალგორითმის დახმარებით განვსაზღვ-როთ ჭეშმარიტობის **4.8** ცხრილით მოცემული ლოგიკური ფუნქციის კონიუნქციური სრულყოფილი ნორმალური ფორმის გამოსახულება.

1. 0001; 0100; 0101; 0111; 1000; 1010; 1011; 1111;

2. $(x_1 + x_2 + x_3 + \overline{x_4})$, $(x_1 + \overline{x_2} + x_3 + x_4)$, $(x_1 + \overline{x_2} + x_3 + \overline{x_4})$, $(x_1 + \overline{x_2} + \overline{x_3} + \overline{x_4})$,

   $(\overline{x_1} + x_2 + x_3 + x_4)$, $(\overline{x_1} + x_2 + \overline{x_3} + x_4)$, $(\overline{x_1} + x_2 + \overline{x_3} + \overline{x_4})$, $(\overline{x_1} + \overline{x_2} + \overline{x_3} + \overline{x_4})$,

3. $(x_1 + x_2 + x_3 + \overline{x_4})$, $(x_1 + \overline{x_2} + x_3 + x_4)$, $(x_1 + \overline{x_2} + x_3 + \overline{x_4})$, $(x_1 + \overline{x_2} + \overline{x_3} + \overline{x_4})$,

   $(\overline{x_1} + x_2 + x_3 + x_4)$, $(\overline{x_1} + x_2 + \overline{x_3} + x_4)$, $(\overline{x_1} + x_2 + \overline{x_3} + \overline{x_4})$, $(\overline{x_1} + \overline{x_2} + \overline{x_3} + \overline{x_4})$,

4. დასასრული

7 გავაანალიზოთ (4.37) ტოლობის მარჯვენა ნაწილი. კონიუნქცია ავიღოთ ყველა $\sigma_1\sigma_2\ldots\sigma_n$ ნაკრებისათვის. სიმარტივისათვის ჩავთვალოთ, რომ $n=2$; ვინაიდან $\{\sigma_1\sigma_2\}=\{00;01;10;11\}$, ამიტომ მივიღებთ:

$$\bigwedge_{f(\sigma_1\sigma_2)=0} (x_1^{\overline{\sigma_1}} + x_2^{\overline{\sigma_2}}) = (x_1^{\overline{0}} + x_2^{\overline{0}})(x_1^{\overline{0}} + x_2^{\overline{1}})(x_1^{\overline{1}} + x_2^{\overline{0}})(x_1^{\overline{1}} + x_2^{\overline{1}}) =$$

$$= (x_1 + x_2)(x_1 + \overline{x_2})(\overline{x_1} + x_2)(\overline{x_1} + \overline{x_2}) \cdot \qquad (4.38)$$

(4.38) გამოსახულებაში თუ გავხსნით ფრჩხილებს და გამოვიყენებთ წინააღმდეგობის (4.15) კანონს. მივიღებთ:

$$\bigwedge_{(\sigma_1\sigma_2)} (x_1^{\overline{\sigma_1}} + x_2^{\overline{\sigma_2}}) = 0. \qquad (4.39)$$

ზოგად შემთხვევაში, როდესაც არგუმენტების რაოდენობა n–ის ტოლია, მივიღებთ:

$$0 = \bigwedge_{(\sigma_1\sigma_2\ldots\sigma_n)} (x_1^{\overline{\sigma_1}} + x_2^{\overline{\sigma_2}} + \cdots + x_n^{\overline{\sigma_n}}). \qquad (4.40)$$

(4.40) გამოსახულების თანახმად **კონსტანტა 0** იშლება 2^n რაოდენობის დიზიუნქციების კონიუნქციებად. მის შემადგენელ კონიუნქციურ წევრებს **ნულის კონსტიტუენტები,** ანუ **მაქსიტერმები** ეწოდება.

ადრე განხილულ **ერთიანის კონსტიტუენტას** ხშირად მხოლოდ **კონსტიტუენტას** სახელით მოიხსენიებენ და **ნულის კონსტიტუენტას** უწოდებენ **ანტიკონსტიტუენტას**.

8 **ᲓᲡᲜᲤ** წარმოადგენს $f(x_1,x_2,\ldots,x_n)$ ფუნქციის დაშლას $x_1,x_2,\ldots,x_n$ ცვლადებად. ზოგადად აღნიშნული ფუნქცია შეიძლება დაიშალოს $x_1,x_2,\ldots,x_k$ ცვლადებად, სადაც $0 < k \leq n$. იმ შემთხვევისათვის, როდესაც $k < n$, (4.30) გამოსახულება იღებს სახეს:

$$f(x_1,x_2,\ldots,x_k, x_{k+1},\ldots, x_n) = \bigvee_{(\sigma_1\sigma_2\ldots\sigma_k)} x_1^{\sigma_1} x_2^{\sigma_2}\ldots x_k^{\sigma_k} \bullet f(\sigma_1,\sigma_2,\ldots,\sigma_k, x_{k+1},x_{k+2},\ldots,x_n), \quad (4.41)$$

სადაც დიზიუნქცია აიღება $x_1,x_2,\ldots,x_k$ არგუმენტების მნიშვნელობათა ყველა შესაძლო $\sigma_1,\sigma_2,\ldots,\sigma_k$ ნაკრებისათვის.

(4.41) გამოსახულება მართებული იქნება თუ $x_1,x_2,\ldots,x_k, x_{k+1},\ldots, x_n$ არგუმენტების მნიშვნელობათა ნებისმიერი $\alpha_1,\alpha_2,\ldots,\alpha_k,\alpha_{k+1},\ldots,\alpha_n$ ნაკრებისათვის იგი გადაიქცევა იგივეობად. შევიტანოთ მასში არგუმენტების მნიშვნელობათა აღნიშნული ნაკრები:

$$f(\alpha_1,\alpha_2,\ldots,\alpha_k,\alpha_{k+1},\ldots,\alpha_n) = \bigvee_{(\sigma_1\sigma_2\ldots\sigma_k)} \alpha_1^{\sigma_1}\alpha_2^{\sigma_2}\ldots\ \alpha_k^{\sigma_k} \bullet f(\sigma_1,\sigma_2,\ldots,\sigma_k, \alpha_{k+1},\ldots,\alpha_n). \qquad (4.42)$$

(4.42) გამოსახულებაში ყველა ისეთი $(\sigma_1,\sigma_2,\ldots,\sigma_k)$ ნაკრებისათვის, რომლებისთვისაც სრულდება პირობა: $\sigma_1 \neq \alpha_1$, $\sigma_2 \neq \alpha_2$, …, $\sigma_k \neq \alpha_k$, მართებულია ტოლობა: $\alpha_1^{\sigma_1}\alpha_2^{\sigma_2}\ldots\ \alpha_k^{\sigma_k}=0$,

ამიტომ მასში ($\sigma_1,\sigma_2,\ldots,\sigma_k$) ნაკრების ნაცვლად დარჩება $\alpha_1,\alpha_2,\ldots,\alpha_k$ ნაკრები; ე.ი. (**4.42**) მიიღებს სახეს:

$$f\,(\alpha_1,\alpha_2,\ldots,\alpha_k,\alpha_{k+1},\ldots,\alpha_n) = \bigvee_{(\sigma_1\,\sigma_2\ldots\sigma_k)} \alpha_1^{\alpha_1}\alpha_2^{\alpha_2}\ldots\ \alpha_k^{\alpha_k} \bullet f(\alpha_1,\alpha_2,\ldots,\alpha_k,\ \alpha_{k+1},\ldots,\alpha_n). \qquad \mathbf{(4.43)}$$

თუ გავითვალისწინებთ იმას, რომ $\alpha_1^{\alpha_1}\alpha_2^{\alpha_2}\ldots\ \alpha_k^{\alpha_k} = 1$, მაშინ (**4.43**) გამოსახულება გადაიქცევა იგივეობად: $f\,(\alpha_1,\alpha_2,\ldots,\alpha_k,\alpha_{k+1},\ldots,\alpha_n) = f\,(\alpha_1,\alpha_2,\ldots,\alpha_k,\alpha_{k+1},\ldots,\alpha_n)$, საიდანაც გამომდინარეობს (**4.41**) გამოსახულების მართობულობა.

კერძო შემთხვევაში როდესაც k=1-ს, (**4.41**) მიიღებს სახეს:

$$f(x_1,x_2,\ldots,\ x_n) = x_1\bullet f\,(1,x_2,\ldots,\ x_n)\ +\ \overline{x_1}\bullet f\,(0,x_2,\ldots,\ x_n). \qquad \mathbf{(4.44)}$$

მიღებული შედეგი საშუალებას გვაძლევს ნებისმიერი x_i არგუმენტის მიხედვით დავშალოთ ლოგიკური $f\,(x_1,x_2,\ldots,x_{i-1},\ x_i,\ x_{i+1}\ldots,\ x_n)$ ფუნქცია შემდეგნაირად:

$$f(x_1,x_2,\ldots,x_{i-1},\ x_i,\ x_{i+1}\ldots,\ x_n) = x_i\ (x_1,x_2,\ldots,x_{i-1},\ 0,\ x_{i+1}\ldots,\ x_n)\ +$$

$$+\ \overline{x_i}\ (x_1,x_2,\ldots,x_{i-1},\ 1,\ x_{i+1}\ldots,\ x_n)\ . \qquad \mathbf{(4.45)}$$

უკანასკნელი გამოსახულება დაედო საფუძვლად ჩვენ მიერ **2005** წელს დამუშავებულ **ბინარული პროგრამების** შედგენის **ფორმალურ მეთოდს**, რომელიც იმავე წელს იქნა გამოქვეყნებული მიმოსვლის გზათა **სანკტ-პეტერბურგის** სახელმწიფო უნივერსიტეტის სამეცნიერო ნაშრომების კრებულში [13]. ბინარული დაპროგრამირების მეთოდი წარმატებით გამოიყენება ლოგიკური ფუნქციების პროგრამული რეალიზაციისათვის.

9

შემდგომში ძირითადად განვიხილავთ დიზიუნქციური სრულყოფილი ნორმალური ფორმით წარმოდგენილ ლოგიკურ ფუნქციებს. $x_1^{\sigma_1}\,x_2^{\sigma_2}\bullet\bullet\bullet\,x_n^{\sigma_n}$ კონიუნქციას ეწოდება **ელემენტალური კონიუნქცია,** თუ მასში ნებისმიერი ლოგიკური ცვლადი მხოლოდ ერთხელ შედის. **ელემენტალური კონიუნქციის რანგი** ეწოდება მასში შემავალი ცვლადების რაოდენობას.

ელემენტალური კონიუნქციების დიზიუნქციას ლოგიკური ფუნქციის **დიზიუნქციური ნორმალური ფორმა** (**ДНФ**) ეწოდება. თუ **ДНФ**-ში შედის ლოგიკური ფუნქციის ყველა არგუმენტი, იგი ჩვენ მიერ ზემოთ განხილულ **ДСНФ**-ად გადაიქცევა. **ДСНФ** წარმოადგენს n–რანგის ელემენტალური კონიუნქციების დიზიუნქციას.

n–რანგის ორ ელემენტალურ კონიუნქციას თუ აქვს $\mathbf{A}\,x_i$ და $\mathbf{A}\,\overline{x_i}$ სახე, სადაც **A** წარმოადგენს n-**1** რანგის ელემენტალურ კონიუნქციას, მაშინ მათი დიზიუნქციის (ლოგიკური შეკრების) შედეგად მიიღება **A** კონიუნქცია:

$$\mathbf{A}\,x_i + \mathbf{A}\,\overline{x_i}\ =\ \mathbf{A}\,(\,x_i + \overline{x_i}\,) = \mathbf{A}. \qquad \mathbf{(4.46)}$$

მოცემულ შემთხვევაში თითქოსდა ორი ელემენტალური კონიუნქციის ერთ ელემენტალურ კონიუნქციად "შეწებება" მოხდა; ამიტომ (**4.46**) სახის გარდაქმნას **"შეწებების წესი"** ეწოდება. შეწებების წესის გამოყენება **ДСНФ**-დან **ДНФ**-ის მიღების საშუალებას გვაძლევს;

თითოეულ ლოგიკურ ფუნქციას აქვს ერთადერთი **ДСНФ** და რამდენიმე **ДНФ.** ამის მაილუსტრირებელ მაგალითად განვიხილოთ **3** არგუმენტზე დამოკიდებული ლოგი-კური ფუნქცია:

$$f_1(x_1,x_2,x_3)\ =\ x_1\,\overline{x_2}\,\overline{x_3}\ +\ x_1\,x_2\,\overline{x_3}\ +\ \overline{x_1}\,x_2\,x_3\ +\ x_1\,x_2\,x_3. \qquad \mathbf{(4.47)}$$

მოცემული ტოლობის მარჯვენა ნაწილი შემდეგნაირად გარდავქმნათ (გარდაქმნის დროს გამოყენებულია გამორიცხული მესამის (**4.16**) კანონი):

$$x_1\,\overline{x_2}\,\overline{x_3}\ +\ x_1\,x_2\,\overline{x_3}\ +\ \overline{x_1}\,x_2\,x_3\ +\ x_1\,x_2\,x_3 = x_1\,\overline{x_3}\,(\overline{x_2} + x_2)\ +\overline{x_1}\,x_2\,x_3\ +\ x_1\,x_2\,x_3\ \ =$$

$$=x_1\,\overline{x_3}\ \ +\overline{x_1}\,x_2\,x_3\ +\ x_1\,x_2\,x_3 \qquad \mathbf{(4.48)}$$

(**4.48**) გამოსახულებას თუ შევიტანთ (**4.47**) გამოსახულებაში მივიღებთ:

$$f_1(x_1,x_2,x_3) = x_1 \overline{x_3} + \overline{x_1} x_2 x_3 + x_1 x_2 x_3. \qquad (4.49)$$

მიღებული ტოლობის მარჯვენა ნაწილი კიდევ შეიძლება შემდეგნაირად გარდაიქმნას:

$$x_1 \overline{x_3} + \overline{x_1} x_2 x_3 + x_1 x_2 x_3 = x_1 \overline{x_3} + x_2 x_3 (\overline{x_1} + x_1) = x_1 \overline{x_3} + x_2 x_3 \qquad (4.50)$$

(**4.50**) გამოსახულებას თუ შევიტანთ (**4.49**) გამოსახულებაში მივიღებთ:

$$f_1(x_1,x_2,x_3) = x_1 \overline{x_3} + x_2 x_3 \qquad (4.51)$$

(**4.47**),(**4.49**) და (**4.51**) გამოსახულებები **3** არგუმენტზე დამოკიდებული ერთი და იგივე ლოგიკური ფუნქციების ანალიზური გამოსახულებებია. ამათგან, (**4.47**) გამოსახულება წარმოადგენს აღნიშნული ფუნქციის **დიზიუნქციურ სრულყოფილ ნორმალურ ფორმას** (**დსნფ**-ს) და იგი ერთადერთია მოცემული ლოგიკური ფუნქციისათვის. (**4.49**) და (**4.51**) გამოსახულებები განხილული ლოგიკური ფუნქციის **დიზიუნქციური ნორმალური ფორმებია** (**დნფ**); მაშასადამე, მოცემულ ფუნქციას აქვს ორი **დნფ.** ზოგიერთ ლოგიკურ ფუნქციას შეიძლება უფრო მეტი რაოდენობის **დნფ**-ები გააჩნდეს.

დნფ-ებს შორის ისეთ **დნფ**-ს, რომელიც ყველაზე მცირე რაოდენობის ლოგიკურ ცვლადს შეიცავს, **მინიმალური დიზიუნქციური ნორმალური ფორმა** (**მდნფ**) ეწოდება. ზემოთ განხულულ შემთხვევაში (**4.49**) გამოსახულებით წარმოდგენილი **დნფ** შეიცავს **8**, ხოლო (**4.51**) გამოსახულებით წარმოდგენილი **დნფ – 4** ლოგიკურ ცვლადს და არ არსებობს უფრო ნაკლები რაოდენობის ცვლადის შემცველი სხვა **დნფ;** მაშასადამე, (**4.51**) წარმოადგენს განხილული ლოგიკური ფუნქციის **მდნფ** –ს.

ლოგიკური ფუნქციის ცნობილი **დსნფ**-დან **მდნფ**–ის მიღების პროცესს **ლოგიკური ფუნქციის მინიმიზაცია** ეწოდება. ლოგიკური ფუნქციების მინიმიზაციის ძირითადი მეთოდები განხილული გვაქვს [2]-ში.

4.6. ლოგიკური ელემენტები: ზოგადი ცნობები და მათი რეალიზაციის საფუძვლები

1 ელემენტალური ლოგიკური ფუნქციების მარეალიზებელ მოწყობილობებს **ლოგიკური ელემენტები** ეწოდება. აღნიშნული ელემენტების ასაგებად გამოყენებული საელემენტო ბაზისაგან დამოკიდებულებით განასხვავებენ მექანიკურ, ელექტრომექანიკურ (ელექტრომაგნიტური რელეებით აგებულ), ელექტრონულ (დიოდებითა და ტრანზისტორებით აგებულ), პნევმატურ, ჰიდრავლიკურ, ოპტიკურ და ა.შ. ლოგიკურ ელემენტებს.

კლოდ შენონის მიგნების თანახმად, ნებისმიერი რთული სტრუქტურის მქონე დისკრეტული მოწყობილობა ახდენს გარკვეული ლოგიკური ფუნქციის რეალიზებას (იხ. პარაგრაფი **4.4**). ეს უკანასკნელი შეიძლება გამოისახოს ფუნქციონალურად სრული სისტემის წარმომქმნელი ელემენტალური ლოგიკური ფუნქციებით, ხოლო ელემენტალური ლოგიკური ფუნქციები რეალიზდება ლოგიკური ელემენტების სახით. აღნიშნულიდან გამომდინარე, სწორია დებულება:

ფუნქციონალურად სრული სისტემის წარმომქმნელი ელემენტალური ლოგიკური ფუნქციების მარეალიზებელი ლოგიკური ელემენტებისაგან წარმოიქმნება საელემენტო ბაზა, რომლის გამოყენებითაც შეიძლება აიგოს ნებისმიერი სირთულის მქონე დისკრეტული მოწყობილობა.

ფუნქციონალურად სრულია:

- ყველა (**თორმეტივე**) ელემენტალური ლოგიკური ფუნქციისაგან შემდგარი სისტემა;
- კონიუნქციის, დიზიუნქციისა და ინვერსიისაგან შემდგარი სისტემა;
- კონიუნქციისა და ინვერსიისაგან შემდგარი სისტემა;

- დიზიუნქციისა და ინვერსიისაგან შემდგარი სისტემა;
- კონიუნქციის უარყოფისაგან (**და-არა** ფუნქციისაგან) შემდგარი სისტემა;
- დიზიუნქციის უარყოფისაგან (**ან-არა** ფუნქციისაგან) შემდგარი სისტემა.

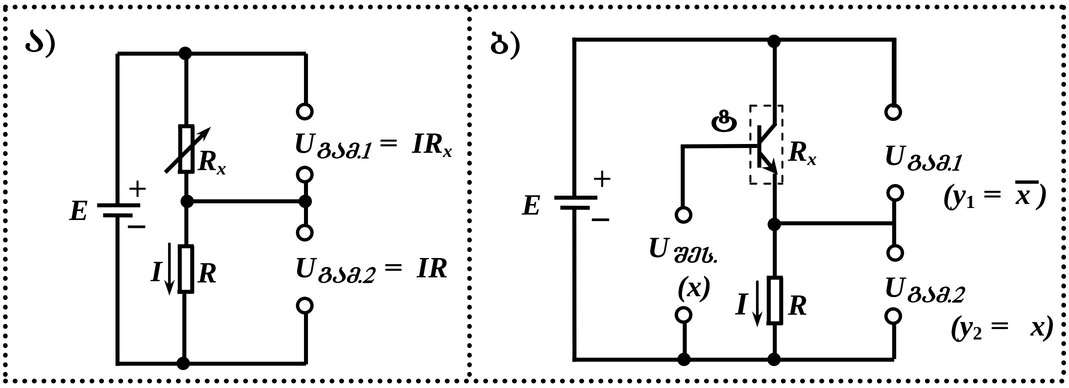
კონიუნქციის, დიზიუნქციის, ინვერსიის, **და-არა** ფუნქციისა და **ან-არა** ფუნქციის მარეალიზებელ ლოგიკურ ელემენტებს შესაბამისად ეწოდებათ:

- კონიუნქტორი, **და** ელემენტი;
- დიზიუნქტორი, **ან** ელემენტი;
- ინვერტორი, **არა** ელემენტი;
- **და-არა** ელემენტი;
- **ან-არა** ელემენტი.

განვიხილოთ ტრანზისტორების გამოყენებით ზემოთ ჩამოთვლილი ლოგიკური ელემენტების აგების პრინციპები.

2 ტრანზისტორების გამოყენების დროს ლოგიკური ცვლადების მნიშვნელობები ელექტრული ძაბვის საშუალებით გამოისახება. ამ მიზნით საჭიროა დამყარდეს გარკვეული შესაბამისობა ლოგიკური ცვლადების მნიშვნელობასა და ძაბვის დონეთა შორის. განასხვავებენ დადებით და უარყოფით **კონვენციებს** (ლათ. **conventio** – "შეთანხმება"). **დადებითი კონვენციის** დროს ძაბვის დაბალ დონეს შეესაბამება ლოგიკური ცვლადის **0**-ის ტოლი, ხოლო ძაბვის მაღალ დონეს – ცვლადის **1**-ის ტოლი მნიშვნელობა. **უარყოფითი კონვენციის** დროს პირიქით, ძაბვის დაბალ დონეს შეესაბამება ლოგიკური ცვლადის **1**-ის ტოლი, ხოლო ძაბვის მაღალ დონეს – ცვლადის **0**-ის ტოლი მნიშვნელობა. აღნიშნული კონვენციები ტოლძალოვნებია, ე.ი. შეიძლება ნებისმიერი მათგანი იქნეს გამოყენებული. ჩვენ გამოვიყენებთ დადებით კონვენციას.

3 ლოგიკური ელემენტების ასაგებად ტრანზისტორების გამოყენების დროს საბაზისო სქემას წარმოადგენს **ძაბვის გამყოფის** სქემა (ნახ.**4.3**).



ნახ. 4.3. ძაბვის გამყოფის რეზისტორული (ა) და ტრანზისტორული (ბ) სქემები

4.3,ა ნახაზზე ნაჩვენები ძაბვის გამყოფის რეზისტორული სქემა შეიცავს ცვლად წინაღობის მქონე $\mathbf{R}_x$ და მუდმივი წინაღობის მქონე **R** რეზისტორებს. ომის კანონის თანახმად:

$$\mathbf{I} = \frac{\mathbf{U}_{გამ.1}}{\mathbf{R}_x} = \frac{\mathbf{U}_{გამ.2}}{R}\ , \qquad (4.52)$$

საიდანაც გამომდინარეობს, რომ მართებულია გამოსახულება:

$$\frac{\mathbf{U}_{გამ.1}}{\mathbf{U}_{გამ.2}} = \frac{R_x}{\mathbf{R}}\ . \qquad (4.53)$$

უკანასკნელი გამოსახულება წარმოადგენს ანალიზური ფორმით ჩაწერილ **ძაბვის გაყოფის პრინციპს,** რომელიც შეიძლება ასე იქნეს ფორმულირებული:

ძაბვის გაყოფის პრინციპის თანახმად, მიმდევრობითად შეერთებულ რეზისტორებზე ძაბვათა ვარდნები ამ რეზისტორების წინაღობების პროპორციულად ნაწილდება.

გავაანალიზოთ (**4.53**) გამოსახულება. მასში არსებულ **R** რეზისტორს აქვს გარკვეული მუდმივი მნიშვნელობა, ხოლო $\mathbf{R}_x$ რეზისტორის წინაღობა შეიძლება ვცვალოთ **0**-დან ისეთ მნიშვნელობამდე, რომელიც მნიშვნელოვნად გადააჭარბებს **R** რეზისტორის წინაღობას. განვიხილოთ შემდეგი სამი შემთხვევა:

- თუ $\mathbf{R}_x \neq \mathbf{0}$ და $\mathbf{R} \neq \mathbf{0}$ და აღნიშნული რეზისტორების წინაღობები თანაზომადი სიდიდეებია, მაშინ $\mathbf{U}_{\text{გამ.1}} \neq \mathbf{0}$, $\mathbf{U}_{\text{გამ.2}} \neq \mathbf{0}$ და ძაბვის ვარდნა ორივე ($\mathbf{R}_x$ და **R**) რეზისტორზე მოხდება, ე.ი. $\mathbf{E} = \mathbf{U}_{\text{გამ.1}} + \mathbf{U}_{\text{გამ.2}}$**;**
- თუ $\mathbf{R}_x = \mathbf{0}$ და $\mathbf{R} \neq \mathbf{0}$, მაშინ $\mathbf{U}_{\text{გამ.1}} = \mathbf{0}$ და ძაბვის ვარდნა მთლიანად **R** რეზიტორზე მოხდება, ე.ი. $\mathbf{E} = \mathbf{U}_{\text{გამ.2}}$**;**
- თუ $\mathbf{R}_x$ რეზისტორის წინაღობა მნიშვნელოვნად აღემატება **R** რეზისტორის წინაღობას, ე.ი. სამართლიანია უტოლობა $\mathbf{R}_x \gg \mathbf{R}$, მაშინ **R** რეზისტორის წინაღობა $\mathbf{R}_x$ რეზისტორის წინაღობასთან შედარებით იმდენად უმნიშვნელოა, რომ იგი შეიძლება დაახლოებითი ნულის ტოლად ჩავთვალოთ, ე.ი. დავწეროთ: $\mathbf{R} \cong \mathbf{0}$**;** ასეთ შემთხვევაში $\mathbf{U}_{\text{გამ.2}} \cong \mathbf{0}$ და ძაბვის ვარდნა მთლიანად $\mathbf{R}_x$ რეზისტორზე მოხდება, ე.ი. $\mathbf{E} = \mathbf{U}_{\text{გამ.1}}$**;**

4.3,ა ნახაზზე ნაჩვენებ სქემაში არსებულ $\mathbf{R}_x$ რეზისტორს თუ **ტ** ტრანზისტორით შევცვლით, მივიღებთ **4.3,ბ** ნახაზზე ნაჩვენებ სქემას. დავუშვათ, რომ ამ უკანასკნელ სქემაში, როგორც ნახაზზეა ნაჩვენები, შესასვლელ $\mathbf{U}_{\text{შეს.}}$ ძაბვას შეესაბამება ლოგიკური (ე.ი. ორობითი) x ცვლადი, ხოლო $\mathbf{U}_{\text{გამ.1}}$ და $\mathbf{U}_{\text{გამ.2}}$ ძაბვებს შესაბამისად ასევე ლოგიკური y_1 და y_2 ცვლადები.

გამოვიყენოთ **დადებითი კონვენცია** და გავაანალიზოთ **4.3,ბ** ნახაზზე მოცემული სქემის მუშაობა:

- როდესაც $x = 0$ (ე.ი. $\mathbf{U}_{\text{შეს.}} = \mathbf{0}$), მაშინ **ტ** ტრანზისტორი ჩაკეტილია და $\mathbf{R}_x = \infty$**;** აქედან გამომდინარე, $\mathbf{R}_x \gg \mathbf{R}$ და ამიტომ: $y_1 = 1$ ($\mathbf{U}_{\text{გამ.1}} = 1$) დ $y_2 = 0$ ($\mathbf{U}_{\text{გამ.2}} = 0$) ;
- როდესაც $x = 1$ (ე.ი. $\mathbf{U}_{\text{შეს.}}$ ძაბვა **ტ** ტრანზისტორის გაღების ძაბვის ტოლია), მაშინ **ტ** ტრანზისტორი ღიაა და $\mathbf{R}_x \cong \mathbf{0}$**;** ამიტომ: $y_1 = 0$ ($\mathbf{U}_{\text{გამ.1}} = 0$) და $y_2 = 1$ ($\mathbf{U}_{\text{გამ.2}} = = \mathbf{IR} = \mathbf{E}$).

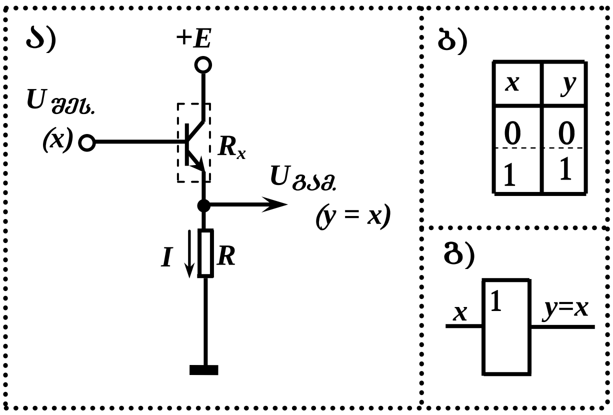
x, y_1 და y_2 ცვლადების ზემოთ მოყვანილი მნიშვნელობები შეიძლება წარმოვადგინოთ **4.9** ცხრილის სახით. როგორც მოცემული ცხრილიდან ჩანს $\mathbf{U}_{\text{გამ.1}}$-ზე ფორმირებული გამოსასვლელი y_2 სიგნალი ახდენს შესასვლელი x სიგნალის ინვერსირებას, ე.ი აღნიშნულ გამოსასვლელზე რეალიზდება ინვერსირების $y = x$ ფუნქცია.

გამოსასვლელი y_2 სიგნალი ახდენს შესასვლელი x სიგნალის გამეორებას, ე.ი აღნიშნულ გამოსასვლელზე რეალიზდება გამეორების $y = x$ ფუნქცია.

ცხრ.4.9. ძაბვის გამყოფის პარამეტრები

x	y_1	y_2
0	1	0
1	0	1

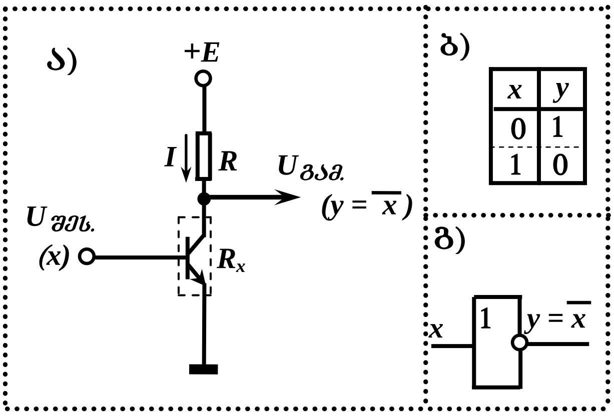
4.3,ბ ნახაზზე ნაჩვენები სქემის საფუძველზე აიგება მამეორებლისა (ნახ.**4.4**) და ინვერტორის (ნახ.**4.5**) სქემები. ისინი წარმოადგენს საბაზისო სქემებს, რომლებითაც აიგება დიზიუნქტორი, კონიუნქტორი, **ᲓᲐ-ᲐᲠᲐ**, **ᲐᲜ-ᲐᲠᲐ** ელემენტები.



ნახ.4.4. მამეორებლის ტრანზისტორული სქემა

4.4,ა ნახაზზე მოყვანილია **მამეორებლის** ტრანზისტორული სქემა, **4.4**,ბ ნახაზზე – მის მიერ რეალიზებული ლოგიკური ფუნქციის ჭეშმარიტობის ცხრილი, ხოლო **4.4**,გ ნახაზზე – ლოგიკურ ელემენტ მამეორებლის პირობითი გამოსახულება;

4.5,ა ნახაზზე მოყვანილია **ინვერტორის** ტრანზისტორული სქემა, **4.ბ**,ბ ნახაზზე – მის მიერ რეალიზებული ლოგიკური ფუნქციის ჭეშმარიტობის ცხრილი, ხოლო **4.5**,გ ნახაზზე – ლოგიკურ ელემენტ ინვერტორის პირობითი გამოსახულება.

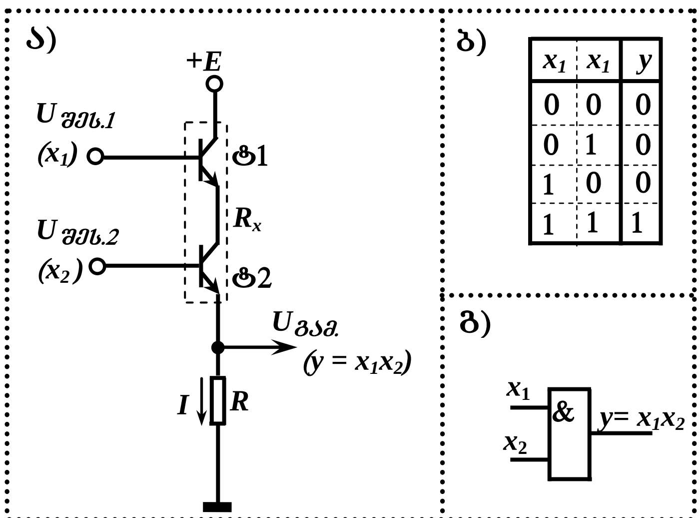


ნახ.4.5. ინვერტორის ტრანზისტორული სქემა

მამეორებლის სქემაში რამდენიმე ტრანზისტორის მიმდევრობითად ჩართვისას მიიღება **კონიუნქტორის სქემა** (ნახ.**4.6**), ხოლო რამდენიმე ტრანზისტორის პარალელურად ჩართვისას – **დიზიუნქტორის სქემა** (ნახ. **4.7**). ასევე, ინვერტორის სქემაში რამდენიმე ტრანზისტორის მიმდევრობითად ჩართვისას მიიღება **ᲓᲐ-ᲐᲠᲐ** ელემენტის სქემა, ხოლო პარალელურად ჩართვისას – **ᲓᲐ-ᲐᲠᲐ** ელემენტის სქემა.

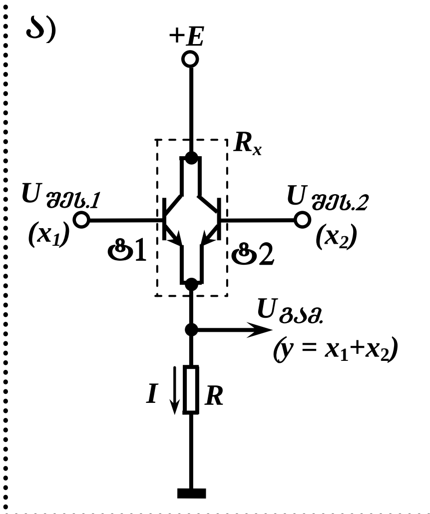
კონიუნქტორის სქემაში (**ნახ.4.6**) $R_x = 0$ –ს მხოლოდ იმ შემთხვევაში, როდესაც ორივე (ტ1 და ტ2) ტრანზისტორი ღიაა, ე.ი. როდესაც $x_1 = x_2 = 1$. ამ დროს ძაბვის ვარდნა მთლიანად R რეზისტორზე ხდება და ამიტომ $y = 1$. ყველა დანარჩენ შემთხვევაში ერთი ტრანზისტორი მაინც ღიაა, ე.ი. $R_x = \infty$. ასეთ შემთხვევებში შეიძლება ჩავთვალოთ, რომ $R \cong 0$ და ამ რეზისტორზე ძაბვის ვარდნა თითქმის ნულის ტოლია, ამიტომ $y=0$.

დიზიუნქტორის სქემაში (ნახ.**4.7**) $R_x >> R$ მხოლოდ იმ შემთხვევაში, როდესაც ორივე ტ1 და ტ2 ტრანზისტორი დაკეტილია, ე.ი. როდესაც $x_1 = x_2 = 0$. ამ დროს $R \cong 0$ და ძაბვის ვარდნა მთლიანად R_x რეზისტორზე ხდება, ე.ი. $y=0$. დანარჩენ შემთხვევებში ღიაა ერთ-ერთი ტრანზისტორი მაინც; ამიტომ $R_x=0$ და ძაბვის ვარდნა მთლიანად R რეზისტორზე ხდება, ე.ი $y=1$.



ნახ.4.6. კონიუნქტორის ტრანზისტორული სქემა

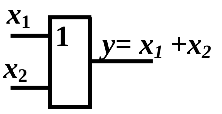
ა)



ბ)

x_1	x_1	y
0	0	0
0	1	1
1	0	1
1	1	1

გ)

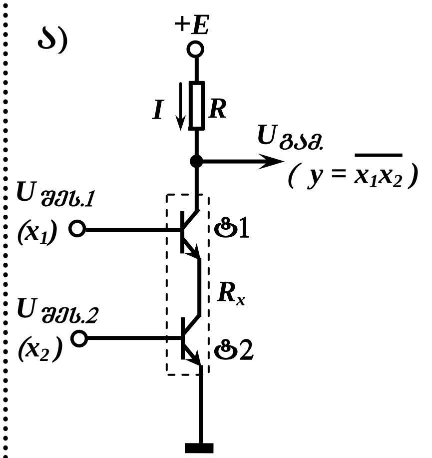


ნახ.4.7. დიზიუნქტორის ტრანზისტორული სქემა

და-არა ელემენტის სქემაში (ნახ.4.8) **R_x=0** მხოლოდ იმ შემთხვევაში, როდესაც ორივე **ტ1** და **ტ2** ტრანზისტორი ღიაა, ე.ი. როდესაც $x_1 = x_2 = 1$. ამ დროს ძაბვის ვარდნა R_x რეზისტორზე (**ტ1** და **ტ2** ტრანზისტორების კოლექტორ-ემიტერის წრედში) ნულის ტოლია, ე.ი. y=0. დანარჩენ შემთხვევებში დაკეტილია ერთ-ერთი ტრანზისტორი მაინც; ამიტომ R_x>>R და ძაბვის ვარდნა მთლიანად R_x რეზისტორზე ხდება (R≅0), ე.ი. y=1.

ან-არა ელემენტის სქემაში (ნახ.4.9) R_x>>R მხოლოდ იმ შემთხვევაში, როდესაც ორივე **ტ1** და **ტ2** ტრანზისტორი დაკეტილია, ე.ი. როდესაც $x_1 = x_2 = 0$. ამ დროს ძაბვის ვარდნა თითქმის მთლიანად R_x რეზისტორზე ხდება და ამიტომ y=1. დანარჩენ შემთხვევებში ღიაა ერთი ტრანზისტორი მაინც, ამიტომ R_x=0 და ძაბვის ვარდნა R_x რეზისტორზე თითქმის ნულის ტოლია, ე.ი. y=0.

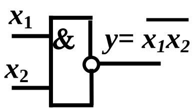
ა)



ბ)

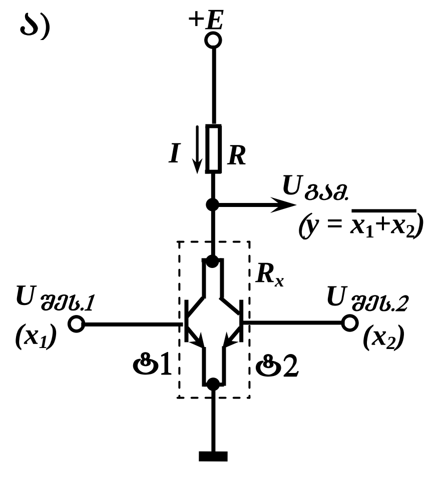
x_1	x_1	y
0	0	1
0	1	1
1	0	1
1	1	0

გ)



ნახ.4.8. და-არა ელემენტის ტრანზისტორული სქემა

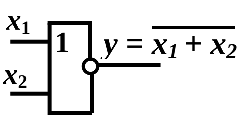
ა)



ბ)

x_1	x_1	y
0	0	1
0	1	0
1	0	0
1	1	0

გ)



ნახ.4.9. ან-არა ელემენტის ტრანზისტორული სქემა

(4.5) – (4.9) ნახაზებზე “**ა**” ასოთი აღნიშნულ ნაწილებში მოცემულია ლოგიკური ელემენტების ტრანზისტორული სქემები, “**ბ**” ასოთი აღნიშნულ ნაწილებში – შესაბამისი ლოგიკური ელემენტების მიერ რეალიზებული ლოგიკურ ფუნქციათა ჭეშმარიტობის ცხრილები, ხოლო “**გ**” ასოთი აღნიშნულ ნაწილებში – ლოგიკური ელემენტების პირობითი გამოსახულებები.

4.7. ლოგიკური ფუნქციების რეალიზება და-არა, ან-არა ელემენტებით

1 როგორც **4.3** პარაგრაფში აღვნიშნეთ, ელემენტალური ფუნქციების ფუნქციონალურად სრულ სისტემას **ბაზისი,** ხოლო ბაზისში შემავალი ფუნქციების მარეალიზებელ ელემენტებს – **ბაზისური ელემენტები** ეწოდება.

ბაზისში შემავალი ელემენტალური ლოგიკური ფუნქციებით შეგვიძლია ნებისმიერი სირთულის ლოგიკური ფუნქცია გამოვსახოთ, ხოლო შესაბამისი **ბაზისური ელემენტებით** შეგვიძლია ნებისმიერი სირთულის ლოგიკური ფუნქციის მარეალიზებელი დისკრეტული მოწყობილობა ავაგოთ.

ლოგიკის (ბულის) ალგებრაში (იხ. **4.4** პარაგრაფი) გამოიყენება კონიუნქციის, დიზიუნქციისა და ინვერსიისაგან შემდგარი ბაზისი. იგი სპეციალურად იქნა შერჩეული ლოგიკის ალგებრისათვის, რადგან მისი გამოყენების დროს მარტივდება ლოგიკური ფუნქციების გარდაქმნის პროცესი; აღნიშნულის გამო კონიუნქციის, დიზიუნქციისა და ინვერსიისაგან შემდგარ ბაზისს პირობითად **ლოგიკის ალგებრის ბაზისი** ვუწოდოთ.

ლოგიკის ბაზისს შეესაბამება სამი ბაზისური ელემენტი: **კონიუნქტორი, დიზიუნქტორი და ინვერტორი**. გარკვეული რაოდენობის აღნიშნული ელემენტების საშუალებით შეიძლება ნებისმიერი სირთულის დისკრეტული მოწყობილობა ავაგოთ; აღნიშნულიდან გამომდინარე, ისინი ფიგურალურად დისკრეტული მოწყობილობის ასაგებად საჭირო ‘‘სამშენებლო მასალად’’ შეიძლება განვიხილოთ. აღნიშნული სამშენებლო მასალის ნომენკლატურა (ნაირსახეობა) სამის ტოლია, ხოლო საერთო რაოდენობა დისკრეტული მოწყობილობის სირთულეზეა დამოკიდებული.

ზემოთ აღნიშნული ‘‘სამშენებლო მასალის’’ ნომენკლატურის (ნაირსახეობის) შემცირება ამაღლებს დისკრეტული მოწყობილობის ტექნოგენურობას. ასეთი შემცირება შესაძლებელი იქნებოდა ლოგიკის ალგებრის ბაზისად რომ აგვეღო **და-არა** ფუნქცია, რომელსაც შეესაბამება ერთადერთი საბაზისო ელემენტი – **და-არა** ელემენტი;

ანალოგურ შედეგს მივიღებდით ბაზისად **ან-არა** ფუნქციის აღების დროსაც, რადგან ამ შემთხვევაშიც გვექნებოდა ერთადერთი საბაზისო ელემენტი - **ან-არა** ელემენტი.

ასეთი გადაწყვეტები არ იქნა მიღებული იმის გამო, რომ ერთი ფუნქციის შემცველი ზემოთ ჩამოთვლილი ბაზისების გამოყენება მნიშვნელოვნად გაართულებდა ლოგიკის ალგებრაში შესასრულებელ გარდაქმნებს; ამიტომ შექმნილი სიტუაციიდან მოინახა შემდეგი გამოსავალი:

- **ლოგიკის ალგებრაში ლოგიკურ გამოსახულებათა გარდაქმნებისათვის გამოყენებული იქნეს კონიუნქციის, დიზიუნქციისა და ინვერსიის ოპერაციები;**
- **გარდაქმნების შედეგად მიღებული ლოგიკური ფუნქციის გამოსახულებაში კონიუნქციის, დიზიუნქციისა და ინვერსიის ოპერაციები შეიცვალოს და-არა ან ან-არა ოპერაციებით;**
- **დისკრეტული მოწყობილობა აიგოს ლოგიკური ფუნქციის იმ გამოსახულებით, რომელიც შეიცავს მხოლოდ და-არა ან ან-არა ოპერაციებს.**

სწორედ **და-არა** და/ან **ან-არა** ელემენტები წარმოადგენს საბაზისო ელემენტებს ინტეგრალურ სქემებში, რომლებშიც გამოყენებულია:

- რეზისტორულ-ტრანზისტორული (**RTL; РТЛ**) ლოგიკა (რომლებიც ფართოდ იყო გავრცელებული **XX** საუკუნის **60**-იან წლებში);
- დიოდურ-ტრანზისტორული ლოგიკა (**DTL; ДТЛ**);
- ტრანზისტორულ-ტრანზისტორული (**TTL; ТТЛ**) ლოგიკა;

● მეტალ-ოქსიდ-ნახევარგამტარულ ტრანზისტორებით აგებული კომპლემენტარული (**CMOS; КМОП**) ლოგიკა და ა.შ.

მოცემულ თავში განვიხილავთ **ᲓᲐ-ᲐᲠᲐ** (**ᲐᲜ-ᲐᲠᲐ**) ოპერაციებით დიზიუნქციის, კონიუნქციისა და ინვერსიის ოპერაციების შეცვლის პრინციპებს.

ლოგიკური **ᲓᲐ-ᲐᲠᲐ** $(f_1 = \overline{x_1 x_2})$ და **ᲐᲜ-ᲐᲠᲐ** $(f_2 = \overline{x_1 + x_2})$ ფუნქციების ჭეშმარიტობის **2.9** ცხრილიდან ჩანს, რომ ისინი ურთიერთშებრუნებულ, ანუ, **ურთიერთორად ფუნქციებს** წარმოადგენს ე.ი. სრულდება პირობა:

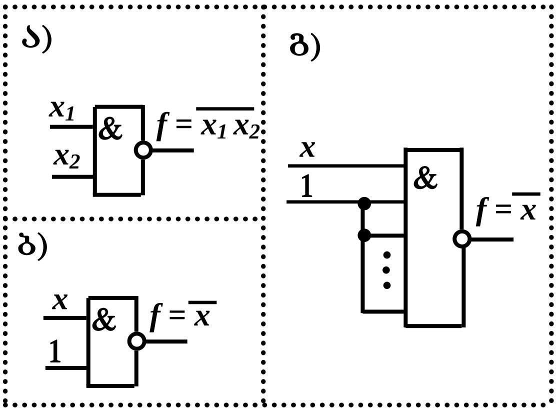
$$f_1 = \overline{f_2} \quad (f_2 = \overline{f_1}). \tag{4.54}$$

ცხრ.4.9. **ᲓᲐ-ᲐᲠᲐ** , **ᲐᲜ-ᲐᲠᲐ** ფუნქციები

x_1	x_2	$f_1 = \overline{x_1 x_2}$	$f_2 = \overline{x_1 + x_2}$
0	0	1	0
0	1	1	0
1	0	1	0
1	1	0	1

მართლაც, x_1 და x_2 არგუმენტების მნიშვნელობათა იმ ნაკრებებზე, რომლებზედაც f_1 ფუნქცია იღებს 1-ის ტოლ მნიშვნელობას, f_2 ფუნქცია ნულის ტოლი ხდება და პირიქით. ამიტომ **ᲓᲐ-ᲐᲠᲐ** ელემენტისათვის მართებული დებულების შებრუნებული (ინვერსირებული) დებულება მართებული იქნება **ᲐᲜ-ᲐᲠᲐ** ელემენტისათვის და პირიქით.

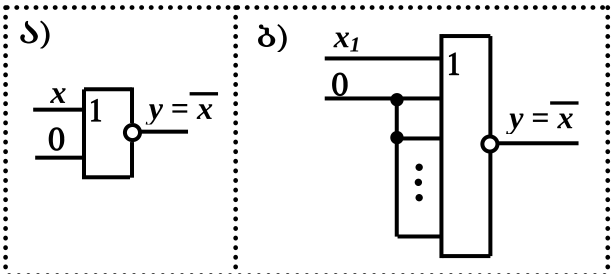
ᲓᲐ-ᲐᲠᲐ ფუნქციის (იხ.ცხრ.**4.9**) შესაბამისი **ᲓᲐ-ᲐᲠᲐ** ელემენტის პირობითი გამოსახულება ნაჩვენებია **4.10,ა** ნახაზზე. **4.9** ცხრილიდან ჩანს, რომ როდესაც f_1 ფუნქციის ერთ-ერთი (მაგალითად, x_2) არგუმენტი **1**-ის ტოლია, მაშინ ფუნქცია იღებს დარჩენილი (ჩვენს შემთხვევაში x_1) არგუმენტის მნიშვნელობის შებრუნებულ მნიშვნელობას. მაშასადამე თუ ორშესასვლელიანი **ᲓᲐ-ᲐᲠᲐ** ელემენტის(ნახ.**4.10,ა**) ერთ-ერთ შესასვლელზე მუდმივად მივაწოდებთ ლოგიკურ სიგანალ **1**-ს (ნახ.**4.10,ბ**), მაშინ გამოსასვლელზე მივიღებთ მეორე შესასვლელზე არსებული სიგნალის ინვერსირებულ მნიშვნელობას. ვინაიდან x_i სიგნალის ინდექსს ამ შემთხვევაში მნიშვნელობა არა აქვს, ამიტომ ნახაზზე იგი უგულველყოფილია. **4.10,ბ** ნახაზზე ნაჩვენები სქემა ახდენს x სიგნალის ინვერსირებას, მაშასადამე, იგი წარმოადგენს **ინვერტორს** . **ᲓᲐ-ᲐᲠᲐ** ელემენტით აგებული ინვერტორის ზოგად შემთხვევაში, როდესაც მისი შესასვლელების რაოდენობა წინასწარ არ არის განსაზღვრული, **4.10,გ** ნახაზზე არის ნაჩვენები.



ნახ.4.10. ინვერტორის რეალიზაცია **ᲓᲐ-ᲐᲠᲐ** ელემენტის საშუალებით

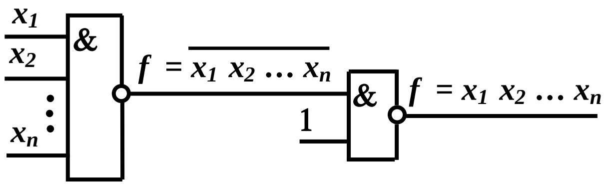
ვინაიდან **ᲐᲜ-ᲐᲠᲐ** ფუნქცია წარმოადგენს **ᲓᲐ-ᲐᲠᲐ** ფუნქციის შებრუნებულ (ინვერსირებულ) ფუნქციას, ამიტომ **ᲐᲜ-ᲐᲠᲐ** ელემენტით აგებულ ინვერტორს ექნება **4.11** ნახაზზე ნაჩვენები სახე.

ორმაგი უარყოფის კანონის თანახმად, $\overline{\overline{x_1 x_2}} = x_1 x_2$, მაშასადამე, **ᲓᲐ-ᲐᲠᲐ** ფუნქციის ინვერსირებით მიიღება კონიუნქციის ფუნქცია; ეს საშუალებას გვაძ-

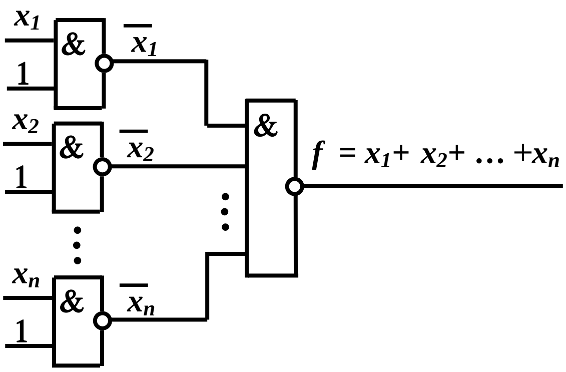


ნახ.4.11. ინვერტორის რეალიზაცია **ᲐᲜ-ᲐᲠᲐ** ელემენტის საშუალებით

ძლევს **და-არა** ელემენტების საშუალებით ავაგოთ კონიუნქტორი. ამისათვის მიმდევრობით უნდა შევაერთოთ ორი **და-არა** ელემენტი, რომელთაგანაც მეორე შეასრულებს ინვერტორის ფუნქციას (ნახ. 4.12). მიღებული სქემის გამოსასვლელზე გვექნება პირველ **და-არა** ელემენტის შესასვლელებზე მიწოდებული სიგნალების კონიუნქცია.



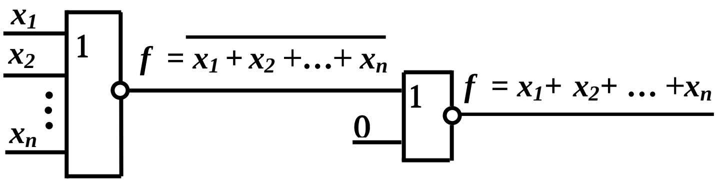
ნახ. 4.12. კონიუნქტორის რეალიზება **და-არა** ელემენტებით.



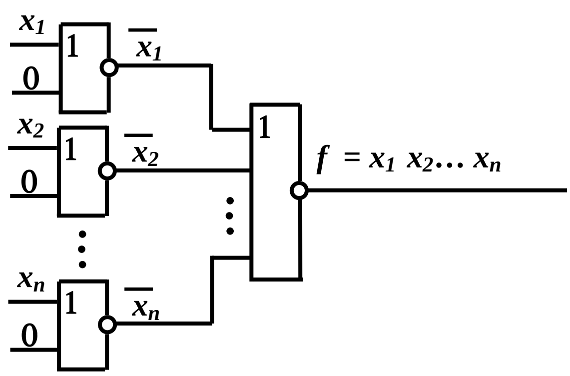
ნახ. 4.13. დიზიუნქტორის რეალიზება

და-არა ელემენტის შესასვლელებზე მივაწოდოთ $\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}$ სიგნალები (ნახ. 4.13); მის გამოსასვლელზე მივიღებთ $x_1, x_2, \ldots, x_n$ სიგნალების დიზიუნქციას. მართლაც $\overline{\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}} = \overline{\overline{x_1}} + \overline{\overline{x_2}} + \ldots + \overline{\overline{x_n}} = x_1, x_2, \ldots, x_n$. მაშასადამე, 4.13 ნახაზზე ნაჩვენები სქემა წარმოადგენს **და-არა** ელემენტებით აგებულ დიზიუნქტორს.

4.12 ნახაზზე ნაჩვენებ სქემაში **და-არა** ელემენტებს თუ შევცვლით **ან-არა** ელემენტებით (**ნახ.4.14**) მივიღებთ **ან-არა** ელემენტებით რეალიზებული დიზიუნქტორის სქემას. ვინაიდან დიზიუნქცია არის კონიუნქციის შებრუნებული (ინვერსირებული) ფუნქცია;



ნახ. 4.14. დიზიუნქტორის რეალიზება **ან-არა** ელემენტებით.



ნახ. 4.15. კონიუნქტორის რეალიზება **ან-არა** ელემენტებით

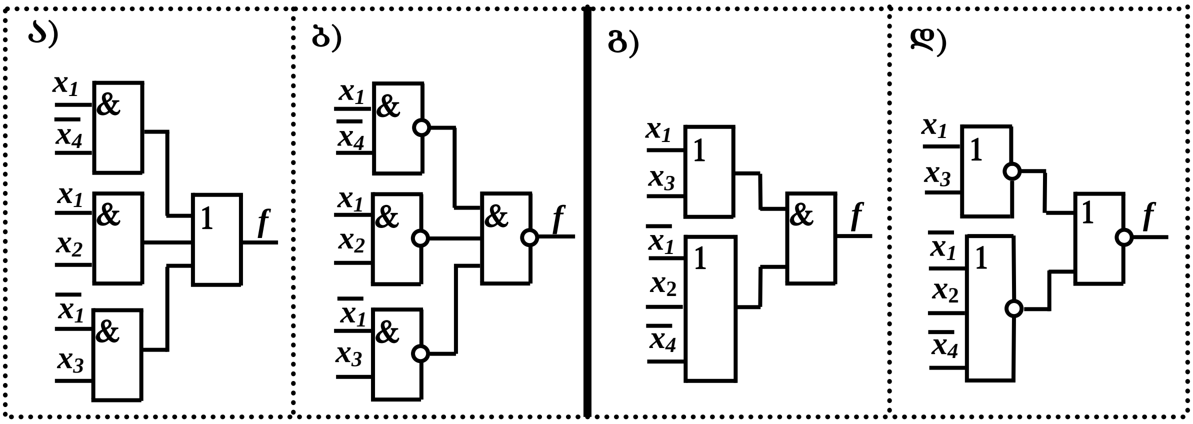
ასევე, 4.13 ნახაზზე ნაჩვენებ სქემაში **და-არა** ელემენტებს თუ შევცვლით **ან-არა** ელემენტებით (ნახ.4.15), მივიღებთ **ან-არა** ელემენტებით რეალიზებული კონიუნქტორის სქემას.

განვიხილოთ **მინიმალური დიზიუნქციური ნორმალური ფორმით** (იხ. პარაგრაფი 4.5) წარმოდგენილი ლოგიკური ფუნქცია:

$$f = x_1\, \overline{x_4} + x_1\, x_2 + \overline{x_1}\, x_3 \qquad (4.55)$$

მოცემული გამოსახულების რეალიზაციის სქემა **4.16,ა** ნახაზზეა მოცემული. მოვახდინოთ მისი მარჯვენა ნაწილის ორმაგი ინვერსირება (ორმაგი უარყოფის კანონის ძალით აღნიშნული ოპერაცია ლოგიკური ფუნქციის მნიშვნელობას ვერ ცვლის) და გამოვიყენოთ დე მორგანის კანონი:

$$f = \overline{\overline{x_1\,\overline{x}_4 + x_1\,x_2 + \overline{x}_1\,x_3}} = \overline{\overline{x_1\,\overline{x}_4}\;\overline{x_1\,x_2}\;\overline{\overline{x}_1\,x_3}} = (\,x_1|\,x_4)|(x_1|\,x_2)|(\,x_1|\,x_3)$$



ნახ. 4.16. და-არა, ან-არა ელემენტებით ლოგიკურ გამოსახულებათა რეალიზება

მოცემული გამოსახულების რეალიზაციის სქემა **4.16,ბ** ნახაზზეა მოყვანილი. მიღებული სქემები (იხ. ნახ. **4.16,ა,ბ**) ერთმანეთის ანალოგიურებია და მხოლოდ ლოგიკური ელემენტების ტიპებით განსხვავდება ერთმანეთისაგან. ამგვარად, შესაძლებელია შემდეგი ზოგადი კანონის ფორმულირება:

მინიმალური დიზიუნქციური ნორმალური ფორმით (მდნფ-ით) წარმოდგენილი ლოგიკური ფუნქციის მარეალიზებელ ლოგიკურ სქემაში არსებული ელემენტები შეიძლება და-არა ელემენტებით შევცვალოთ, რის შედეგადაც აღნიშნული ფუნქცია და-არა ელემენტების საშუალებით აგებული აღმოჩნდება.

ან-არა ბაზისის გამოყენებით მოწყობილობების აგების მეთოდიკა **და-არა** ბაზისის გამოყენებით მოწყობილობების აგების ანალოგიურია, ოღონდ ამ დროს სარეალიზებელი ლოგიკური ფუნქცია **მდნფ-**ის ნაცვლად უნდა **მინიმალური კონიუნქციური ნორმალური ფორმის (კდნფ)** სახით უნდა იყოს წარმოდგენილი. ზემოთ განხილული ფუნქციის **კდნფ**-ს აქვს შემდეგი სახე:

$$f = (x_1 + x_3)\,(\overline{x_1} + x_2 + \overline{x}_4). \qquad (4.56)$$

მოცემული ფუნქციის მარეალიზებელი სქემა **4.16,გ** ნახაზზეა ნაჩვენები. წინა შემთხვევის ანალოგიურად მოვახდინოთ (**4.56**) ტოლობის მარჯვენა ნაწილის ორმაგი ინვერსირება:

$$f = \overline{\overline{(x_1 + x_3)\,(\overline{x_1} + x_2 + \overline{x}_4)}} = \overline{\overline{x_1 + x_3} + \overline{\overline{x}_1 + x_2 + \overline{x}_4}} = (x_1{\downarrow}x_3){\downarrow}(x_1{\downarrow}x_2{\downarrow}x_4)\,. \qquad (4.57)$$

მაშასადამე, მიღებული ფუნქცია მხოლოდ **ან-არა** ოპერაციებით არის გამოსახული. ამ ფუნქციის რეალიზებისათვის, ჯერ ერთი, იმდენივე **ან-არა** ელემენტი გვჭირდება, რამდენი კონიუნქტორი და დიზიუნქტორია გამოყენებული **კდნფ**-ის მარეალიზებელ სქემაში, და მეორეც, ამ უკანასკნელში არსებული ელემენტების **ან-არა** ელემენტებით

შეცვლისას უცვლელი რჩება ელემენტების შესასვლელების რაოდენობაც. აღნიშნულიდან გამომდინარე შეიძლება შემდეგი წესის ფორმურირება:

> **მინიმალური კონიუნქციური ნორმალური ფორმით (მკნფ-ით) წარმოდგენილი ლოგიკური ფუნქციის მარეალიზებელ ლოგიკური სქემაში არსებული ელემენტები შეიძლება ან-არა ელემენტებით შევცვალოთ, რის შედეგადაც აღნიშნული ფუნქცია ან-არა ელემენტების საშუალებით აგებული აღმოჩნდება.**

(**4.57**) გამოსახულების რეალიზაციის სქემა **4.16,დ** ნახაზზეა მოყვანილი.

მოცემულ პარაგრაფში გამოვიყენეთ ლოგიკური ფუნქციის მინიმალური დიზიუნქციური და კონიუნქციური ფორმები. ქვემოთ განვიხილავთ კარნოს მატრიცის დახმარებით აღნიშნული ფორმების მიღების მეთოდს.

4.8. ლოგიკური ფუნქციების მინიმიზაციის საკითხები

1 როგორც **4.5** პარაგრაფში ვაჩვენეთ, ლოგიკური ფუნქციის **დიზიუნქციური სრულყოფილი ნორმალური ფორმიდან (დსნფ-დან)** ლოგიკის ალგებრის კანონების გამოყენებით შესაძლებელია მიღებული იქნეს **დიზიუნქციური ნორმალური ფორმები (დნფ-ები)**, რომლებიც **დსნფ**-ასთან შედარებით ნაკლები რაოდენობის ლოგიკურ ცვლადებს შეიცავს.

ნებისმიერ ლოგიკურ ფუნქციას გააჩნია ერთადერთი **დსნფ**. რაც შეეხება **დნფ**-ებს, მათი რაოდენობა შეიძლება იყოს **0**-ის ტოლი ან რამდენიმე.

რამდენიმე **დნფ**-ების არსებობის შემთხვევაში, ისინი ერთმანეთისაგან შეიძლება განსხვავდებოდეს დიზიუნქციური წევრების (ე.წ. **მინიტერმების** ანუ **1-ის კონსტიტუენტების**) ან ლოგიკური ცვლადების რაოდენობებით.

მინიმალური რაოდენობის მინიტერმების მქონე **დნფ**-ს ეწოდება ლოგიკური ფუნქციის **შემოკლებული დიზიუნქციური ნორმალური ფორმა** (**შდნფ**), ხოლო მინიმალური რაოდენობის ლოგიკური ცვლადების მქონე **დნფ**-ს – **მინიმალური დიზიუნქციური ნორმალური ფორმა** (**მდნფ**);

ლოგიკური ფუნქციის მინიმიზაციის ამოცანაა ამ ფუნქციის **დსნფ**-დან მისი **მდნფ**-ის პოვნა. აღსანიშნავია, რომ ფუნქციის **მდნფ** ზოგჯერ შეიძლება დაემთხვეს მის **შდნფ**-ს, ზოგჯერ კი მის მისაღებად საჭიროა **შდნფ**-დან გამოირიცხოს ზოგიერთი ელემენტალური კონიუნქცია.

2 ფუნქციის დიზიუნქციური სრულყოფილი ნორმალური ფორმის შესახებ ზემოთ მოყვანილი მსჯელობა მთლიანად შეიძლება გავრცელდეს მოცემული ფუნქციის კონიუნქციური სრულყოფილი ნორმალური ფორმისთვისაც (**კსნფ**-საც). მაშასადამე, არსებობს ლოგიკური ფუნქციის:

- **კონიუნქციური ნორმალური ფორმები** (**კნფ**-ები) რომლებიც **კსნფ**–თან შედარებით შეიცავს ნაკლები რაოდენობის ლოგიკურ ცვლადებს;
- **შემოკლებული კონიუნქციური ნორმალური ფორმა** (**შკნფ**), რომელიც შეიცავს მინიმალური რაოდენობის დიზიუნქციურ წევრებს (ე.წ. **მაქსიტერმებს,** ანუ **ანტიკონსტიტუენტებს**);
- **მინიმალური კონიუნქციური ნორმალური ფორმა** (**მკნფ**), რომელიც შეიცავს მინიმალური რაოდენობის ლოგიკურ ცვლადებს.

ლოგიკური ფუნქციის მინიმიზაციის ამოცანაა ამ ფუნქციის **კსნფ**-დან მისი **მკნფ**-ის პოვნა.

ქვემოთ ჩვენ განვიხილავთ მხოლოდ დიზიუნქციურ სრულყოფილი ნორმალური ფორმით წარმოდგენილ ლოგიკურ ფუნქციებს; მიღებული შედეგები,საჭიროების შემთხვევაში, შეიძლება ადვილად განზოგადდეს კონიუნქციური სრულყოფილ ნორმალური ფორმით წარმოდგენილი ლოგიკური ფუნქციებისთვისაც.

3

ლოგიკის ალგებრის კანონების საშუალებით **ᲓᲡᲜᲤ**-დან **ᲛᲓᲜᲤ**-ის პოვნის მეთოდს **ლოგიკური ფუნქციის მინიმაზაციის ალგებრული მეთოდი** (იხ. **4.5** პარაგრაფი) ეწოდება. აღნიშნული მეთოდის ნაკლია მაღალი შრომატევადობა და მინიმიზაციის თითოეული ამოცანისათვის ინდივიდუალური მიდგომის აუცილებლობა; აღნიშნული შრომატევადობის შესამცირებლად დამუშავებული იქნა ლოგიკური ფუნქციის მინიმიზაციის მეთოდები, რომლებშიც ზემოთ აღნიშნული მიზანი მინიმიზაციის პროცესი მაქსიმალური ფორმალიზების გზით იქნა მიღწეული (ე.ი. გამოირიცხა თითოეული შემთხვევისათვის მკაცრი ინდივიდუალური მიდგომის აუცილებლობა). ასეთ მეთოდებს შორის გამოირჩევა ლოგიკური ფუნქციების მინიმიზაციის:

● **ანალიზური** მეთოდები, რომელთაგანაც ფართოდ გავრცელდა ქვაინისა და მაკ-კლასკის მიერ შემოთავაზებული მეთოდი**,** რომელიც **ქვაინისა და მაკ-კლასკის მეთოდის** სახელწოდებით არის ცნობილი;

● **კოორდინატული მეთოდი,** რომელშიც კარნოს ბარათებია გამოყენებული და ამიტომ მას **კარნოს ბარათებით მინიმიზაციის მეთოდსაც** უწოდებენ.

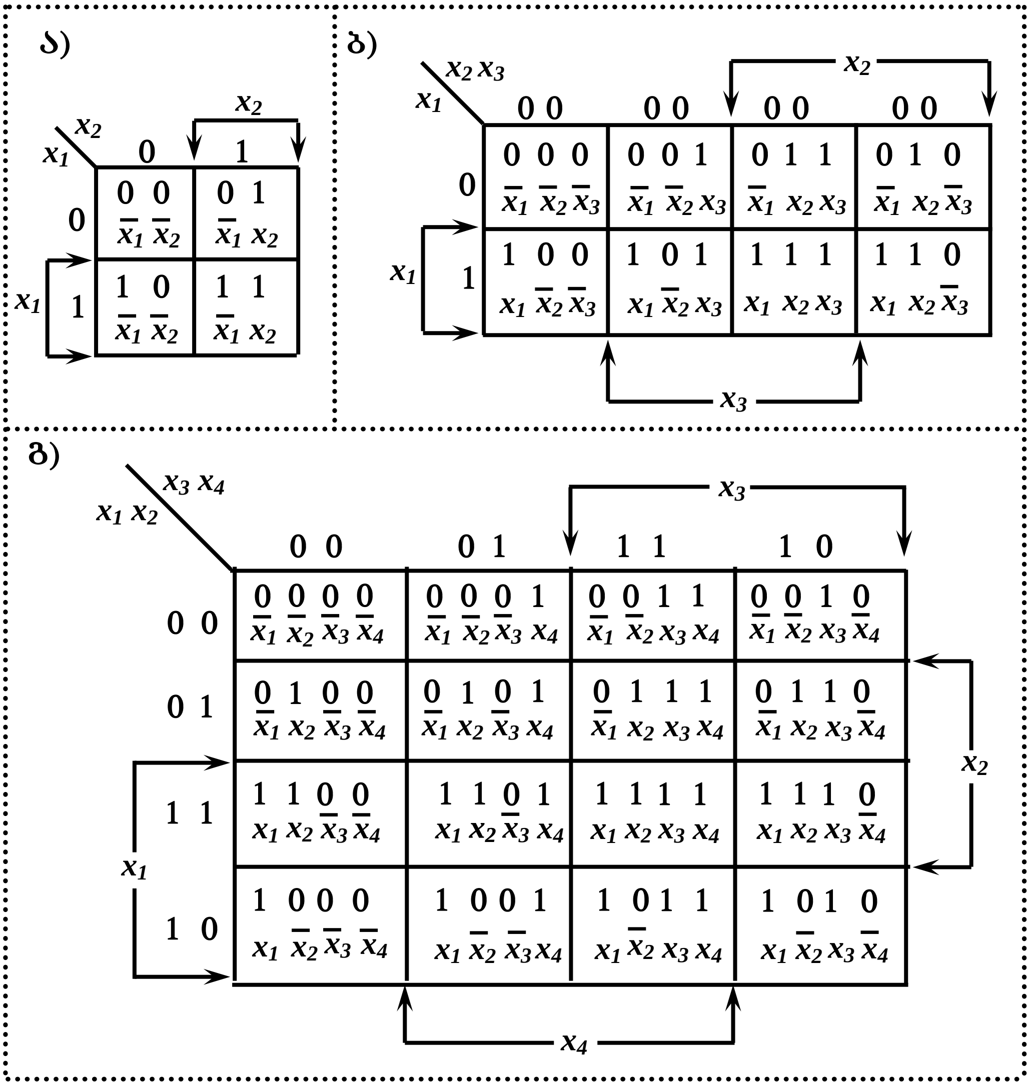
ზემოთ აღნიშნული ორივე მეთოდი ჩვენ [2]-ში გვაქვს დეტალურად განხილული; ამიტომ საერთო სურათის შესაქმნელად მოცემულ შემთხვევაში მხოლოდ კარნოს ბარათის დახმარებით მინიმიზების მეთოდის განხილვით შემოვიფარგლებით, რომელიც გამოირჩევა კომპაქტურობითა და თვალსაჩინოებით.

4

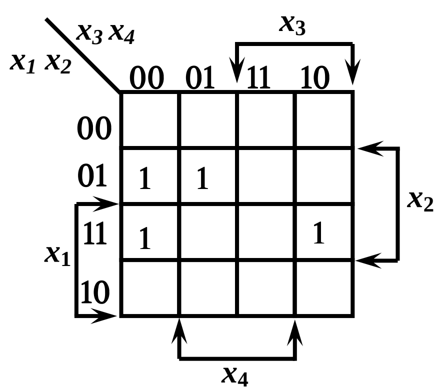
n რაოდენობის ლოგიკური ცვლადებისათვის აგებული **კარნოს ბარათი (მატრიცა)** წარმოადგენს 2^n რაოდენობის უჯრების გარკვეული წესით ფორმირებულ ერთობლიობას. **4.17** ნახაზზე მოცემულია n = **2;3;4** რაოდენობის ლოგიკური ცვლადებისათვის აგებული კარნოს ბარათები (მატრიცები). როგორც ნახაზებიდან ჩანს, ნებისმიერი ლოგიკური ცვლადისათვის კარნოს ბარათში გამოყოფილია ორი მწკრივი ან სვეტი მაინც, რომელთაგანაც ერთ-ერთი მათგანი განკუთვნილია ამ ცვლადის **0**-ის ტოლი, ხოლო მეორე – **1**-ის ტოლი მნიშვნელობისათვის; ამასთანავე, თითოეული მწკრივი ან სვეტი განკუთვნილია ორზე არა უმეტესი რაოდენობის ცვლადის მნიშვნელობისათვის; მათი ერთობლიობა წარმოქმნის ამ მწკრივის ან სვეტის **კოორდინატებს.**

უჯრის მისამართი მიიღება თუ მისი შესაბამისი მწკრივის კოორდინატს მარჯვნიდან მივუწერთ ამავე უჯრის სვეტის კოორდინატს. **უჯრის მისამართი** წარმოადგენს ***n*** რაოდენობის ბიტებისაგან შედგენილ ორობით რიცხვს. ბიტების თანრიგები მარცხნიდან მარჯვნივ დავნომროთ **1; 2; ... ;** ***n*** ციფრებით.

უჯრის მისამართის გამომსახველი ორობითი რიცხვის i-ური თანრიგი ($i \in \{1;2;...,n\}$) ვუწოდოთ მარჯვნიდან i-ურ ადგილზე მდგარ ციფრს. აღნიშნული რიცხვის i-ურ თანრიგში მდგარ **0**-ს თუ შევცვლით $\overline{x_i}$ სიმბოლოთი, ხოლო **1**-ს - x_i სიმბოლოთი, მივიღებთ განხილული ორობითი რიცხვის შესაბამის **მინიტერმს.** მაგალითად, როგორც **4.17,გ** ნახაზიდან ჩანს, მოცემული კარნოს მეორე მწკრივის კოორდინატაა **01,** ხოლო მეოთხე სვეტის კოორდინატა – **10**; აღნიშნული მწკრივისა და სვეტის გადაკვეთაზე არსებული **უჯრის მისამართია 0110;** ამ უკანასკნელ ორობით რიცხვს კი შეესაბამება $\overline{x_1}\,x_2\,x_3\,\overline{x_4}$ მინიტერმი. მინიტერმში შემავალი ცვლადების რაოდენობას მოცემული **მინიტერმის რანგი** ეწოდება. კარნოს ბარათის ნებისმიერ უჯრას შეესაბამება ***n*** რანგის გარკვეული მინიტერმი (ნახ. **4.17,ა,ბ,გ**).



ნახ. 4.17. ორი (**ა**), სამი (**ბ**) და ოთხი (**გ**) ლოგიკური ცვლადისათვის აგებული კარნოს ბარათები



ნახ. 4.18. კარნოს ბარათის მეშვეობით ლოგიკური ფუნქციის წარმოდგენის მაგალითი

n რაოდენობის არგუმენტზე დამოკიდებული ფუნქციის დიზიუნქციური სრულყოფილი ნორმალური ფორმა წარმოადგენს ***n*** რანგის მინიტერმების დიზიუნქციას. მის თითოეულ წევრს ***n*** რაოდენობის ცვლადებისათვის აგებულ კარნოს ბარათში შეესაბამება გარკვეული უჯრა. ამ უჯრებს თუ აღვნიშნავთ 1-ით, ხოლო დანარჩენებს დავტოვებთ შეუვსებლად ან აღვნიშნავთ 0-ით, მივიღებთ **ლოგიკური ფუნქციის გამოსახვის კოორდინატულ სახეს.**

მაგალითად, განვიხილოთ $n=4$ ცვლადზე დამოკიდებული ლოგიკური ფუნქცია:

$$f\,(x_1x_2\,x_3\,x_4) = \overline{x_1}\;x_2\;\overline{x_3}\;\overline{x_4}\;+\overline{x_1}\;x_2\;\overline{x_3}\;x_4\;+x_1\;x_2\;\overline{x_3}\;\overline{x_4}\;+x_1\;x_2\;x_3\;\overline{x_4}\,. \qquad (4.58)$$

მოცემული ფუნქციის კოორდინატული სახე **4.18,** ნახაზზეა ნაჩვენები.

5

კარნოს ბარათის უჯრების წარმომქმნელ ორ ხაზს, რომელთა შორისაც მოთავსებულ მწკრივებს ან სვეტებს შეესაბამება x_i ცვლადის 1-ის ტოლი მნიშვნელობები, x_i **ცვლადის ღერძები** ვუწოდოთ. ჩავთვალოთ, რომ **ბარათის უჯრა მოთავსებულია x_i ცვლადის ღერძების შიგა მხარეს, თუ ამ უჯრის მინიტერმი შეიცავს აღნიშნული ცვლადის უინვერსიო გამოსახულებას.** წინააღმდეგ შემთხვევაში ჩავთვალოთ, რომ იგი **მოთავსებულია x_i ცვლადის ღერძების გარე მხარეს.**

4.17,ა,ბ,გ ნახაზებზე x_i **ცვლადის ღერძები** სპეციალურად ისრებითაა მითითებული; ისრების შემაერთებელი ხაზებით კი აღნიშნული ღერძების შიგა მხარეებია შემოსაზღვრული.

კარნოს ბარათი ისეა აგებული, რომ x_i ღერძის სხვადასხვა მხარეზე სიმეტრიულად განლაგებულ მინიტერმებს აქვს $x_i\mathbf{M}$ და $\overline{x_i}\mathbf{M}$ სახე, სადაც არის n-1 რანგის მინიტერმი. მათი დიზიუნქცია შეწებების წესის თანახმად შეიძლება მინიტერმით შეიცვალოს:

$$x_i\mathbf{M} + \overline{x_i}\mathbf{M} = \mathbf{M}\;(x_i + \overline{x_i}) = \mathbf{M}. \qquad (4.59)$$

x_i ცვლადის ღერძის სხვადასხვა მხარეზე სიმეტრიულად განთავსებულ უჯრებს **მეზობელი უჯრები** ეწოდება. სიმეტრიულობის კანონის თანახმად, მეზობელი უჯრების რაოდენობა ლუწია და მათი რაოდენობა 2^d-ს ტოლია, სადაც d ($d\in\{0;1;...;n\}$) – უჯრების **სიმეტრიულობის ღერძების** რაოდენობაა. 2^d რაოდენობის მეზობელი უჯრების ერთობლიობას d-**კუბი** ვუწოდოთ.

d-**კუბის** უჯრედებს ჰკვეთს x_i-ცვლადების რაოდენობის ღერძები, რომლებსაც d-**კუბის** სიმეტრიულობის ღერძები ვუწოდოთ. კერძოდ:

- 2^1=2 რაოდენობის მეზობელი უჯრა წარმოქმნის **1-კუბს** და მას აქვს სიმეტრიულობის ერთი ღერძი;
- 2^2=4 რაოდენობის მეზობელი უჯრა წარმოქმნის **2-კუბს** და მას აქვს სიმეტრიიულობის ორი ღერძი;
- 2^3 რაოდენობის მეზობელი უჯრა წარმოქმნის **3-კუბს** და მას აქვს სიმეტრიულობის სამი ღერძი და ა.შ.

6

d-**კუბის** უჯრედების შესაბამისი მინიტერმები შეიცავს n-**d** რანგის ერთი და იგივე **M** მინიტერმს. იგი d-**კუბის** ნებისმიერი უჯრის შესაბამისი მინიტერმიდან ამ კუბის სიმეტრიულობის ღერძების ამოშლით მიიღება. d-**კუბის** უჯრების შესაბამისი მინიტერმების დიზიუნქცია n-**d** რანგის **M** მინიტერმის ტოლია, რომელსაც d-**კუბის ფუძე** ეწოდება.

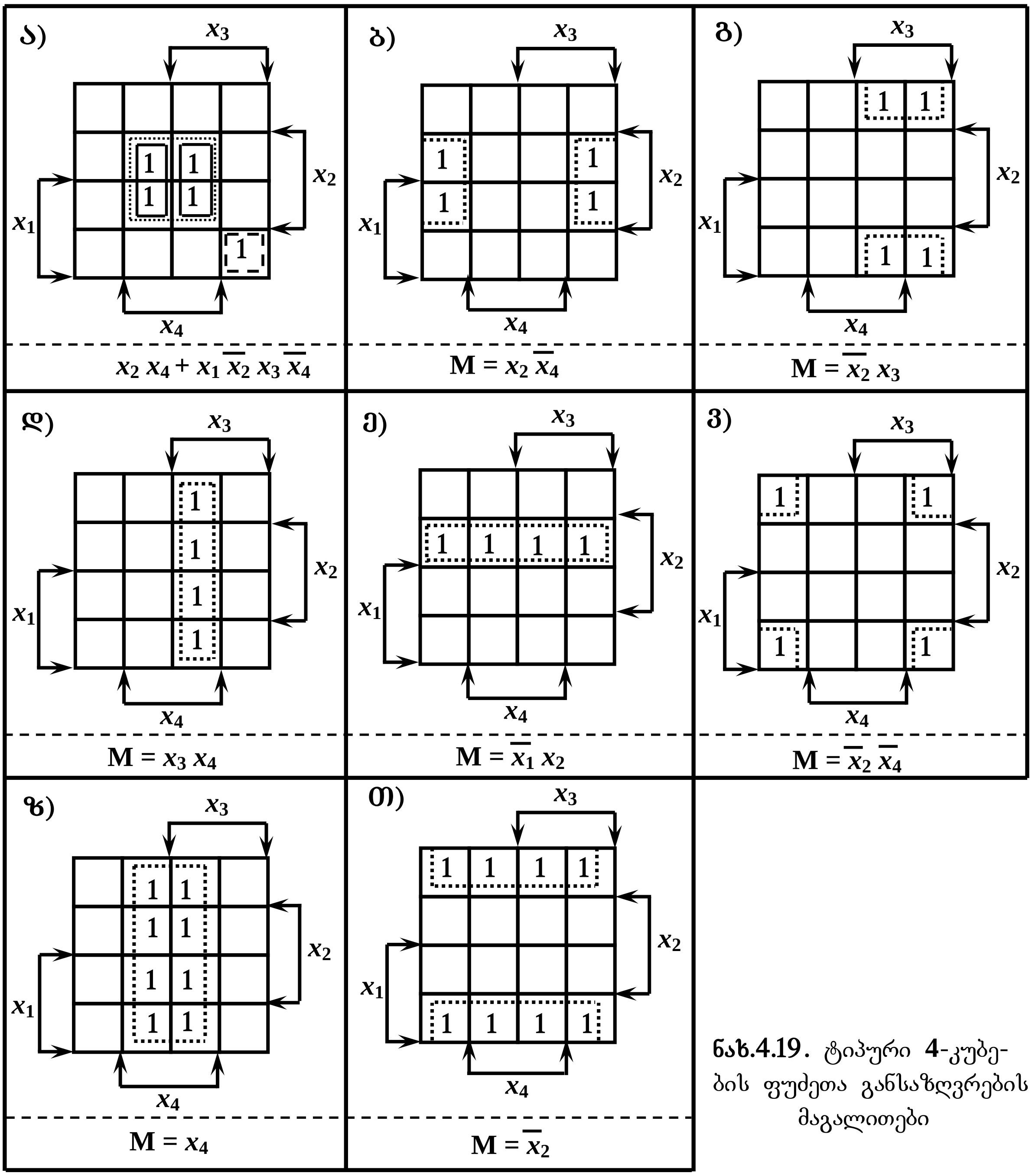
განვიხილოთ **4.19,ა** ნახაზზე ნაჩვენები კარნოს მატრიცა. მისი საშუალებით წარმოდგენილია შემდეგი ლოგიკური ფუნქცია; მის დიზიუნქციურ სრულყოფილ ნორმალურ ფორმას აქვს შემდეგი სახე:

$$F(x_1\,x_2\,x_3\,x_4\,) \;=\; \overline{x_1}\,x_2\,\overline{x_3}\,x_4 \;+\; \overline{x_1}\,x_2\;x_3\,x_4 + x_1\,x_2\,\overline{x_3}\,x_4 \;+$$
$$+\;x_1\,x_2\;x_3\,x_4 + \;x_1\,\overline{x_2}\;x_3\,\overline{x_4}\;\;. \qquad (4.60)$$

წერტილოვანი ხაზებით აგებული კვადრატით შემოფარგლულია **2-კუბის** შესაბამისი 2^2=4 რაოდენობის უჯრები. ამ კუბის სიმეტრიულობის ღერძებია (ე.ი. ამ კუბს ჰკვეთენ) x_1 და x_3 ცვლადების ღერძები. კუბის ფუძის მისაღებად განვიხილოთ მისი ნებისმიერი უჯრის მინიტერმი, მაგალითად $\overline{x_1}\,x_2\,\overline{x_3}\,x_4$; ამ მინიტერმიდან x_1 და x_3 ცვლადების ამოშლით მივიღებთ **2-კუბის** M= $x_2\,x_4$ ფუძეს. განხილული **2-კუბის** უჯრების შესაბამისი მი-

ნიტერმების დიზიუნქცია შეიძლება შეიცვალოს აღნიშნული ფუძით, რის შედეგადაც (**4.60**) გამოსახულება მიიღებს შემდეგ სახეს:

$$F(x_1\ x_2\ x_3\ x_4\ ) \ =\ x_2\ x_4\ +\ x_1\ x_2\ x_3\ x_4\ . \qquad (4.61)$$



ნახ.4.19. ტიპური **4**-კუბე-ბის ფუძეთა განსაზღვრების მაგალითები

ზემოთ განხილული **2-კუბში** შედის უწყვეტი ხაზებით აგებული მართკუთხედებით შემოფარგლული ორი **1-კუბი** და თავად იგი არ შედის არც ერთ სხვა **d-კუბში.** ასეთ **d-კუბს** მაქსიმალური **d-კუბი** ვუწოდოთ. ზოგადად, **მაქსიმალური d-კუბი** ეწოდება ისეთ **d-**კუბს, რომელიც არ შედის რომელიმე სხვა **d-კუბის** შემადგენლობაში**.**

ტიპური **4-კუბის M** ფუძეები **4.19** ნახაზზეა მოცემული.

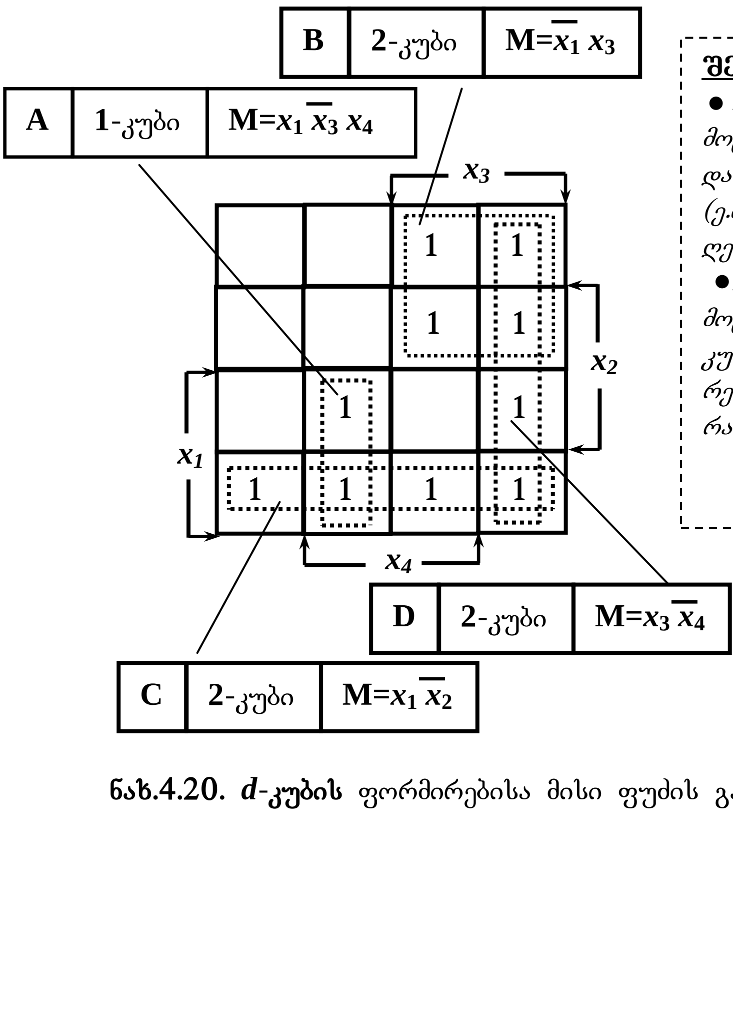
კარნოს ბარათის გამოყენებით **ლოგიკური ფუნქციის მინიმიზაციის პროცედურა** ასეთია:

- ლოგიკური ფუნქცია წარმოვადგინოთ კარნოს ბარათის საშუალებით;
- მიღებულ კარნოს ბარათში წარმოვქმნათ მინიმალური რაოდენობის ისეთი მაქსიმალური d-კუბები, რომლებშიც გაერთიანებული იქნება 1-იანებით აღნიშნული ყველა ან მაქსიმალური რაოდენობის უჯრები;
- განვსაზღვროთ წარმოქმნილი d-კუბების ფუძეთა გამოსახულებები;
- შევადგინოთ d-კუბების მიღებული გამოსახულებებისა და იმ უჯრების მინიტერმების დიზიუნქციები, რომლებიც არ შედის არც ერთ წარმოქმნილ d-კუბში (თუ ასეთი უჯრები არსებობს);
- მიღებული გამოსახულება წარმოადგენს ლოგიკური ფუნქციის მინიმალურ დიზიუნქციურ ნორმალურ ფორმას.

7 განვიხილოთ მაგალითი. დავუშვათ, რომ მოცემულია ოთხ არგუმენტზე დამოკიდებული შემდეგი ლოგიკური ფუნქცია:

$$
\begin{aligned}
F(x_1\, x_2\, x_3\, x_4) = {} & \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 + \bar{x}_1 \bar{x}_2 x_3 x_4 + \bar{x}_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 x_3 x_4 + \\
& + x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 + x_1 \bar{x}_2 \bar{x}_3 x_4 + x_1 \bar{x}_2 x_3 \bar{x}_4 + x_1 x_2 x_3 x_4 + \\
& + x_1 x_2 \bar{x}_3 x_4 + x_1 x_2 x_3 \bar{x}_4 .
\end{aligned}
\qquad (4.62)
$$

იგი შეიცავს 10 მინიტერმს, რომელთაგანაც თითოეულის რანგი 4-ის ტოლია. ლოგიკური ფუნქციის **დიზიუნქციური სრულყოფილი ნორმალური ფორმის ჯამური რანგი** მასში შემავალი მინიტერმების რანგების ჯამს ვუწოდოთ და პირობითად **Q** ასოთი აღვნიშნოთ; მოცემული ფუნქციისათვის **Q = 40**.



შენიშვნები:

- ***2-კუბი*** *ნიშნავს, რომ უჯრების მოცემულ ჯგუფს ჰკვეთს 2 სხვადასხვა ლოგიკური ცვლადის ღერძი (ე.ი. ამ უჯრების სიმეტრიულობის ღერძების რაოდენობა 2-ის ტოლია);*
- ***1-კუბი*** *ნიშნავს, რომ უჯრების მოცემულ ჯგუფს ჰკვეთს 1 ლოგიკური ცვლადის ღერძი (ე.ი. ამ უჯრების სიმეტრიულობის ღეძების რაოდენობა 1-ის ტოლია);*

ნახ.4.20. *d*-კუბის ფორმირებისა მისი ფუძის განსაზღვრის მაგალითი

ზემოთ მოყვანილი პროცედურის გამოყენებით ვიპოვოთ მისი მინიმალური დიზიუნქციური ნორმალური ფორმა.

მოცემული ლოგიკური ფუნქცია წარმოვადგინოთ კარნოს ბარათის საშუალებით. მივიღებთ **4.20** ნახაზზე მოყვანილ კარნოს ბარათს.

● **4.20** ნახაზზე ნაჩვენებ კარნოს ბარათში გამოვყოთ მაქსიმალური **d–კუბები**. ნახაზზე ისინი წერტილოვანი ხაზებით აგებული ოთკუთხედებითაა ნაჩვენები. როგორც ნახაზიდან ჩანს, მოცემულ შემთხვევაში არსებობს სამი **2-კუბი** და ერთი **1-კუბი**, რომლებიც მოიცავს 1-იანებით აღნიშნულ ყველა უჯრას;

● განვსაზღვროთ მიღებული ***d*–კუბების M ფუძეები:**

1) A უჯრები წარმოქმნის **1-კუბს** და აქვს სიმეტრიულობის ერთი, კერძოდ x_2 ცვლადის ღერძი. აღნიშნული კუბის ფუძის საპოვნელად საკმარისია ავიღოთ ამ კუბის წარმომქმნელი ერთ-ერთი ნებისმიერი უჯრის შესაბამისი მინიტერმი და მისგან ამოვშალოთ x_2 ცვლადი, რომლის ღერძიც ჰკვეთს მოცემულ **1-კუბს**. აღნიშნული ოპერაციის შესრულების შედეგად მივიღებთ მოცემული კუბის შემდეგ ფუძეს:

$$\mathbf{A:}\quad \mathbf{M} = x_1 \cancel{x_2}\, \overline{x_3}\, x_4 = x_1 \cancel{\overline{x_2}}\, \overline{x_3}\, x_4 = x_1\, \overline{x_3}\, x_4. \qquad \mathbf{(4.63)}$$

2) B უჯრები წარმოქმნის **2-კუბს** და აქვს სიმეტრიულობის ორი, კერძოდ x_2 და x_4 ცვლადების ღერძები. აღნიშნული კუბის ფუძის საპოვნელად საკმარისია ავიღოთ ამ კუბის წარმომქმნელი ერთ-ერთი ნებისმიერი უჯრის შესაბამისი მინიტერმი და მისგან ამოვშალოთ x_2 და x_4 ცვლადები, რომელთა ღერძებიც ჰკვეთს მოცემულ **2-კუბს**. აღნიშნული ოპერაციის შესრულების შედეგად მივიღებთ მოცემული კუბის შემდეგ ფუძეს:

$$\mathbf{B:}\quad \mathbf{M} = \overline{x_1}\, \cancel{\overline{x_2}}\, x_3 \cancel{x_4} = \overline{x_1}\, \cancel{\overline{x_2}}\, x_3 \cancel{\overline{x_4}} = \overline{x_1} \cancel{x_2}\, x_3 \cancel{x_4} = \overline{x_1} \cancel{x_2}\, x_3 \cancel{\overline{x_4}} = \overline{x_1}\, x_3. \qquad \mathbf{(4.64)}$$

3) C უჯრები წარმოქმნის **2-კუბს** და აქვს სიმეტრიულობის ორი, კერძოდ x_3 და x_4 ცვლადების ღერძები. აღნიშნული კუბის ფუძის საპოვნელად საკმარისია ავიღოთ ამ კუბის წარმომქმნელი ერთ-ერთი ნებისმიერი უჯრის შესაბამისი მინიტერმი და მისგან ამოვშალოთ x_3 და x_4 ცვლადები, რომელთა ღერძებიც ჰკვეთს მოცემულ **2-კუბს**. აღნიშნული ოპერაციის შესრულების შედეგად მივიღებთ მოცემული კუბის შემდეგ ფუძეს:

$$\mathbf{C:}\quad \mathbf{M} = x_1\, x_2 \cancel{x_3} \cancel{\overline{x_4}} = x_1\, \overline{x_2}\, \cancel{\overline{x_3}} \cancel{x_4} = x_1\, \overline{x_2}\, \cancel{x_3} \cancel{x_4} = x_1\, \overline{x_2}\, \cancel{x_3} \cancel{\overline{x_4}} = x_1\, \overline{x_2}\,. \qquad \mathbf{(4.65)}$$

4) D უჯრები წარმოქმნის **2-კუბს** და გააჩნია სიმეტრიულობის ორი, კერძოდ x_1 და x_2 ცვლადების ღერძები. აღნიშნული კუბის ფუძის საპოვნელად საკმარისია ავიღოთ ამ კუბის წარმომქმნელი ერთ-ერთი ნებისმიერი უჯრის შესაბამისი მინიტერმი და მისგან ამოვშალოთ x_1 და x_2 ცვლადები, რომელთა ღერძებიც ჰკვეთს მოცემულ **2-კუბს**; აღნიშნული ოპერაციის შესრულების შედეგად მივიღებთ მოცემული კუბის შემდეგ ფუძეს:

$$\mathbf{D:}\quad \mathbf{M} = \cancel{\overline{x_1}} \cancel{\overline{x_2}}\, x_3\, \overline{x_4} = \cancel{\overline{x_1}} \cancel{x_2}\, x_3\, \overline{x_4} = \cancel{x_1} \cancel{x_2}\, x_3\, \overline{x_4} = \cancel{x_1} \cancel{\overline{x_2}}\, x_3\, \overline{x_4} = x_3\, \overline{x_4}\,. \qquad \mathbf{(4.66)}$$

მივიღეთ **4.20** ნახაზზე არსებული ყველა **d**-კუბის ფუძე; ამიტომ გადავიდეთ ზემოთ მოყვანილი პროცედურის შემდეგ საფეხურზე.

● შევადგინოთ მიღებული **d**-კუბების მიღებული გამოსახულებების დიზიუნქცია:

$$f(x_1\, x_2\, x_3\, x_4) = x_1\, \overline{x_3}\, x_4 + \overline{x_1}\, x_3 + x_1\, \overline{x_2} + x_3\, \overline{x_4}\,. \qquad \mathbf{(4.67)}$$

მიღებული გამოსახულება წარმოადგენს ლოგიკური (**4.62**) ფუნქციის მინიმალურ დიზიუნქციურ ნორმალურ ფორმას. მისი ჯამური რანგი **9**-ის ტოლია. ვინაიდან განსაზღვრების თანახმად ლოგიკური მინიმალური დიზიუნქციური ნორმალური ფორმა შეიცავს

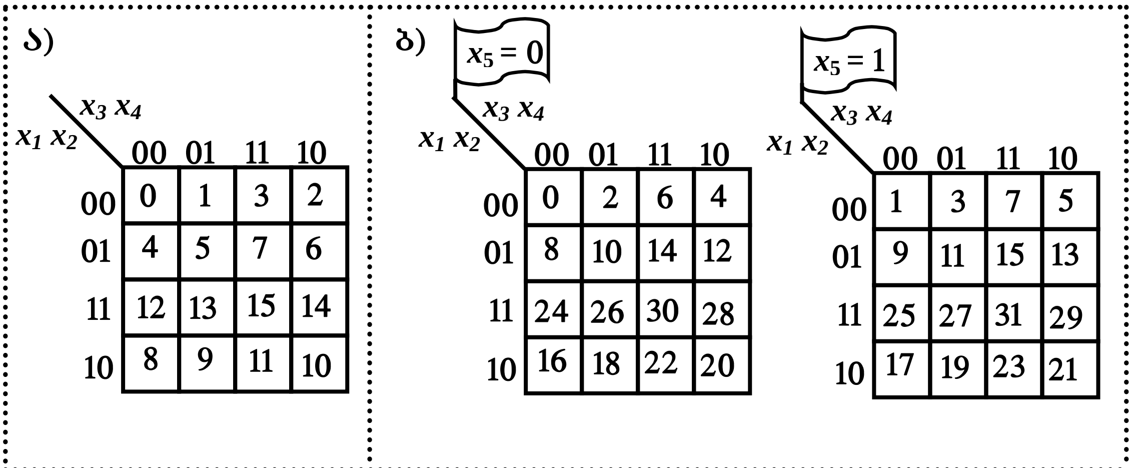
მინიმალური რაოდენობის ლოგიკურ ცვლადებს, ამიტომ მინიმალური იქნება მისი ჯამური რანგიც, ე.ი. შეგვიძლია დავწეროთ: $\mathbf{Q_{min} = 9}$; როგორც ვხედავთ მისი ჯამური რანგი (**4.62**) გამოსახულების ჯამურ რანგზე დაახლოებით **4,4**-ჯერ ნაკლებია ($\frac{Q}{Q_{min}} = \frac{40}{9} \cong$ **4,4**).

რადგან (**4.62**) და (**4.67**) გამოსახულებები **იგივური გამოსახულებებია**, ამიტომ მათი რეალიზაციის შედეგად მიღებული დისკრეტული მოწყობილობები აბსოლუტურად ერთი და იგივე ფუნქციებს შეასრულებს; ოღონდ (**4.65**) გამოსახულების რეალიზებით მიღებული დისკრეტულ მოწყობილობას ექნება გაცილებით მარტივი სტრუქტურა, რაც მნიშვნელოვნად აამაღლებს მის ტექნიკურ, ეკონომიკურ და საიმედოობით მაჩვენებლებს.

იგივური ლოგიკური გამოსახულებების მარეალიზებელ დისკრეტულ მოწყობილობებს **ეკვივალენტური დისკრეტული მოწყობილობები** ეწოდება. განსაზღვრების თანახმად, ეკვივალენტური დისკრეტული მოწყობილობები ერთი და იგივე ლოგიკური ფუნქციების მარეალიზებელი სხვადასხვა სტრუქტურის მქონე დისკრეტულ მოწყობილობებია.

8

კარნოს ბარათის უჯრების მისამართები შეიძლება ათობითი რიცხვებითაც გამოვსახოთ. ამისათვის მისამართების მაჩვენებელი ორობითი რიცხვები ათვლის ათობით სისტემაში უნდა გადავიყვანოთ. **4.21,ა** ნახაზზე ნაჩვენებია **4** ცვლადისათვის აგებული კარნოს მატრიცის უჯრების მისამართების ათობითი რიცხვებით გამოსახვის მაგალითი.



ნახ.4.21. 4 და 5 ცვლადისათვის აგებული კარნოს ბარათები

n=5 ცვლადების შემთხვევაში $2^5 = 32$ უჯრიანი ბარათის ნაცვლად მოსახერხებელია გამოვიყენოთ ორი **16** უჯრიანი ბარათი (ნახ.**4.21,ბ**), რომელთაგანაც ერთ-ერთი შეესაბამება x_5-ის **0**-ის ტოლ, ხოლო მეორე – **1**-ის ტოლ მნიშვნელობას. ასევე, n=6 ცვლადისათვის ერთი $2^6 = 64$ უჯრიანი ბარათის ნაცვლად მოსახერხებელია გამოვიყენოთ ოთხი **16** უჯრიანი ბარათი

9

კარნოს ბარათის საშუალებით გამოსახული ლოგიკური ფუნქციის **მინიმალური კონიუნქციური ნორმალური ფორმის** მისაღებად საჭიროა განვიხილოთ ბარათის ცარიელი (**0**-ებით მონიშნული) უჯრები. ვიპოვოთ მათი მინიმალური რაოდენობის მაქსიმალური ***d*–კუბების M ფუძეები**, მათში მოვახდინოთ ცვლადების ინვერსირება და მიღებული ცვლადები ერთმანეთს დიზიუნქციის ნიშნებით დავაკავშიროთ. აღნიშნული ოპერაციის ჩატარების შედეგად მივიღებთ (***n-d***) რანგის **N** **მაქსიტერმებს.** მიღებული მაქსიტერმებისა და ***d*–კუბებში გაუერთიანებელი უჯრების** (ასეთი უჯრების არსებობის შემთხვევაში) **მაქსიტერმების** კონიუნქცია წარმოადგენს ფუნქციის **მინიმალურ**

კონიუნქციურ ნორმალურ ფორმას. კარნოს ნებისმიერი უჯრის მაქსიტერმს მივიღებთ ამ უჯრის მინიტერმისაგან; ამისათვის უნდა მოვახდინოთ ამ უკანასკნელში არსებული ცვლადების ინვერტირება და მიღებული ცვლადები ერთმანეთს დავუკავშიროთ დიზიუნქციის ნიშნებით.

4.9. არასრულად განსაზღვრული ლოგიკური ფუნქციების მინიმიზაცია. ლოგიკურ ფუნქციათა სისტემების მინიმიზაცია

1 ლოგიკურ ფუნქციებს, რომლებიც არგუნენტების თითოეულ ნაკრებზე იღებს წინასწარ განსაზღვრულ მნიშვნელობებს, **სრულად განსაზღვრული ლოგიკური ფუნქციები** ეწოდება. არსებობს ლოგიკური ფუნქციები, რომელთა მნიშვნელობები არგუმენტების ზოგიერთ ნაკრებზე მკაცრად განსაზღვრული არ არის; ასეთ ნაკრებზე მათ შეუძლია მიიღოს ნებისმიერი, ე.ი. როგორც **0**-ის, ასევე **1**-ის ტო-ლი მნიშვნელობები; ასეთ ფუნქციებს **არასრულად განსაზღვრული ლოგიკური ფუნქციები** ეწოდება. არგუმენტების ნაკრებებს, რომლებზედაც ფუნქციების ნებისმიერი მნიშვნელობის მიღება შეუძლია – **მნიშვნელობათა ნეიტრალური ნაკრებები ეწოდება.** აღნიშნული ნაკრებების შესაბამისი მინიტერმები და მაქსიტერმები შესაბამისად წარმოადგენენ **ნეიტრალურ მინიტერმებს და მაქსიტერმებს.** მაგალითად, ჭეშმარიტობის **4.10** ცხრილით წარმოდგენილ ფუნქციაში განსაზღვრული არ არის არგუმენტების მნიშვნელობათა **0001, 1000, 1001** და **1110** ნაკრებებზე ამიტომ მათი მიხედვით შედგენილი $\overline{x}_1\,\overline{x}_2\,\overline{x}_3\,x_4$, $x_1\,\overline{x}_2\,\overline{x}_3\,\overline{x}_3$, $x_1\,\overline{x}_2\,\overline{x}_3\,x_4$, $x_1\,x_2\,x_3\,\overline{x}_4$ მინიტერმები ($x_1+x_2+x_3+\overline{x}_4$, $\overline{x}_1+x_2+x_3+x_4$, $\overline{x}_1+x_2+x_3+\overline{x}_4$, $\overline{x}_1+\overline{x}_2+\overline{x}_3+x_4$ მაქსიტერმები) წარმოადგენს **ნეიტრალურ მინიტერმებს** (**მაქსიტერმებს**).

ცხრ. 4.10 არასრულად განსაზღვრული ლოგიკური ფუნქცია

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

კარნოს ბარათზე არასრულად განსაზღვრული ფუნქციის ნეიტრალური ნაკრებების შესაბამისი ფუნქციის მნიშვნელობები "~" ნიშნებითაა აღნიშნული. **4.10** ცხრილით განსაზღვრული ფუნქციისათვის შედგენილი კარნოს ბარათი **4.22,ა** ნახაზზეა ნაჩვენები. ამის მიხედვით გასაზღვრულ მინიმალურ დიზიუნქციურ ნორმალურ ფორმას ექნება შემდეგი სახე:

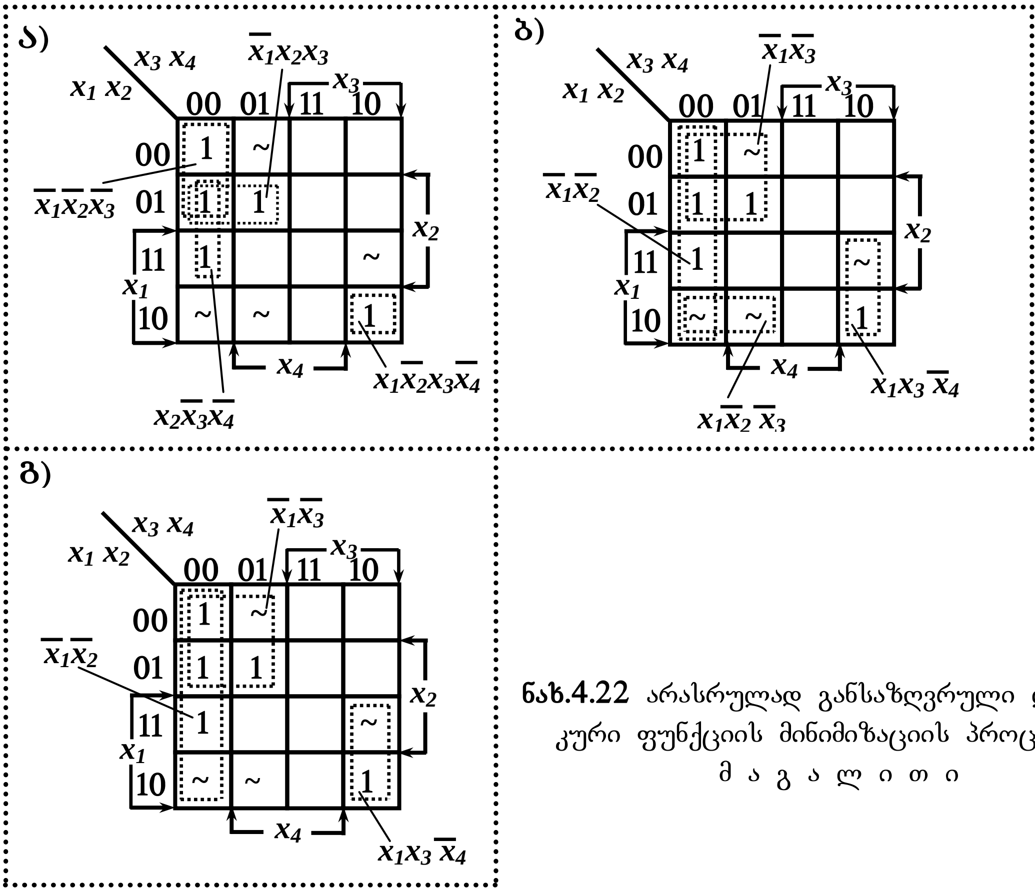
$$f(x_1x_2x_3\,x_4) = \overline{x}_1\,\overline{x}_2\,\overline{x}_3 + x_2\overline{x}_3\overline{x}_4 + \overline{x}_1x_2\overline{x}_3 + x_1\overline{x}_2\,x_3\,\overline{x}_4\,, \qquad (4.68)$$

რომლისთვისაც ჯამური რანგი **Q = 13.**

2 მოცემული ფუნქციის მარეალიზებელი დისკრეტული მოწყობილობის ფუნქციონირების მთელ პროცესში მის შესასვლელზე არასოდეს მიეწოდება შესასვლელი სიგნალების **0001, 1000, 1001** და **1110** ნაკრებები, რომლებისთვისაც ჭეშმარიტობის **4.10** ცხრილში ფუნქციის y მნიშვნელობა განსაზღვრული არ არის; აღნიშნულიდან გამომდინარე, დისკრეტული მოწყობილობის ფუნქციონირების ალგორითმი არ დაირღვევა, თუ ჭეშმარიტობის **4.10** ცხრილში ფუნქციის y მნიშვნელობას პირობითად **0**-ის ან **1**-ის ტოლად ჩავთვლით. ასეთ შემთხვევაში ნაწილობრივად

განსაზღვრული ლოგიკური ფუნქცია გარდაიქმნება სრულად განსაზღვრულ ლოგიკურ ფუნქციად.

ნეიტრალური მინიტერმებისათვის ფუნქციის მნიშვნელობის ნებისმიერი განსაზღვრის გზით სრულად განსაზღვრულ ლოგიკურ ფუნქციად არასრული ლოგიკური ფუნქციის გარდაქმნის წესს მოცემული ფუნქციის **განსაზღვრების ხელოვნურად დასრულების წესი** ვუწოდოთ.



ნახ.4.22 არასრულად განსაზღვრული ლოგიკური ფუნქციის მინიმიზაციის პროცესის მ ა გ ა ლ ი თ ი

გამოვიყენოთ არასრულად განსაზღვრული ლოგიკური ფუნქციის განსაზღვრების ხელოვნურად დასრულების მეთოდი და:

- ჩავთვალოთ, რომ **4.10** ცხრილში არსებული ყველა "~" ნიშანი ციფრ "**1**"-ის ეკვივალენტურია. ასეთ შემთხვევაში **4.22,ა** ნახაზზე ნაჩვენები კარნოს ბარათი გარდაიქმნება **4.22,ბ** ნახაზზე ნაჩვენებ ბარათად და ფუნქციის მინიმალურ დიზიუნქციურ ნორმალურ ფორმას ექნება შემდეგი სახე:

$$f(x_1x_2x_3\,x_4) = \overline{x_1}\,\overline{x_2} + \overline{x_1}\,\overline{x_3} + x_1\overline{x_2}\,\overline{x_3} + x_1x_3\overline{x_4}, \qquad \textbf{(4.69)}$$

რომლის ჯამური რანგი **Q=10,** ე.ი. (**4.68**) გამსახულებასთან შედარებით დაახლოებით **33%**-ით შემცირდა.

- ჩავთვალოთ, რომ **4.10** ცხრილში **1001** ნაკრების გასწვრივ არსებული "~" ნიშანი "**0**"-ის, ხოლო ყველა დანარჩენი "~" ნიშანი - "**1**"-ის ეკვივალენტურია; ასეთ შემთხვევაში ფუნქციის კარნოს ბარათს **4.22,გ** ნახაზზე ნაჩვენები სახე ექნება, საიდანაც მივიღებთ ფუნქციის შემდეგ მინიმალური დიზიუნქციურ ნორმალურ ფორმას:

$$f(x_1x_2x_3\,x_4) = \overline{x_1}\,\overline{x_2} + \overline{x_1}\,\overline{x_3} + x_1x_3\overline{x_4}, \qquad \textbf{(4.70)}$$

რომლის ჯამური რანგი **Q=7,** ე.ი. (**4.68**) გამსახულებასთან შედარებით დაახლოებით **47,2%**-ით შემცირდა.

(**4.70**) არის **4.20** ცხრილიდან მიღებულ მინიმალურ დიზიუნქციურ ფორმებს შორის ყველაზე მარტივი გამოსახულება; მის მიხედვით აგებულ დისკრეტულ მოწყობილობას ექნება დანარჩენი გამოსახულებების მიხედვით აგებულ ნებისმიერ დისკრეტულ მოწყობილობაზე უკეთესი ტექნიკური, ეკონომიკური და საიმედოობითი მაჩვენელები.

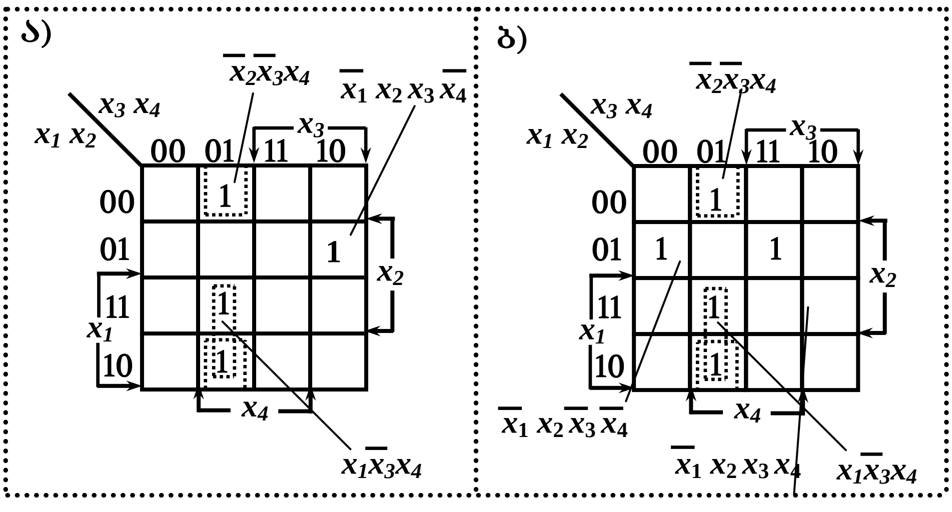
ზემოთ ჩვენ განვიხილავდით n რაოდენობის $x_1, x_2, \dots, x_n$ შესასვლელისა და ერთი y გამოსასვლელის მქონე დისკრეტულ მოწყობილობის მქონე დისკრეტულ მოწყობილობას, რომელიც ახდენს ლოგიკური $y = f(x_1, x_2, \dots, x_n)$ ფუნქციის რეალიზაციას. აღნიშნული მოწყობილობა წარმოადგენს n რაოდენობის შესასვლელისა და m რაოდენობის გამოსასვლელის მქონე (იხ. ნახ. **4.2**) დისკრეტული მოწყობილობის კერძო შემთხვევას, რომლისათვისაც $m = 1$. ახლა განვიხილოთ $m > 1$ შემთხვევა; ასეთი მოწყობილობის საშუალებით რეალიზდება ლოგიკურ ფუნქციათა შემდეგი სისტემა:

$$\left.\begin{aligned} y_1 &= f_1(x_1, x_2, \dots, x_n),\\ y_2 &= f_2(x_1, x_2, \dots, x_n),\\ &\;\;\vdots\\ y_m &= f_m(x_1, x_2, \dots, x_n). \end{aligned}\right\} \qquad (4.71)$$

დისკრეტული მოწყობილობის სტრუქტურის გასამარტივებლად, უპირველეს ყოვლისა, ინდივიდუალურად უნდა მოვახდინოთ (**4.71**) სისტემაში შემავალი თითოეული ფუნქციის მინიმიზირება. ამის შედეგად მიღებული მინიმიზირებული ფუნქციები თუ შეიცავს წევრების ერთნაირ ერთობლიობას, მაშინ მათი რეალიზებისათვის შეიძლება მოწყობილობის ერთი და იგივე კვანძები გამოვიყენოთ, რაც დამატებით გაამარტივებს მოწყობილობის სტრუქტურას. ამიტომ ლოგიკური ფუნქციების მინიმიზაციის დროს საჭიროა განისაზღვროს ზემოთ აღნიშნული ერთნაირი წევრების ერთობლიობა.

ფუნქციათა სისტემების მინიმიზაცია შეიძლება მოხდეს ამ სისტემაში შემავალი m რაოდენობის ფუნქციებისათვის აგებული იმპლიკანტური მატრიცას [**15;16**] ან ფუნქციების ურთიერთდაკავშირებით მიღებული რაოდენობის გამოსასვლელიანი ფუნქციის [**16**] გამოყენების საშუალებით. გავეცნოთ ამ უკანასკნელს. განვიხილოთ ფუნქციათა სისტემა, რომლისთვისაც $m = 2$:

$$\left.\begin{aligned} f_1 &= \bar{x}_1\bar{x}_2\bar{x}_3x_4 + x_1x_2\bar{x}_3x_4 + x_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1x_2x_3\bar{x}_4,\\ f_2 &= \bar{x}_1\bar{x}_2\bar{x}_3x_4 + x_1x_2\bar{x}_3x_4 + x_1\bar{x}_2\bar{x}_3x_4 + \bar{x}_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2x_3x_4. \end{aligned}\right\} \qquad (4.72)$$



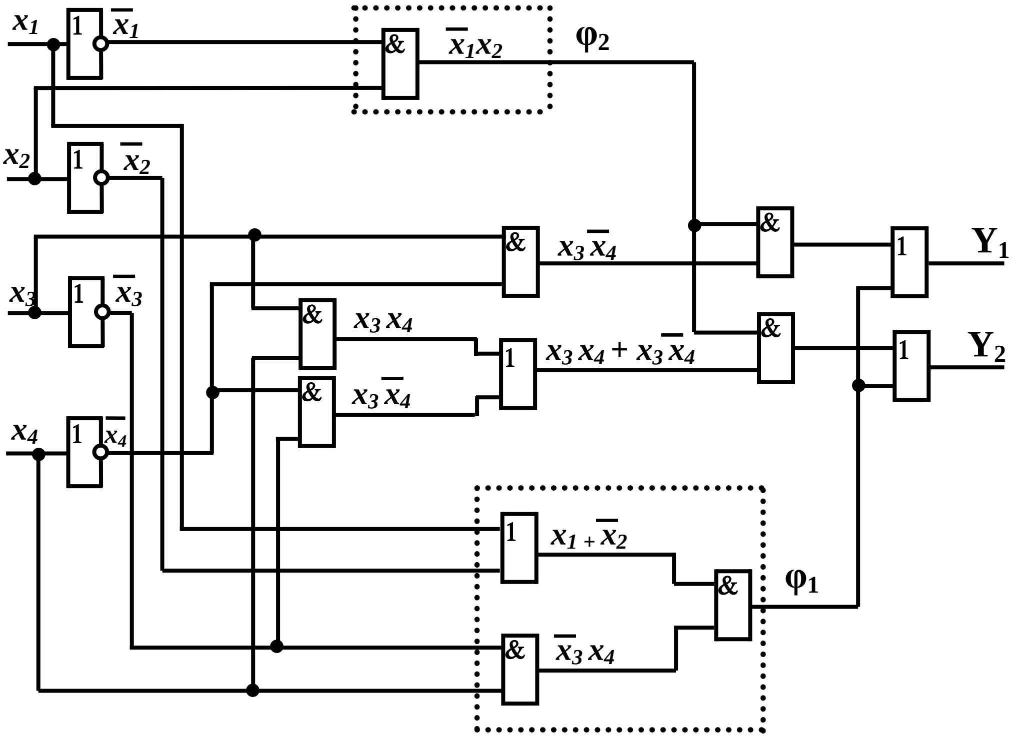
ნახ. 4.23. (4.72) სისტემაში შემავალი ლოგიკური ფუნქციების შესაბამისი კარნოს ბარათები

რომელიმე ცნობილი მეთოდის გამოყენებით მოვახდინოთ (4.72) სისტემაში შემავალი ლოგიკური f_1 და f_2 ფუნქციების მინიმიზაცია. **4.23,ა,ბ** ნახაზებზე მოყვანილია აღნიშნული ფუნქციების შესაბამისი კარნოს ბარათები, რომელთა საშუალებითაც მიიღება მათი მინი-მიზირებული f_1^* და f_2^* გამოსახულებები:

$$\left.\begin{aligned} f_1^* &= \overline{x_3}\, x_4\, (x_1 + \overline{x_2}) + \overline{x_1}\, x_2\, x_3\, \overline{x_4}\,, \\ f_2^* &= \overline{x_3}\, x_4\, (x_1 + \overline{x_2}) + \overline{x_1}\, x_2\, (x_3\, x_4 + \overline{x_3}\,\overline{x_4})\,. \end{aligned}\right\} \qquad (4.73)$$

სისტემაში შემავალი მინიმიზირებული $f_1^*, f_2^*, \bullet\bullet\bullet, f_m^*$ ფუნქციებისაგან ერთი Φ ფუნქციის მისაღებად საჭიროა f_i^*, $i = 1; 2; \ldots ; m$ ფუნქციაში შემავალი დიზიუნქციური წევრები აღვნიშნოთ $\mathbf{Y}_i$ **ასო-იარლიყით** და შემდეგ სისტემაში შემავალ ფუნქციათა ყველა წევრი ერთმანეთს დავუკავშიროთ დიზიუნქციის ნიშნებით. Φ ფუნქციაში არსებული $\mathbf{Y}_i$ ასო-იარლიყი გვიჩვენებს თუ რომელ f_i^* ფუნქციაში შედის მოცემული დიზიუნქციური წევრი. $\mathbf{Y}_i$ ასო-იარლიყით f_i^* ფუნქციის დიზიუნქციური წევრების აღნიშვნის ფაქტს თუ $f_i^*\,\mathbf{Y}_i$ გამოსახულებით ავღნიშნავთ, მაშინ მივიღებთ:

$$\Phi = f_1^*\, \mathrm{Y}_1 + f_2^*\, \mathrm{Y}_2 + \ldots + f_m^*\, \mathrm{Y}_m\,. \qquad (4.74)$$



ნახ.4.24. ლოგიკურ ფუნქციათა სისტემის მინიმიზაციის მაგალითი

ზემოთ განხილულ მაგალითში მიღებული (4.73) სისტემა მიიღებს შემდეგ სახეს:

$$\left.\begin{aligned} f_1^*\mathrm{Y}_1 &= \overline{x_3}\, x_4\, (x_1 + \overline{x_2}) + \overline{x_1}\, x_2\, x_3\, \overline{x_4}\,, \\ f_2^*\mathrm{Y}_2 &= \overline{x_3}\, x_4\, (x_1 + \overline{x_2}) + \overline{x_1}\, x_2\, (x_3\, x_4 + \overline{x_3}\,\overline{x_4})\,. \end{aligned}\right\} \qquad (4.75)$$

ამიტომ, (4.74) გამოსახულების თანახმად გვექნება:

$$\Phi = \overline{x_3}\, x_4\, (x_1 + \overline{x_2})\mathrm{Y}_1 + \overline{x_1}\, x_2\, x_3\, \overline{x_4}\, \mathrm{Y}_1 + \overline{x_3}\, x_4\, (x_1 + \overline{x_2})\, \mathrm{Y}_2 + $$
$$+ \overline{x_1}\, x_2\, (x_3\, x_4 + \overline{x_3}\,\overline{x_4})\, \mathrm{Y}_2 \qquad (4.76)$$

ერთიანი $\mathbf{\Phi}$ ფუნქციის შედგენის შემდეგ სისტემის მინიმიზაციის ამოცანა დაიყვანება მასში შემავალი $\boldsymbol{\varphi}\mathbf{Y_i}$ და $\boldsymbol{\varphi}\mathbf{Y_j}$ $\mathbf{(i;j \in \{1,2,\ldots,\mathit{m}\},\ i \neq j)}$ სახის დიზიუნქციური წევრების პოვნისა და $\boldsymbol{\varphi}\mathbf{(Y_i+Y_j)}$ სახით მათ წარმოდგენამდე, სადაც $\mathbf{(Y_i+Y_j)}$ გამოსახულება ნიშნავს, რომ $\boldsymbol{\varphi}$ ეკუთვნის როგორც $\boldsymbol{f}_1^*$, ასევე $\boldsymbol{f}_2^*$ ფუნქციას. ამასთანავე, **თუ დისკრეტული მოწყობილობის ასაგებად გამოიყენება კონტაქტური ელემენტები**, მაშინ საერთო $\boldsymbol{\varphi}$ ნაწილი შეიძლება შეიცავდეს ერთ ცვლადს (იგი გარკვეულ კონტაქტს შეესაბამება), ხოლო **თუ უკონტაქტო ელემენტები გამოიყენება**, მაშინ იგი უნდა შეიცავდეს ორზე არანაკლებ ცვლადს.

უკონტაქტო ელემენტების გამოყენების შემთხვევისათვის (**4.76**) გამოსახულება მიიღებს შემდეგ სახეს:

$$\mathbf{\Phi} = \overline{x_3}\, x_4\, (x_1 + \overline{x_2})\, (\mathbf{Y_1} + \mathbf{Y_2}) + \overline{x_1}\, x_2\, [\, x_3\, \overline{x_4}\, \mathbf{Y_1} + (\overline{x_3}\, \overline{x_4} + x_3\, x_4)\mathbf{Y_2}\,], \qquad (4.77)$$

ხოლო კონტაქტური ელემენტების გამოყენების შემთხვევისათვის – სახეს:

$$\mathbf{\Phi} = \overline{x_3}\, x_4\, (x_1 + \overline{x_2})\, (\mathbf{Y_1} + \mathbf{Y_2}) + \overline{x_1}\, x_2\, [\, x_4\, (x_3\mathbf{Y_1} + x_3\, \mathbf{Y_2}\,] + x_3\, x_4\, \mathbf{Y_2}\,]. \qquad (4.78)$$

(**4.78**) ფუნქციის მარეალიზებელი დისკრეტული მოწყობილობის სქემა **4.24** ნახაზზეა ნაჩვენები. მასზე პუნქტირული ხაზებით გამოსახული ოთკუთხედებით გამოყოფილია კვანძები, რომლებიც ახდენს როგორც $\boldsymbol{f}_1^*$, ასევე $\boldsymbol{f}_2^*$ ფუნქციაში შემავალი საერთო $\boldsymbol{\varphi}_1 = x_3\, x_4\, (x_1 + x_2)$ და $\boldsymbol{\varphi}_2 = x_1\, x_2$ წევრების რეალიზაციას

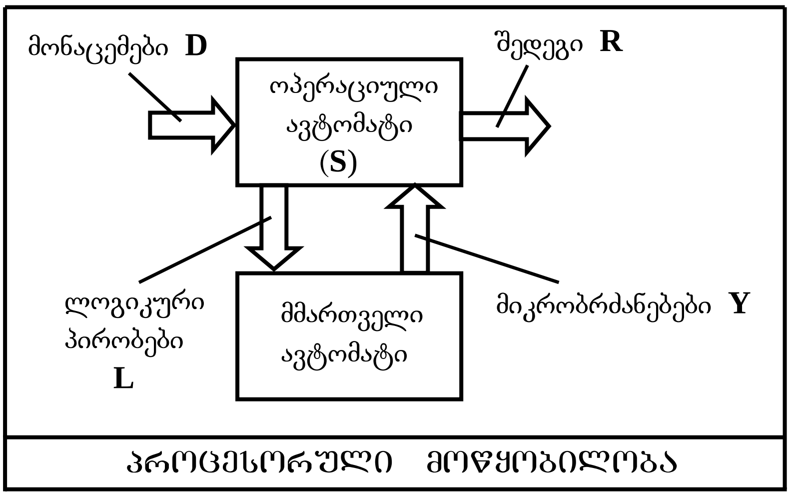
V თ ა ვ ი
კომპიუტერული სისტემების ასაგებად ლოგიკის ალგებრის გამოყენების საკითხები

“არა ვიქმ, ცოდნა რას მარგებს ფილოსოფოსთა ბრძნობისა”
შოთა რუსთაველი

5.1. პროცესორული მოწყობილობების ფუნქციონირების საკითხები

კომპიუტერულ სისტემებში ინფორმაციის დასამუშავებლად საჭირო ფუნქციებს პროცესორი ასრულებს. პროცესორის ასაგებად გამოიყენება **მიკროპროგრამული ლი მართვის პრინციპები,** რომლის თანახმადაც:

- ორობით კოდებზე (ე.წ. **სიტყვებზე**) პროცესორის მიერ შესრულებული თითოეული ოპერაცია რთული მოქმედებაა, რომელიც იყოფა **მიკროოპერაციებად** წოდებულ ელემენტარულ მოქმედებებად;
- მიკროოპერაციების შესრულების თანამიმდევრობის სამართავად გამოიყენება **ლოგიკური პირობები,** რომლებიც შესრულებული მიკროოპერაციების შემდეგ ლოგიკური ნულის ან ერთის სახით ასახავს პროცესორის მდგომარეობას;
- პროცესორში შესასრულებელი პროცესი აღიწერება მიკროოპერაციებისა და ლოგიკური პირობების ტერმინებში წარმოდგენილი ალგორითმის ფორმით, რომელსაც **მიკროპროგრამა** ეწოდება;
- მიკროპროგრამა პროცესორის ფუნქციის გამოსახვის ფორმაა, რომლის საფუძველზე განისაზღვრება პროცესორის სტრუქტურა და დროში მისი ფუნქციონირების წესი.



ნახ. 5.1 პროცესორის სტრუქტურა

სტრუქტურულ-ფუნქციონალური თვალსაზრისით პროცესორი იყოფა ოპერაციულ და მმართველ ავტომატებად (ნახ.5.1).

ოპერაციული ავტომატის დანიშნულებაა:

● **R** შედეგის მისაღებად დაამუშაოს შესასვლელი სიტყვების **D** სიმრავლე. დამუშავების პროცესს მართავს **Y** მიკრობრძანებები; თითოეული მიკრობრძანება ახდენს მიკროოპერაციათა სრულ $\{y_1; y_2, \ldots, y_N\}$ ნაკრებში შემავალი გარკვეული მიკროოპერაციის შესრულების ინიცირებას.

● მოახდინოს სამაუწყებლო სიგნალების **L** სიმრავლის ფორმირება; თითოეული სამაუწყებლო სიგნალი გარკვეული ლოგიკური პირობის სახით გამოიხატება;

● შეინახოს შესასვლელი სიტყვების **D**, აგრეთვე ინფორმაციის დამუშავების პროცესში წარმოშობილი ლოგიკური პირობების **L**, შინაგანი სიტყვების **S** და გამოსასვლელი სიტყვების **R** სიმრავლეები.

ზემოთ ჩამოთვლილი ფუნქციების შესასრულებლად ოპერაციული ავტომატი **მეხსიერებასა** და **კომბინაციურ დისკრეტულ მოწყობილობებს** შეიცავს; ამ უკანასკნელთა დანიშნულებაა შეასრულოს მიკოოპერაციები და გამოითვალოს ლოგიკური პირობების მნიშვნელობები.

3 **კომბინაციური დისკრეტული მოწყობილობა** წარმოადგენს $n \times m$ პოლუსას, (იხ.ნახ.4.2). მას მიეწოდება შესასვლელი $x_1, x_2, \ldots, x_n$ სიგნალები, რომელთა დამუშავების შედეგად წარმოიშვება გამოსასვლელი $y_1, y_2, \ldots, y_m$ სიგნალები. შესასვლელ და გამოსასვლელ სიგნალებს შორის კავშირს განსაზღვრავს რაოდენობის ლოგიკური ფუნქციებისაგან შემდგარი სისტემა:

$$\left.\begin{aligned} y_1 &= f_1(x_1, x_2, \ldots, x_n), \\ y_2 &= f_2(x_1, x_2, \ldots, x_n), \\ &\vdots \\ y_m &= f_m(x_1, x_2, \ldots, x_n). \end{aligned}\right\} \qquad (5.1)$$

ასეთი ზოგადი სახის სისტემები და მათი მარეალიზებელი მოწყობილობების ბლოკსქემა ჩვენ მეოთხე თავში განვიხილეთ. მოცემულ თავში კი შევისწავლით ფართოდ გავრცელებული ზოგიერთი კონკრეტული სახის კომბინაციური დისკრეტული მოწყობილობის სინთეზისა და ფუნქციონირების საკითხებს.

4 **მმართველი ავტომატი** (**იხ. ნახ.5.1**) განკუთვნილია **Y** სიმრავლეში შემავალი მმართველი სიგნალების გენერირებისათვის. ლოგიკური **L** პირობების შესაბამისი მმართველი სიგნალები მიკროპროგრამებში წერილობითი ფორმით არის ფიქსირებული. მიკროპროგრამების ერთობლიობის (პროგრამის) რეალიზებას ახდენს პროცესორი. აღნიშნული პროცესის შესრულების დროს პროცესორის შესასვლელებს მიეწოდება ამა თუ იმ მიკროპროგრამის შესაბამისი ოპერაციების კოდები; შესასვლელებს იმავდროულად ლოგიკური გარე პირობების შესაბამისი სიგნალებიც შეიძლება მიეწოდებოდეს. რაც შეეხება პროცესორის გამოსასვლელებს, იქ ფორმირდება გარეშე მოწყობილობების მართვისათვის საჭირო სიგნალები.

მმართველი ავტომატი შეიძლება ავაგოთ **აპარატურულად** ან **პროგრამულად**. პირველ შემთხვევაში ავტომატი წარმოადგენს გარკვეული ლოგიკური ელემენტებით აგებულ რეალურ დისკრეტულ მოწყობილობას, რომელიც გამოიმუშავებს მმართველ ბრძანებებს. მეორე შემთხვევაში ფიზიკური დისკრეტული მოწყობილობა არ არსებობს და მმართველი ბრძანებები მეხსიერების სპეციალურ მოწყობილობაში შენახული მიკროპროგრამის შესრულების პროცესში გამომუშავდება.

5 ზემოთ აღნიშნულის ნათლად გააზრებისათვის განვიხილოთ უმარტივესი ჰიპოტეტური პროცესორი, რომელიც ასრულებს **A•B** გამრავლების ოპერაციას, სადაც **A=0111**, ხოლო **B=0101.**

ოპერაციული ავტომატის აგების პრინციპის განსაზღვრისათვის ვისარგებლოთ ორობითი რიცხვების გამრავლების წესით, რომლის დროსაც **ნაწილობრივი ნამრავლების** ფორმირება იწყება მამრავლის უმცროსი თანრიგებიდან (**ნახ.5.2**).

×				0	1	1	1	– ს ა მ რ ა ვ ლ ი A
				0	1	0	1	– მ ა მ რ ა ვ ლ ი B
+				0	1	1	1	– 1-ლი ნაწილობრივი ნამრავლი
+			0	0	0	0		– მე-2 ნაწილობრივი ნამრავლი
+		0	1	1	1			– მე-3 ნაწილობრივი ნამრავლი
	0	0	0	0				– მე-4 ნაწილობრივი ნამრავლი
	0	1	0	0	0	1	1	– ნ ა მ რ ა ვ ლ ი

ნახ.5.2. მამრავლის უმცროსი თანრიგებიდან დაწყებული გამრავლ-ების წესის გამოყენება ორობითი რიცხვების გამრავლებისათვის

მიღებული ნაწილობრივი ნამრავლების შესაკრებად გამოიყენება **სუმატორად** წოდებული კომბინაციური მოწყობილობა. სუმატორის სტრუქტურის **სირთულე იზრდება** შესაკრები ოპერანდების რაოდენობის, აგრეთვე თითოეულ მათგანში შემავალი ბიტების რაოდენობის გაზრდით. აღნიშნულის თავიდან ასაცილებლად ნაწილობრივი ნამრავლების რაოდენობისაგან დამოუკიდებლად მათი შეკრებისათვის გამოიყენება ორი ოპერანდის შესაკრებად აგებული ერთი სუმატორი; იგი ნაწილობრივ ნამრავლებს დროში თანამიმდევრობით კრებს.

როგორც **5.2**. ნახაზიდან ჩანს, ორობითი რიცხვების გამრავლების თავისებურებაა ის, რომ ნაწილობრივ მამრავლებს შეუძლია მიიღოს მხოლოდ ორი მნიშვნელობა, კერძოდ:

- სამრავლ **A**–ს ტოლი მნიშვნელობა ან
- **0**-ის ტოლი მნიშვნელობა (იგი წარმოადგენს იმდენი **0**–საგან შემდგარ რიცხვს, რამდენ ბიტსაც შეიცავს ორობითი **A** რიცხვი).

ნაწილობრივი ნამრავლის სიდიდე განისაზღვრება **B** მამრავლის მიმდინარე თანრიგის მნიშვნელობით. ნაწილობრივი ნამრავლი თუ ნულის ტოლია, მაშინ შეკრების მიკროოპერაცია შეიძლება არ შევასრულოთ.

ზემოთ აღნიშნულიდან გამომდინარე, **ნაწილობრივ მამრავლად გამოიყენება** მხოლოდ ორობითი **A** რიცხვი და ამიტომ იგი მუდმივად უნდა შევინახოთ ამისათვის სპეციალურად გამოყოფილ **რეგისტრში**; **5.3** ნახაზზე იგი $\mathbf{RG_1}$ სახელითაა აღნიშნული.

B მამრავლის მიმდინარე თანრიგის აპარატურად განსაზღვრისათვის აღნიშნული მამრავლი საჭიროა მოვათავსოთ ე.წ. **ძვრის რეგისტრში** (შიგთავსის მარჯვნივ ან მარცხნივ გადაძვრის უნარის მქონე რეგისტრში); იგი **5.3** ნახაზზე $\mathbf{RG_2}$ სახელითაა აღნიშნული; მასში **B** მამრავლი ისე უნდა ჩავტვირთოთ, რომ გამოსასვლელი სიგნალი მამრავლის ყველაზე უმცროს თანრიგს შეესაბამებოდეს. **B** მამრავლის მომდევნო თანრიგის მნიშვნელობის განსასაზღვრავად შეკრების ყოველი მორიგი მიკროოპერაციის შესრულების შემდეგ $\mathbf{RG_2}$ რეგისტრში მოთავსებული **B** რიცხვი უმცროსი თანრიგის მხარეზე თითო თანრიგით უნდა გადაიძრას.

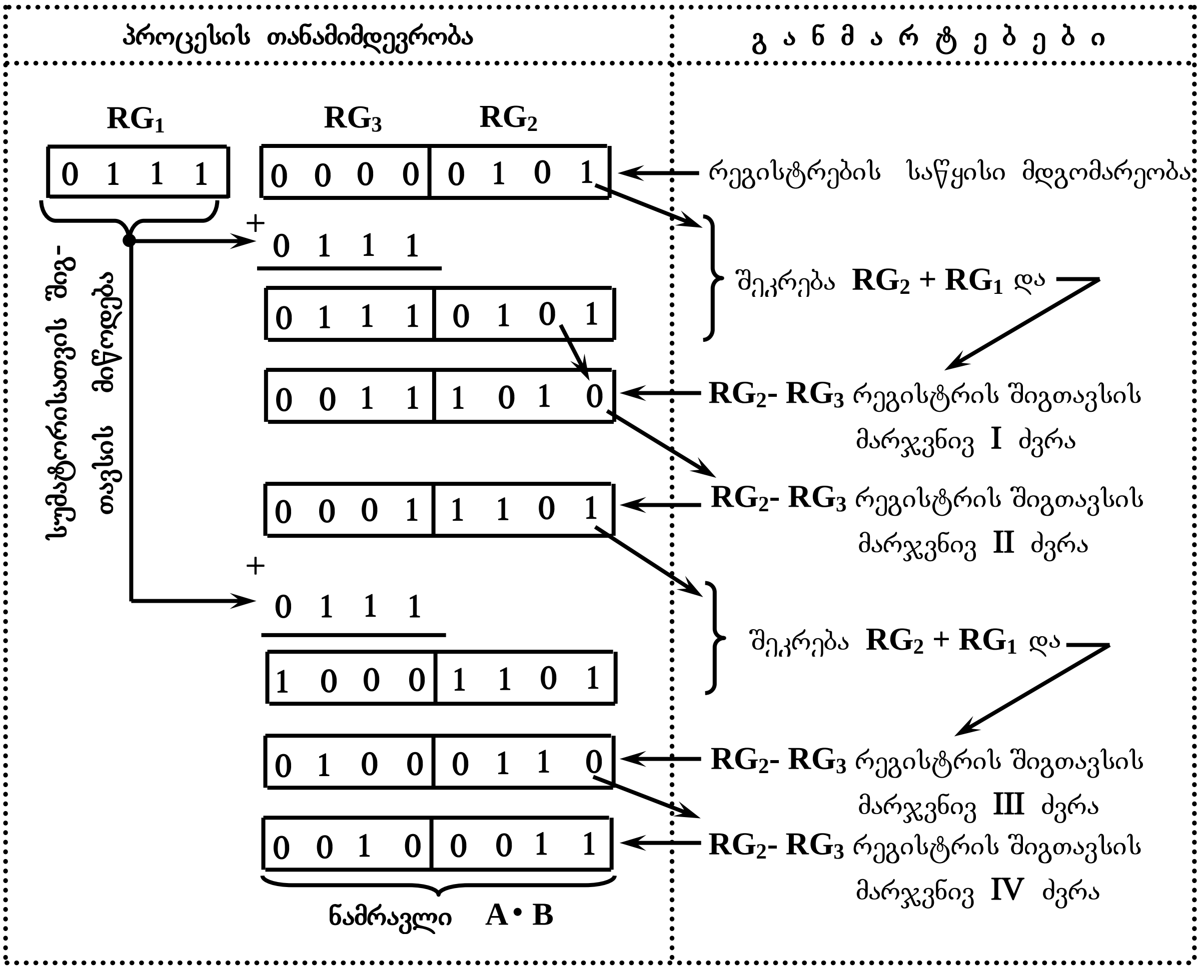
აუცილებელია შევინახოთ ნაწილობრივი მამრავლების თითოეული შეკრების შემდეგ მიღებული ჯამი. ამისათვის ოპერაციულ ავტომატში საჭიროა მესამე რეგისტრიც არსებობდეს. იგი **5.3** ნახაზზე $\mathbf{RG_3}$ სახელითაა აღნიშნული (იხ. ნახ.**5.3**). გამრავლების ოპერაციის დაწყების წინ $\mathbf{RG_3}$ რეგისტრში საჭიროა ნულები ჩავტვირთოთ.

გამრავლების პროცესი ასე მიმდინარეობს:

- $\mathbf{RG_1}$ რეგისტრის შიგთავსი ემატება ნაწილობრივ მამრავლად გამოყენებულ ორობით **A** რიცხვს, რომელიც $\mathbf{RG_3}$ რეგისტრშია მოთავსებული (ე.ი. იკრიბება $\mathbf{RG_3}$ და $\mathbf{RG_1}$ რეგისტრების შიგთავსები);

- მიღებული შედეგი თავსდება **RG**$_3$ რეგისტრში (ცვლის მასში არსებულ შიგთავსს);
- ხდება **RG**$_3$ რეგისტრის ახალი შიგთავსის ერთი თანრიგით მარჯვნივ დაძვრა, რადგან **B** მამრავლის თითოეული მორიგი თანრიგის წონა ორჯერ იზრდება (ნახ. **5.3**).

როგორც ვხედავთ დასახული გამრავლების ოპერაციის შესრულებისას საჭირო ხდება როგორც **RG**$_2$, ასევე **RG**$_3$ რეგისტრებში არსებული შიგთავსების მარჯვნივ დაძვრა; ამიტომ სასურველია ისინი ერთ რეგისტრად გავაერთიანოთ; აღნიშნული გაერთიანებით წარმოქმნილ რეგისტრს **შედგენილი რეგისტრი** ვუწოდოთ და პრობითად აღვნიშნოთ როგორც **RG**$_2$÷**RG**$_3$.



ნახ. 5.3. სამი რეგისტრისა და ერთი სუმატორით ორობითი რიცხვების გამრავლების ალგორითმის ილუსტრაცია

6 სამი რეგისტრისა და ერთი სუმატორის დახმარებით ორობითი რიცხვების გამრავლების მაგალითი **5.3** ნახაზზეა მოყვანილი. **A** სამრავლი მუდმივად **RG**$_1$ რეგისტრშია მოთავსებული; საწყის მომენტში **RG**$_3$ რეგისტრში **0000** რიცხვი, ხოლო **RG**$_2$ რეგისტრში – **B** მამრავლია მოთავსებული. **A** და **B** ოპერანდების უფროს თანრიგებში ციფრ **0**-ის არსებობა იმას გვიჩვენებს, რომ ორივე რიცხვი დადებითია.

RG$_3$ რეგისტრში ჩაიწერება:

- გამრავლების პროცესის მსვლელობის დროს – **ნაწილობრივი ნამრავლები**;
- გამრავლების დასასრულს – **საბოლოო ნამრავლი**.

RG$_2$ რეგისტრის **უმ**ცროს **თ**ანრიგში არსებული ციფრი პირობითად აღვნიშნოთ როგორც **უმთ(RG**$_2$**). უმთ(RG**$_2$**)**-ს მნიშვნელობა წარმოადგენს **L**$_1$ ლოგიკურ პირობას და გამრავლების პროცესში ხდება ამ პირობის განსაზღვრა; კერძოდ:

● თუ **უმთ(RG**$_2$**) = L**$_1$ **= 1,** მაშინ სრულდება შემდეგი **ორი ოპერაცია:**

1) შეიკრიბება **RG**$_3$ და **RG**$_2$ რეგისტრების შიგთავსები და მიღებული შიგთავსი მოთავსდება **RG**$_3$ რეგისტრში. აღნიშნული ოპერაცია პირობითად ასე ჩაიწერება:

$$RG_3 : = RG_3 + RG_3 \,. \qquad (5.2)$$

2) შედგენილი **RG**$_2$**÷RG**$_3$ რეგისტრის შიგთავსი მარჯვნივ ერთი თანრიგით (**R1**-ით) გადაიძვრება. აღნიშნული ოპერაცია პირობითად ასე ჩაიწერება:

$$RG_2 \div RG_3 : = R1\,(RG_2 \div RG_3). \qquad (5.3)$$

● თუ **უმთ(RG**$_2$**)=L**$_1$**=0,** მაშინ სრულდება **ერთადერთი ოპერაცია**, კერძოდ შედგენილი **RG**$_2$**÷RG**$_3$ რეგისტრის შიგთავსი მარჯვნივ ერთი თანრიგით გადაიძვრება. აღნიშნული ოპერაცია პირობითად **(5.3)** სახით ჩაიწერება.

5.3 ნახაზიდან ჩანს, რომ პროცესს აქვს ციკლური ხასიათი. ციკლების ***n*** რაოდენობა მამრავლის თანრიგების რაოდენობის ტოლია (მოცემულ შემთხვევაში ***n* = 4**). ამიტომ პროცესორის სქემური რეალიზაციის დროს ოპერაციის დამთავრების მომენტის ავტომატურად დაფიქსირებისათვის მიზანშეწონილია გამოვიყენოთ ციკლის გამეორების რაოდენობის მაჩვენებელი რიცხვიდან **გამოკლების ოპერაციის შემსრულებელი მთვლელი**; მას **გ**ამომკლები **მთ**ვლელი ვუწოდოთ და პირობითად **გმთ** აბრევიატურით აღვნიშნოთ. საწყის მდგომარეობაში **გმთ**-ში ჩაიტვირთება ციკლის რაოდენობის მაჩვენებელი ***n*** რიცხვი (ჩვენს შემთხვევაში n=4($100_{(2)}$)). თითოეული ციკლის დამთავრებისას მთვლელის შიგთავსი **1**-ით შემცირდება; მეოთხე ციკლის დამთავრების შემდეგ მთვლელი აღმოჩნდება ცარიელი (მასში ჩაწერილი იქნება რიცხვი **000**). მთვლელის გამოსასვლელს ლოგიკურ **ან-არა** ელემენტს თუ მივუერთებთ და ამ უკანასკნელის გამოსასვლელ სიგნალს ლოგიკურ **L**$_2$ პირობად გამოვიყენებთ, მაშინ **L**$_2$**=0** იქნება მეოთხე ციკლის დასრულებისა და გამრავლების ოპერაციის დამთავრების ნიშანი.

7 ორობითი რიცხვების მამრავლებელი ავტომატის სტრუქტურული სქემა **5.4** ნახაზზეა ნაჩვენები. შედგენილი **RG**$_2$**÷RG**$_3$ რეგისტრის მისაღებად **RG**$_3$ რეგისტრის უმცროსი თანრიგის ტრიგერის გამოსასვლელი **RG**$_2$ რეგისტრის უფროსი თანრიგის შესასვლელთან არის მიერთებული.

გამრავლების ოპერაციის შესასრულებლად **მმართველი ავტომატი** (იხ. ნახ. **5.1**) თანამიმდევრულად გამოიმუშავებს მმართველ სიგნალებს, რომელთა ფუნქციები ასეთია:

● მმართველი **y**$_1$ სიგნალის დახმარებით **RG**$_3$ რეგისტრი ნულოვან მდგომარეობაში გადაიყვანება;

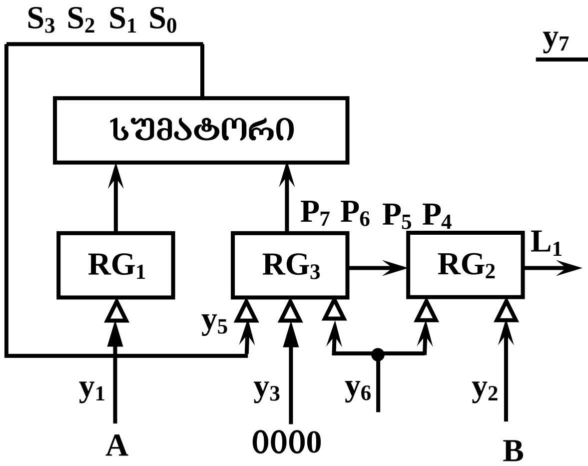
● მმართველი **y**$_2$ სიგნალის დახმარებით **გმთ** მთვლელში შეიტანება n=4($100_{(2)}$) რიცხვი;

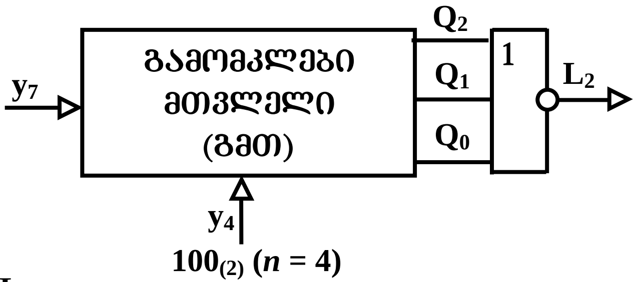
● მმართველი **y**$_3$ სიგნალი გამოიყენება **RG**$_3$ და **RG**$_1$ რეგისტრების შიგთავსების შეკრების შედეგად სუმატორის შესასვლელზე წარმოქმნილი ორობითი რიცხვის მისაწოდებლად **RG**$_3$ რეგისტრისათვის;

● მმართველი **y**$_4$ სიგნალი გამოიყენება შედგენილი **RG**$_2$**÷RG**$_3$ რეგისტრის შიგთავსის მარჯვნივ ერთი თანრიგით დაძვრისათვის;

● მმართველი **y**$_5$ სიგნალი **გმთ** მთვლელის შიგთავსს ერთი ერთეულით ამცირებს;

ივარაუდება, რომ **A** და **B** ოპერანდები უკვე ჩატვირთულია რეგისტრებში. მმართველი **y**$_1$ – **y**$_5$ სიგნალები იგივდება მიკროოპერაციებთან.





შენიშვნები:

y_6 – რეგიტრთა წყვილის შიგთავსის მარჯვნივ ერთი თანრიგით ძვრა;

y_7 – მთვლელის შიგთავსის შემცირება ერთი ერეთეულით.

ნახ. 5.4. ორობითი რიცხვების გამამრავლებელი ოპერაციული ავტომატის სტრუქტურული სქემა

8

დროით ერთ ინტერვალში (ტაქტში) შესრულებად მიკროოპერაციებს **მიკრობრძანება** ეწოდება. მოვახდინოთ პროცესორული მოწყობილობის მიერ შესასრულებელი მიკრობრძანებების ფორმირება;

- $\mathbf{RG_3}$ რეგისტრისა და **გმთ** მთვლელის ჩატვირთვისათვის გამოყენებული y_1; y_2 მიკროოპერაციები შეიძლება ერთდროულად შესრულდეს, ამიტომ ისინი შეიძლება ერთიან $\mathbf{Y_1}$ მიკრობრძანებად გავაერთიანოთ და აღვნიშნოთ როგორც $\mathbf{Y_1: = \{ y_1; y_2\}}$;
- $\mathbf{RG_3}$ და $\mathbf{RG_1}$ რეგისტრების შიგთავსების $\mathbf{RG_3 + RG_1}$ შეკრების y_3 მიკროოპერაცია დამოუკიდებელ $\mathbf{Y_2}$ მიკრობრძანებად გამოვყოთ და აღვნიშნოთ როგორც $\mathbf{Y_2: = y_3}$;
- რეგისტთა $\mathbf{RG_2 \div RG_3}$ წყვილის მარჯვნივ ერთი თანრიგით დაძვრის y_4 და **გმთ** მთვლელის შიგთავსის ერთი ერთეულით შემცირების y_5 მიკროოპერაცია გავაერთიანოთ ერთ $\mathbf{Y_3}$ მიკრობრძანებად და აღვნიშნოთ როგორც $\mathbf{Y_3: = \{ y_4; y_5\}}$;

ოპერაციულ ავტომატში ფორმირდება შემდეგი **ლოგიკური პირობები**, რომლებსაც ზოგჯერ **ალმებს** ან **ნიშნებსაც** უწოდებენ:

- $\mathbf{X_1}$, რომელიც გამოხატავს $\mathbf{RG_2}$ რეგისტრის შიგთავსის უმცროსი თანრიგის მნიშვნელობას; $\mathbf{X_1=1}$ ნიშნავს, რომ $\mathbf{RG_2}$ რეგისტრის შიგთავსის უმცროსი თანრიგის მნიშვნელობა **1**-ის ტოლია, რაც ნიშნავს, რომ ჯერ სრულდება ზემოთ აღწერილი შეკრების ოპერაციის, ხოლო შემდეგ ძვრის ოპერაციები. $\mathbf{X_1=0}$–ის დროს $\mathbf{RG_2}$ რეგისტრის შიგთავსის უმცროსი თანრიგის მნიშვნელობა **1**-ის ტოლია, და ამიტომ მხოლოდ ძვრის ოპერაცია სრულდება;
- $\mathbf{X_2}$, რომელიც წარმოადგენს **გმთ** მთვლელის შიგთავსის ნულზე ტოლობის შემოწმების შედეგს; $\mathbf{X_2=1}$ ნიშნავს, რომ მთვლელი ცარიელია (**გმთ =000**) და გამრავლების ოპერაცია მთავრდება $\mathbf{X_2=0}$–ის დროს იწყება გამრავლების ოპერაციის მორიგი ციკლი;

9

როგორც ვხედავთ, ჩვენ მიერ განხილული უმარტივესი პროცესორის ასაგებად საჭიროა სუმატორის, რეგისტებისა და მთვლელების გამოყენება. ისინი წარმოადგენს კომბინაციურ დისკრეტულ მოწყობილობებს, რომელთა აგების მათემატიკურ საფუძველს **ლოგიკის ალგებრა** წარმოადგენს. აღნიშნული ალგებრა საშუალებას გვაძლევს დისკრეტული მოწყობილობების სინეზისათვის (ასაგებად) გამოვიყენოთ ფორმალური მეთოდები, რომლებიც მაქსიმალურად ამარტივებს აგების პროცესს

და, საშუალებას იძლევა მოწყობილობის სტრუქტურაში გარკვეული სიჭარბის შეტანის გზით უზრუნველყოფილი იქნეს მათი სათანადო საიმედოობითი მახასიათებლები.

ქვემოთ განვიხილავთ ლოგიკის ალგებრის გამოყენებით ზოგიერთი ტიპური კომბინაციური დისკრეტული მოწყობილობების აგების (სინთეზის) მაგალითებს. აღნიშნული მოწყობილობები შეიძლება აგებული იყოს როგორც **აპარატურულად** ასევე **პროგრამულად.**

5.2. ორობითი სუმატორი

ინფორმაციის დასამუშავებლად კომპიუტერს უნდა ჰქონდეს რიცხვით მონაცემებზე ძირითადი არითმეტიკული და ლოგიკური ოპერაციების შემსრულებელი მოწყობილობა. ასეთ მოწყობილობას **არითმეტიკულ-ლოგიკური მოწყობილობა** (შემოკლებით **ალმ**) ეწოდება. **ალმ**–ის საფუძველს წარმოადგენს ორი მთელი რიცხვის შეკრების არითმეტიკული ოპერაციის მარეალიზებელი **ორობითი სუმატორი**.

განვიხილოთ n თანრიგიანი ორი ორობითი რიცხვი:

$$A = a_{n-1}\, a_{n-2} \cdots a_1 a_0\,,\ B = b_{n-1}\, b_{n-2} \cdots b_1\, b_0\,, \qquad (5.4)$$

სადაც $a_i\,;\, b_i \in \{0;1\}$.

მოცემული რიცხვების შეკრებას ვიწყებთ ნულოვან თანრიგში არსებული a_0 და b_0 ციფრების შეკრებით, რის შედეგადაც ვიღებთ ორობითი $S = A + B$ რიცხვის ნულოვან თანრიგში არსებულ s_0 ციფრსა და ერთით მაღალ (პირველ) თანრიგში გადასატან $p_1 \in \{0;1\}$ ციფრს (ნახ.5.5,ა). შემდეგ ვკრებთ პირველ თანრიგში არსებულ a_0; b_0 და ნულოვანი თანრიგიდან გადმოსულ p_1 ციფრებს; ვიღებთ S რიცხვის პირველ თანრიგში არსებულ s_1 ციფრსა და ერთით მაღალ (მეორე) თანრიგში გადასატან p_2 ციფრს. ასე იკრიბება ნებისმიერ $i \in \{0;1;\ ...\ ;n\}$ თანრიგში არსებული ციფრები. **5.5,ბ** ნახაზზე მოყვანილ ცხრილში განსაზღვრულია a_i; b_i; p_i ციფრების ყველა შესაძლო მნიშვნელობებისათვის s_i და p_{i+1} ციფრების ყველა სათანადო მნიშვნელობა.

ა)

a_0	b_0	s_0	p_1
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

ბ)

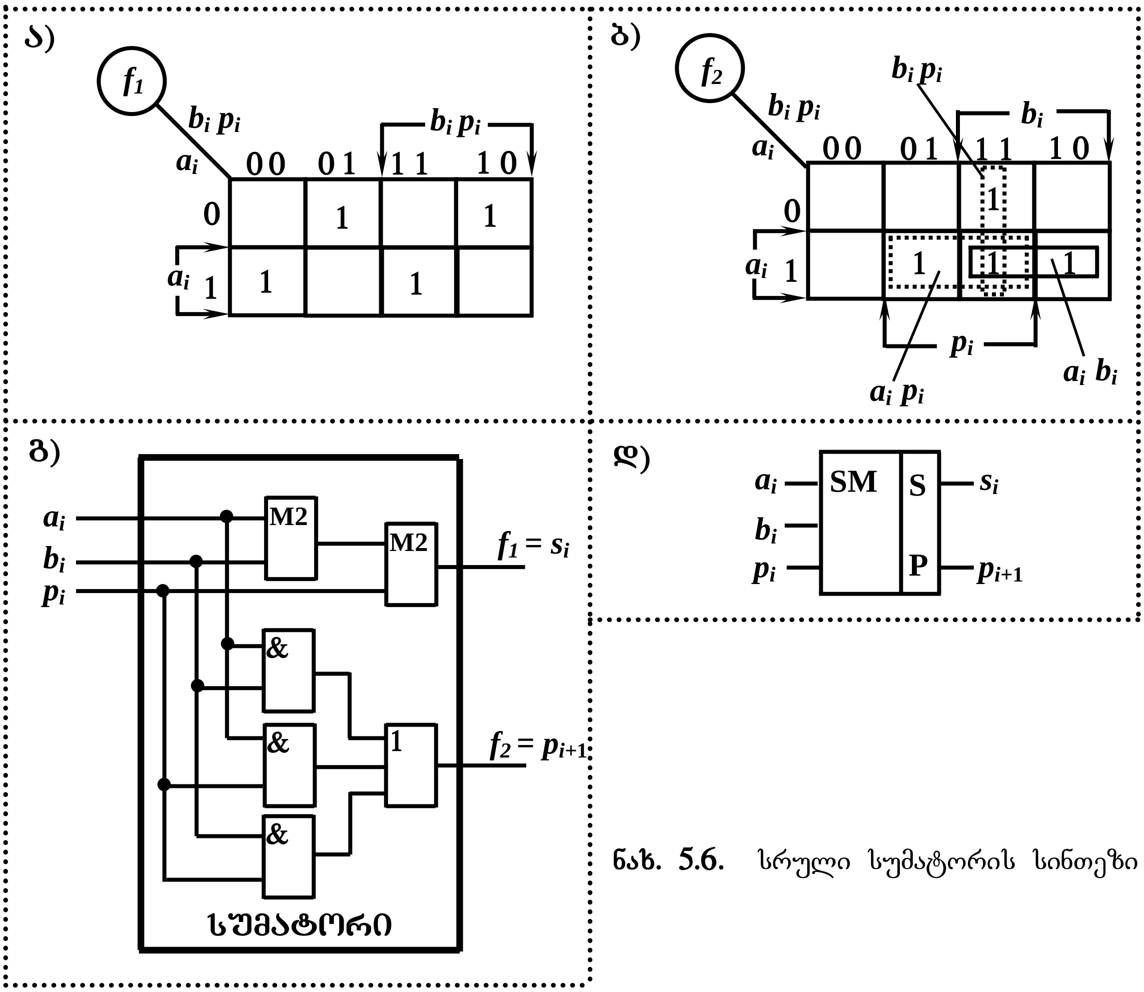
a_i	b_i	p_i	$s_i(f_1)$	$p_{i+1}(f_2)$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	0	0
1	0	1	1	1
1	1	0	0	1
1	1	1	1	1

ნახ. 5.5. ორობითი რიცხვების თანრიგების შეკრება

ნახ.5.5,ბ-ზე მოყვანილ ცხრილში თუ s_i-ს აღვნიშნავთ f_1-ით, ხოლო p_{i+1}-ს – f_2-ით, მაშინ იგი გადაიქცევა ლოგიკური f_1 და f_2 ფუნქციების **ჭეშმარიტობის ცხრილად,** საიდანაც შეგვიძლია განვსაზღვროთ ამ ფუნქციების **დიზიუნქციური სრულყოფილი ნორმალური ფორმები:**

$$\left.\begin{aligned} f_1 &= \bar{a}_i \bar{b}_i\, p_i + \bar{a}_i\, b_i\, \bar{p}_i + a_i \bar{b}_i\, \bar{p}_i + a_i\, b_i\, p_i \\ f_2 &= \bar{a}_i\, b_i\, p_i + a_i \bar{b}_i\, p_i + a_i\, b_i\, \bar{p}_i + a_i\, b_i\, p_i \end{aligned}\right\}, \qquad (5.5)$$

ლოგიკური ფუნქციების (5.5) სისტემის მარეალიზებელი კომბინაციური დისკრეტული მოწყობილობა ახდენს ორი ორობითი რიცხვის i–ურ თანრიგში არსებული ციფრების შეკრებას. მას უწოდებენ **სრულ ორობით სუმატორს** (ზოგიერთ ლიტერატურაში მას **ერთთანრიგიანი ორობითი სუმატორის** სახელითაც მოიხსენიებენ). მოვახდინოთ სრული ორობითი სუმატორის სინთეზი (**ნახ.5.6**).



ნახ. 5.6. სრული სუმატორის სინთეზი

ზემოთ აღნიშნული ამოცანის გადასაწყვეტად, უპირველეს ყოვლისა, უნდა მოვახდი-ოთ (5.5) სისტემაში შემავალი ლოგიკური ფუნქციების მინიმიზირება. ამ მიზნით გამო-ვიყენოთ კარნოს ბარათები. ლოგიკური f_1 ფუნქციისათვის აგებული კარნოს ბარათიდან (ნახ. 5.6,ა) ჩანს, რომ აღნიშნული ფუნქციის მინიმიზირება შეუძლებელია (d-კუბების არარსებობის გამო). სამაგიეროდ იგი ფაქტობრივად წარმოადგენს ლოგიკური a_i; b_i; p_i ცვლადების არაერთმნიშვნელიანობის, ანუ 2-ის მოდულით შეკრების ფუნქციას, ე.ი. $f_1 = a_i \oplus b_i \oplus p_i$; უკა-ნასკნელი გამოსახულება ანალიზურადაც მიიღება:

$$f_1 = \bar{a}_i \bar{b}_i p_i + \bar{a}_i b_i \bar{p}_i + a_i \bar{b}_i \bar{p}_i + a_i b_i p_i = \bar{a}_i (\bar{b}_i p_i + b_i \bar{p}_i) + a_i (\bar{b}_i \bar{p}_i + b_i p_i) =$$
$$= \bar{a}_i (b_i \oplus p_i) + a_i \overline{(b_i \oplus p_i)} = a_i \oplus (b_i \oplus p_i) = a_i \oplus b_i \oplus p_i ; \qquad (5.6)$$

5.6,ბ ნახაზზე მოყვანილი კარნოს ბარათის თანახმად f_1 ფუნქციის მინიმალურ დიზი-უნქციურ ნორმალურ ფორმას აქვს შემდეგი სახე:

$$f_2 = a_i b_i + a_i\ p_i + b_i p_i\ , \qquad (5.7)$$

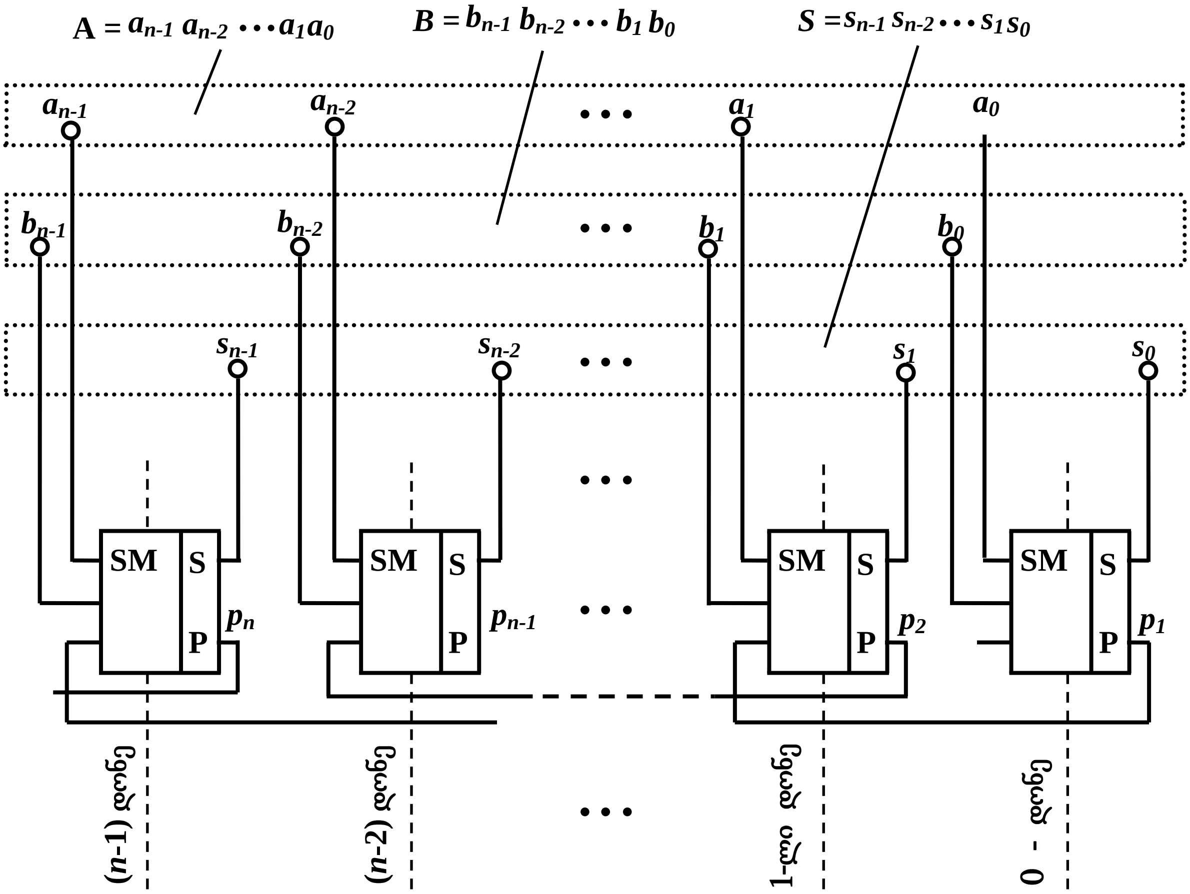
(5.6) და (5.7) გამოსახულებების გათვალისწინებით (5.5) სისტემა მიიღებს შემდეგ სახეს:

$$\left.\begin{aligned} f_1 &= a_i \oplus b_i \oplus p_i ; \\ f_2 &= a_i b_i + a_i\ p_i + b_i p_i\ , \end{aligned}\right\} \qquad (5.8)$$

რომლის მიხედვითაც აგებული **სრული სუმატორის** სტრუქტურული სქემა 5.6,გ ნახაზზე, ხოლო მისი პირობითი გამოსახულება - 5.6,დ ნახაზზეა ნაჩვენები.

2

n თანრიგიანი ორი ორობითი რიცხვის შესაკრებად შეიძლება გამოვიყენოთ ერთი სრული სუმატორი ან კასკადურად შეერთებული ***n*** რაოდენობის სრული სუმატორი;



ნახ. 5.7. პარალელური მოქმედების ორობითი სუმატორი

პირველ შემთხვევაში სრული სუმატორის შესასვლელზე თანდათანობით მიეწოდება **A** და **B** რიცხვების $i = 0;1; \dots ;n$ თანრიგებში არსებული ციფრები და სპეციალური სქემის საშუალებით ფიქსირდება მის **S** გამოსასვლელზე წარმოშობილი ციფრები. ასეთ სუმატორს **მიმდევრობითი მოქმედების ორობითი სუმატორი** ეწოდება. მისი ღირსებაა სტრუქტურული სიმარტივე, ხოლო ნაკლი კი – დაბალი სწრაფმოქმედება.

n რაოდენობის სრული სუმატორების კასკადურად შეერთების შედეგად მიღებული ორობითი სუმატორის სტრუქტურული სქემა **5.7** ნახაზზეა მოყვანილი. მის შესასვლელს ერთდროულად მიეწოდება შესაკრები **A** და **B** რიცხვები. კერძოდ, ამ რიცხვების i–ურ თანრიგში არსებული a_i და b_i ციფრები მიეწოდება i–ურ სრულ სუმატორს. ასეთ სუმატორს **პარალელური მოქმედების ორობითი სუმატორი** ეწოდება. მართალია, მისი სტრუქტურა მიმდევრობითი მოქმედების სუმატორთან შედარებით რთულია, მაგრამ ხასიათდება შედარებით მაღალი სწრაფმოქმედებით; ამ უკანასკნელის კიდევ უფრო მეტად გაზრდა შესაძლებელია სუმატორის სტრუქტურის შემდგომი გართულების, კერძოდ, მასში დაჩქარებული გადატანების დამატებითი ბლოკის გამოყენების გზით. განვიხილოთ თუ როგორ ხდება ეს.

5.7 ნახაზზე ნაჩვენები ორობითი სუმატორი შედგება ***n*** რაოდენობის ლოგიკური დონისაგან. თითოეულ დონეზე გადატანის p_i იმპულსი წარმოიქმნება წინა დონეზე გადატანის p_{i-1} იმპულსის წარმოქმნის შემდეგ, ე.ი. იგი წარმოადგენს **მიმდევრობით გადატანიან**

სუმატორს [12]. გადატანის p_i იმპულსის ფორმირებისათვის თუ τ_i დროა საჭირო, მაშინ გადატანის ყველა იმპულსის ფორმირებისათვის საჭირო მაქსიმალური დრო $n\tau_i$ –ს ტოლი იქნება. შეკრების პროცესის დასაჩქარებლად საჭიროა შემცირდეს ლოგიკურ დონეთა n რაოდენობა. საუკეთესო შემთხვევაში დროის გარკვეულ t_1 მომენტში უნდა წარმოიქმნას გადატანის ყველა p_i იმპულსი, რომელიც დროის მომდევნო t_2 მომენტში ერთდროულად მიეწოდება ყველა სრულ სუმატორს. ასეთ სუმატორს **პარალელურ გადატანიანი სუმატორი** ეწოდება და $n = 2$ დონისაგან შედგება. **პირველი დონე** შედგება დამატებითი გამოსასვლელი სიგნალების მაფორმირებელი n რაოდენობის სრული სუმატორებისაგან. დამატებითი სიგნალები დროის t_1 მომენტში ერთდროულად წარმოიშვება და მიეწოდება სუმატორის მეორე დონეში არსებულ დაჩქარებული გადატანების ბლოკს. ეს ბლოკი დროის მომდევნო t_2 მომენტში წარმოქმნის ყველა $p_1; p_2; \dots, p_n$ იმპულსს და მიაწვდის პირველ დონეში არსებულ სრულ სუმატორებს, რის შემდეგაც ეს უკანასკნელები აჯამებს **A** და **B** რიცხვებს.

5.3. ორობით-ათობითი სუმატორები

განვიხილოთ ორობით-ათობითი რიცხვები (იხ. პარაგრაფი 3.2), რომლებშიც ათობითი ციფრები **8421** კოდის შესაბამისი (იხ. ნახ. 3.4) კოდური სიტყვებით (ტეტრადებით) არის წარმოდგენილი (იხ. პარაგრაფი 3.3).

ათობით-ორობითი რიცხვების შეკრებისას მარჯვნიდან მარცხნივ თანამიმდევრულად იკრიბება მათში არსებული ტეტრადები; ორი ტეტრადის შეკრების შედეგად წარმოშობილი გადატანის ერთეული გადაიტანება მომდევნო ტეტრადაში და იგი ამ ტეტრადის მნიშვნელობას 10 ერთეულით ზრდის.

ორი ტეტრადის შეკრების შედეგად მიღებული ჯამი Σ სიმბოლოთი აღვნიშნოთ. თითოეული ტეტრადა გამოხატავს გარკვეულ ათობით ციფრს; ორი ასეთი ტეტრადის შეკრებით მიღებული ჯამის მაქსიმალური Σ_{max} მნიშვნელობა მასში წინა ტეტრადიდან გადატანის არსებობის გათვალისწინებით 19-ის ტოლია ($\Sigma_{max} = 9+9+1=19$). **5.1** ცხრილში Σ-ს ყველა შესაძლო მნიშვნელობა **8421** კოდის კოდური სიტყვებითაა წარმოდგენილი.

გადატანის გათვალისწინებით ორი ერთთანრიგიანი რიცხვის შეკრების დროს მცდარი ($\Sigma \geq 0$) შედეგი მიიღება, თუ ქვემოთ მოყვანილი სამი პირობიდან სრულდება ერთ-ერთი:

$$c_4 = 1; \;\; s_3 = s_2 = 1; \;\; s_3 = s_1 = 1; \qquad (5.9)$$

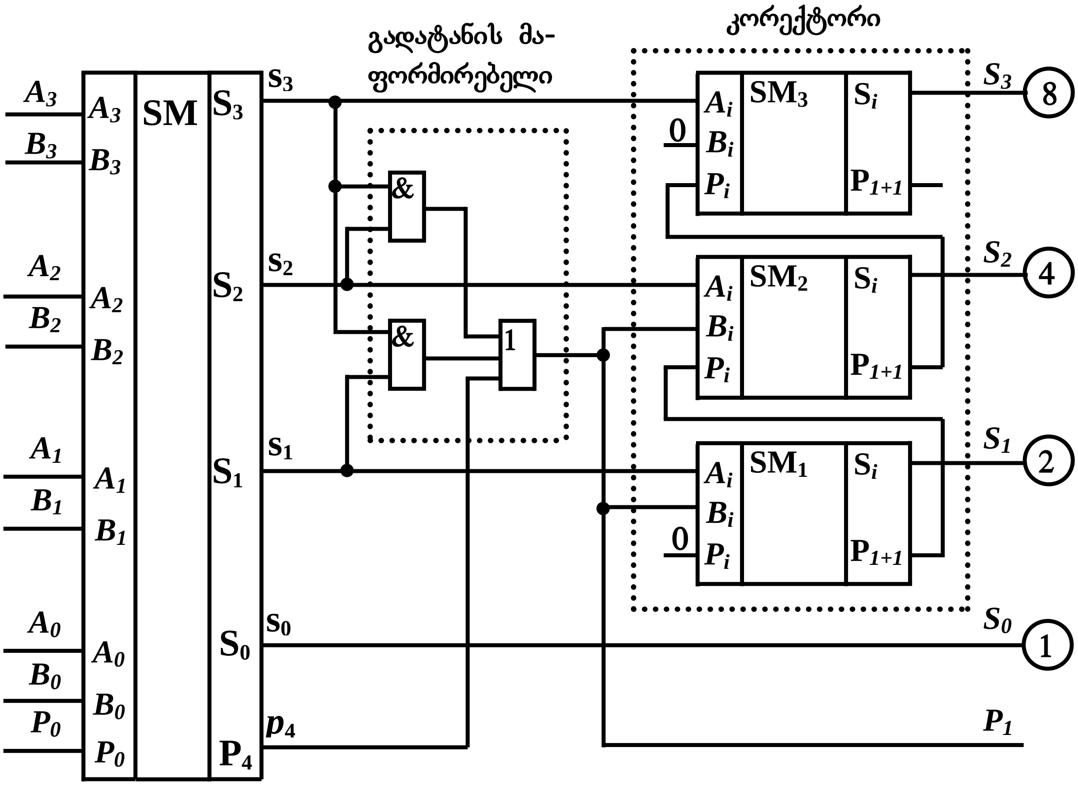
ამ პირობების არსებობის მაგალითები **5.1** ცხრილში სხვადასხვა ტონალობის მუქი უჯრედებით არის გამოყოფილი.

(5.9) პირობის არსებობისას საჭიროა ჩავატაროთ კორექცია, რაც მიღებული შედეგისათვის ორობითი **0110** ($6_{(10)}$) რიცხვის მიმატებას გულისხმობს.

(5.9) პირობები შეიძლება ერთ პირობად გავაერთიანოთ და გადატანის ფუნქციის გამომსახველი **ლოგიკური ფუნქციის დიზიუნქციური ნორმალური ფორმის** სახით ჩავწეროთ:

$$P_1 = p_4 + s_3 s_2 + s_3 s_1 . \qquad (5.10)$$

ორობით-ათობითი სუმატორის სქემა, რომელიც აჯამებს შესასვლელ **A** და **B** ოპერანდებს და ახდენს მიღებული შედეგის კორექციას, **5.8** ნახაზზეა ნაჩვენები. იგი შეიცავს:



ნახ. 5.8. კორექციის უნარის მქონე ორობით-ათობითი სუმატორი

● **4-თანრიგიან ორობით სუმატორს**, რომლის შესასვლელს მიეწოდება $A_3A_2A_1A_0$ და $B_3B_2B_1B_0$ ტეტრადების სახის მქონე ერთთანრიგიანი ათობითი რიცხვები და P_4 გადატანა, ხოლო გამოსასვლელზე ფორმირდება 4-თანრიგიანი $S_3S_2S_1S_0$ ჯამი და შემდეგ ტეტრადა-ში გადასატანი p_4 ციფრი;

● სტრუქტურული (5.10) ფორმულის მიხედვით აგებულ **გადატანის მაფორმირებელს**;

● სამი ერთთანრიგიანი ორობითი სუმატორისაგან აგებულ **კორექტორს**. რადგან ჯამის კორექციის დროს ორობით $S_3S_2S_1S_0$ რიცხვს ემატება ორობითი 0110 ($6_{(10)}$) რიცხვი; მისი უმცროსი თანრიგი 0-ის ტოლია, ამიტომ უმცროსი S_0 თანრიგის ერთთანრიგიანი ორობითი სუმატორი არ გამოიყენება (იხ.ნახ. 5.8). მოცემულ კორექტორში;

- SM_3, SM_2, SM_1 სუმატორების **A** შესასასვლელს მიეწოდება შესაბამისად s_3, s_2, s_1 სიგნალები;

- SM_2, SM_1 სუმატორების **B** შესასვლელებს მიეწოდება გადატანის მაფორმირებელის გამოსასვლელზე არსებული გადატანის P_1 სიგნალი;

- SM_1 სუმატორის P_i და SM_2 სუმატორის B_i შესასვლელს მიეწოდებათ ლოგიკური 0-ები;

P_1=1-ის დროს 4-თანრიგიანი ორობით სუმატორის გამოსასვლელზე ფორმირებული $s_3s_2s_1s_0$ შედეგი კორექტირდება; P_1=0-ის დროს კორექცია არ ხდება, რადგან SM_3, SM_2, SM_1 სუმატორების B_i შესასვლელებს 0-ოვანი სიგნალები მიეწოდებათ და ამიტომ გამოსასვლელი სიგნალია $S_3S_2S_1S_0 = s_3s_2s_1s_0$.

ცხრ. 5.1. Σ-ის გამოსახვა **8421** კოდის კოდური სიტყვებით.

Σ	p_4	s_3	s_2	s_1	s_0
0	0	0	0	0	0
1	0	0	0	0	1
2	0	0	0	1	0
3	0	0	0	1	1
4	0	0	1	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	1	0	0	0
9	0	1	0	0	1
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1
16	1	0	0	0	0
17	1	0	0	0	1
18	1	0	0	1	0
19	1	0	0	1	1

$p_4 = 1$ $s_3 = s_2 = 1$ $s_3 = s_1 = 1$

2

შეიძლება ისეთი ორობით-ათობითი სუმატორი ავაგოთ, რომელშიც **4**-თანრიგიანი სუმატორის გამოსასვლელზე მიღებული შედეგის კორექტირება საჭირო არ არის. ასეთი სუმატორის აგების შესაძლებლობის დასასაბუთებლად ვისარგებლოთ **5.2** ცხრილით.

ათობითი **0**-დან **9**-მდე ციფრებისა და გადატანის p_0**=1** ციფრის შეკრებით მიღებული მაქსიმალური რიცხვი **9+9+1=19**-ის ტოლია. **0**-დან დაწყებული **19**-ით დამთავრებული ყველა ეს ათობითი რიცხვი **5.2** ცხრილის **პირველ** (Σ) **სვეტში** არის მოყვანილი. **მომდევნო ხუთ სვეტში** იგივე რიცხვები ჩაწერილი **5**-თანრიგიანი ორობითი რიცხვების სახით. აღნიშნულ ჩანაწერში p_1 წარმოადგენს **4**-თანრიგიანი **ორობითი სუმატორის** გამოსასვლელზე წარმოშობილ გადატანის სიგნალს, ხოლო s_3, s_2, s_1, s_0 – იმავე გამოსასვლელზე მიღებული ჯამის წარმომქმნელ სიგნალებს. **ბოლო ხუთ სვეტში** **0**-დან **19**-მდე ათობითი რიცხვები წარმოდგენილია **8421** კოდის კოდური სიტყვების სახით; ასეთი სახე უნდა ჰქონდეს მათ **4**-თანრიგიანი **ათობით-ორობითი სუმატორის** გამოსასვლელზე. აღსანიშნავია, რომ $\mathbf{P}_1$ გადატანა შეესაბამება უფროსი თანრიგის მნიშვნელობას, ხოლო $S_3S_2S_1S_0$ კოდი – ორთანრიგიანი ათობითი რიცხვის უმცროს თანრიგს, რომელიც **8421** კოდის გამოყენებით არის წარმოდგენილი.

ცხრ. 5.2. ორობით-ათობითი სუმატორის ჭეშმარიტობის საწყისი ცხრილი

	Σ	გამოსასვლელი სიგნალები									
		ორობით სუმატორზე					ორობით-ათობით სუმატორზე				
		p_1	s_3	s_2	s_1	s_0	P_1	S_3	S_2	S_1	S_0
1	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	1	0	0	0	0	1
2	2	0	0	0	1	0	0	0	0	1	0
	3	0	0	0	1	1	0	0	0	1	1
3	4	0	0	1	0	0	0	0	1	0	0
	5	0	0	1	0	1	0	0	1	0	1
4	6	0	0	1	1	0	0	0	1	1	0
	7	0	0	1	1	1	0	0	1	1	1
5	8	0	1	0	0	0	0	1	0	0	0
	9	0	1	0	0	1	0	1	0	0	1
6	10	0	1	0	1	0	0	1	0	1	0
	11	0	1	0	1	1	0	1	0	1	1
7	12	0	1	1	0	0	0	1	1	0	0
	13	0	1	1	0	1	0	1	1	0	1
8	14	0	0	0	0	0	0	0	0	0	0
	15	0	0	0	0	1	0	0	0	0	1
9	16	0	0	0	0	0	0	0	0	0	0
	17	0	0	0	0	1	0	0	0	0	1
10	18	0	0	0	0	0	0	0	0	0	0
	19	0	0	0	0	1	0	0	0	0	1

5.2 ცხრილი შეიძლება განვიხილოთ: ა) ხუთი შესასვლელისა და ხუთი გამოსასვლელის მქონე კომბინაციური დისკრეტული მოწყობილობის ჭეშმარიტობის ცხრილად (შესასვლელი სიგნალებია p_1, s_3 , s_2, s_1 და s_0 , ხოლო გამოსასვლელი სიგნალები - P_1, S_3, S_2, S_1 და S_0) ; ბ) კოდების გარდაქმნის ცხრილად.

პირველ შემთხვევაში ათობით-ორობითი სუმატორის სინთეზის ამოცანა გარდაიქმნება 5 შესასვლელისა და 5 გამოსასვლელის მქონე კომბინაციური დისკრეტული მოწყობილობის სინთეზის კლასიკურ ამოცანად, ხოლო **მეორე შემთხვევაში** – **კოდების გარდამქნელის** სინთეზის ამოცანად.

5.2 ცხრილის ანალიზი სინთეზის ამოცანის მნიშვნელოვნად გამარტივების საშუალებას გვაძლევს; მართლაც, რადგან $S_0=s_0$, ამიტომ შეიძლება ასაგები კომბინაციური დისკრეტული მოწყობილობის შესასვლელებისა და გამოსასვლელების რაოდენობები ერთით შევამციროთ; გარდა ამისა, 5.2 ცხრილში ფიგურული ცვლადებით მონიშნულ მწკრივებში

p_1, s_3, s_2, s_1 ცვლადები ერთნაირ მნიშვნელობებს იღებს; ამიტომ შესაძლებელია ისინი გავაერთიანოთ. ამის შედეგად მწკრივების რაოდენობა **10**-მდე შემცირდება (ცხრ. **5.3**).

4 არგუმენტის მნიშვნელობათა ნაკრების საერთო რაოდენობა 2^4**=16**-ს ტოლია, ამიტომ **5.3** ცხრილს p_1, s_3 , s_2 , s_1 ცვლადების დანარჩენი **6** ნაკრებიც (**1010, 1011, 1100, 1101,1110,1111**) უნდა დავუმატოთ (ისინი **5.3** ცხრილში მუქ ფონზეა აღნიშნული); ვინაიდან აღნიშნულ ნაკრებზე ფუნქცია განსაზღვრული არ არის (**5.2** ცხრილი მათ არ შეიცავს), ამიტომ გამოსასვლელების შესაბამისი მნიშვნელობები "~" სიმბოლოთი აღვნიშნოთ (იხ. პარაგრაფი **4.9**).

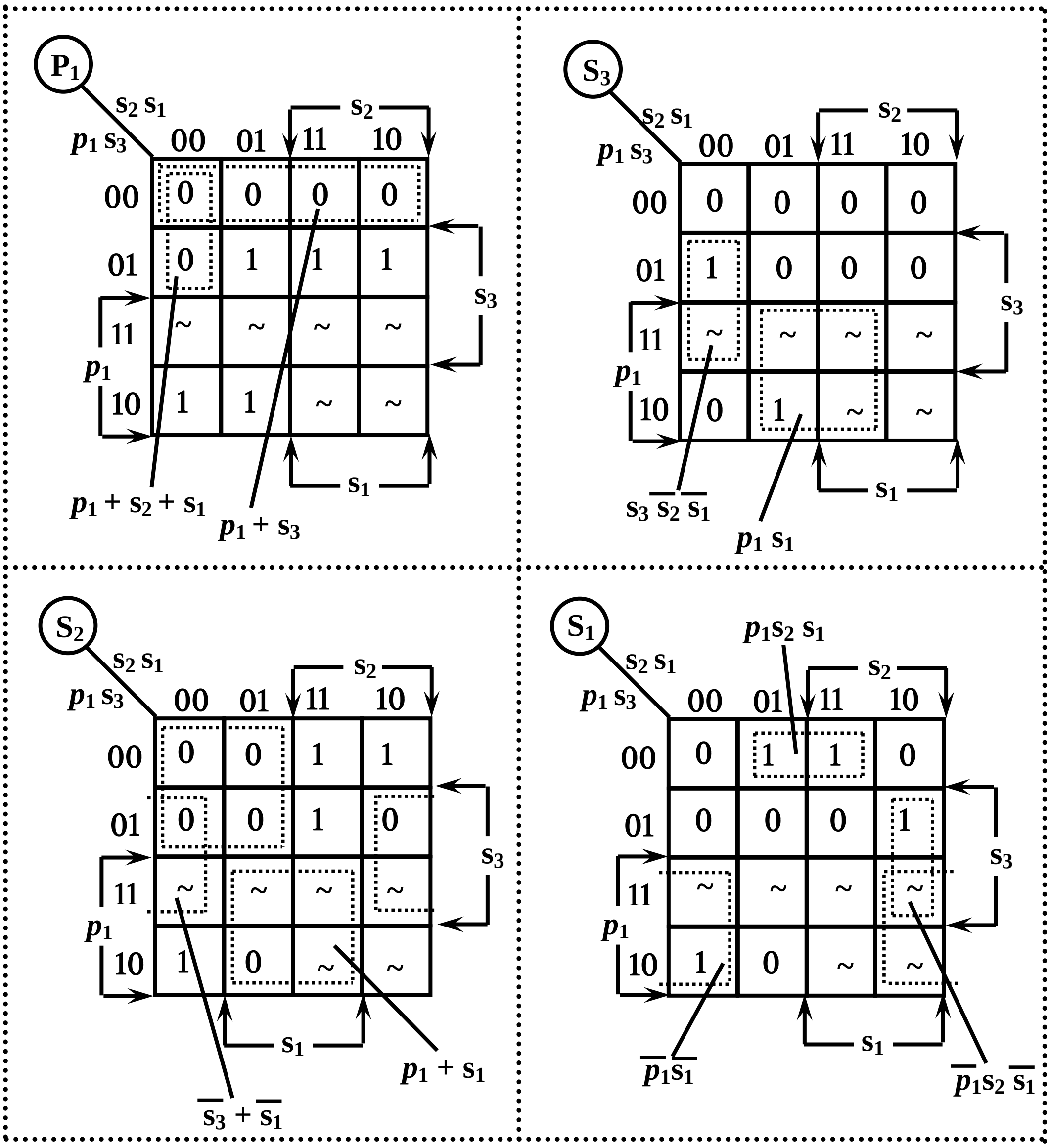
გამოსასვლელი სიგნალების გამომსახველი ლოგიკური ფუნქციების მინიმიზაცია მოვახდინოთ **5.9** ნახაზზე ნაჩვენები კარნოს ბარათების საშუალებით, საიდანაც შეიძლება დავწეროთ:

$$\left.\begin{array}{ll} P_1=(p_1+s_3)(p_1+s_2+s_1)=\overline{p_1}\,\overline{s_3}+\overline{\overline{p_1}\,\overline{s_2}\,\overline{s_1}}, & S_3=p_1 s_1+s_3\overline{s_2}\,\overline{s_1}, \\ S_2=(p_1+s_3)(s_3+s_1)(\overline{p_1}+\overline{s_1})=\overline{p_1}\,\overline{s_2}+s_3\overline{s_1}+p_1 s_1, & S_3=p_1\overline{s_1}+\overline{p_1}\,\overline{s_3}\,s_1+s_3 s_2\overline{s_1}, \end{array}\right\} \quad (5.11)$$

ლოგიკურ ფუნქციათა **(5.11)** სისტემის მიხედვით აგებული კოდების გარდამქმნელიანი ერთთანრიგიანი ათობით-ორობითი სუმატორის სქემა **5.10** ნახაზზეა მოცემული.

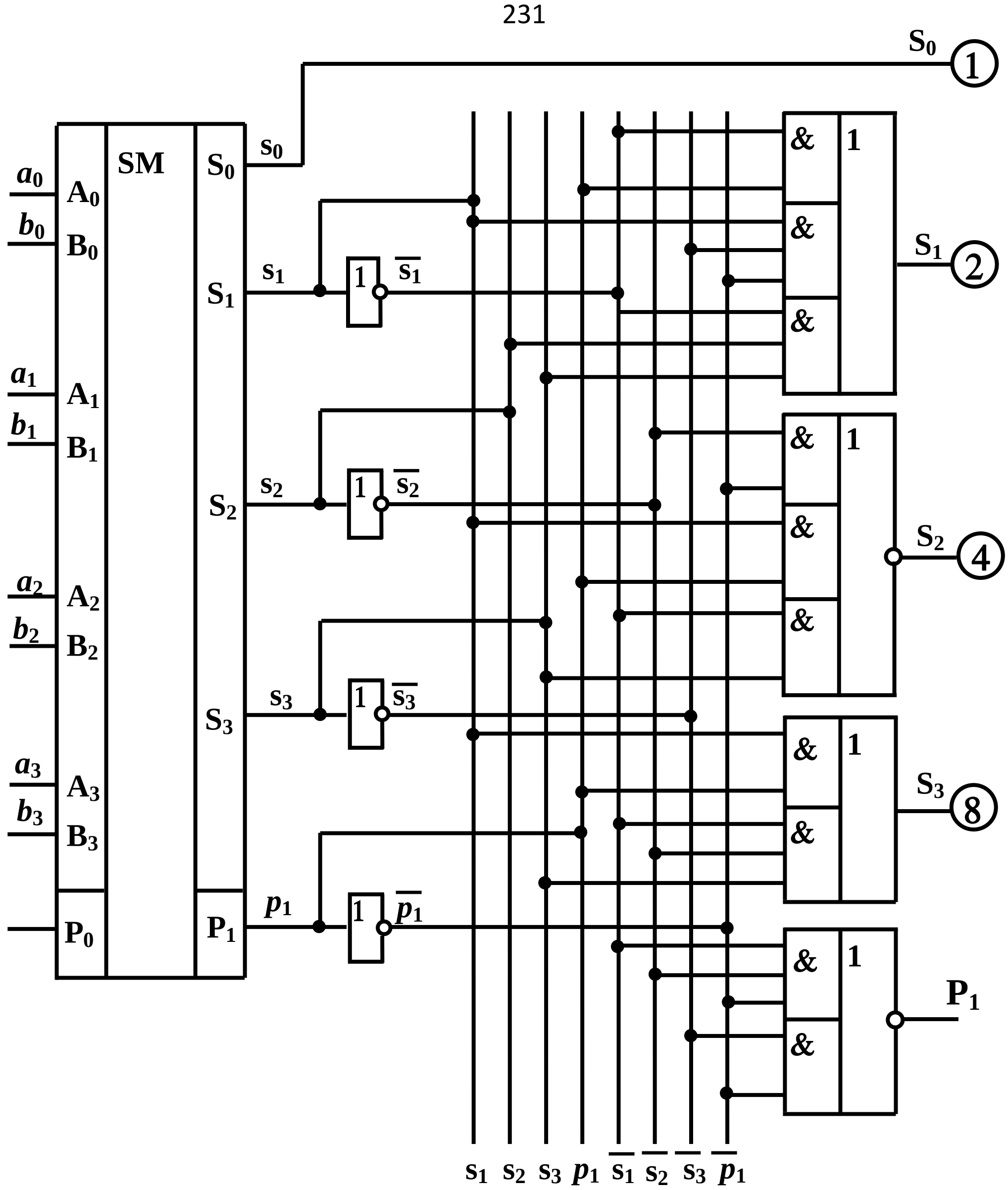
ცხრ. 5.3. ორობით-ათობითი სუმატორის ჭეშმარიტობის საბოლოო ცხრილი

k	შესასვლელი სიგნალები				გამოსვლელი სიგნალები			
	p_1	s_3	s_2	s_1	P_1	S_3	S_2	S_1
0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0
11	0	0	0	0	~	~	~	~
12	0	0	0	0	~	~	~	~
13	0	0	0	0	~	~	~	~
14	0	0	0	0	~	~	~	~
15	0	0	0	0	~	~	~	~
16	0	0	0	0	~	~	~	~



ნახ.5.9. გარდაქმნის სქემის ლოგიკური ფორმულების კარნოს ბარათები

ლოგიკურ ფუნქციათა **(5.11)** სისტემის მიხედვით აგებული კოდების გარდამქმნელიანი ერთთანრიგიანი ათობით-ორობითი სუმატორის სქემა **5.10** ნახაზზეა მოცემული.



ნახ. 5.10. კოდების გარდამქმნელიანი ათობით-ორობითი სუმატორი

5.4. ორობით-ათობითი გამომკლები სუმატორები

ორ დადებით ერთთანრიგიან რიცხვებს შორის სხვაობის ან სხვადასხვა ნიშნიანი ორი რიცხვის ალგებრული ჯამის გამოსათვლელად ვისარგებლოთ **9**-მდე დამატებით **8421** კოდად პირდაპირი **8421** კოდის გარდაქმნის ოპერაციით. გავეცნოთ აღნიშნულ ოპერაციას.

პირდაპირი **8421** კოდით წარმოდგენილი ათობით-ორობითი რიცხვი შედგება Q_i ტეტრადებისაგან, რომლებსაც შეესაბამება გარკვეული ათობითი $k\in\{0,1,2,\ldots,9\}$ ციფრები. **9**-მდე დამატებით **8421** კოდად მისი გარდაქმნისათვის საჭიროა ვისარგებლოთ შემდეგი ალგორითმით:

1. განვიხილოთ **8421** კოდით წარმოდგენილი ათობით-ორობითი რიცხვი;

2. ათობით-ორობითი რიცხვის შემადგენლობაში არსებული თითოეული $\mathbf{Q}_i$ ტეტრად-ისათვის განვსაზღვროთ მისი შესაბამისი ათობითი k_i ციფრი;

3. პირველ პუნქტში განსაზღვრული ათობითი k_i ციფრი შევცვალოთ $\ddot{k}_\iota = (9 - k)$ ციფრით;

4. განვსაზღვროთ ათობითი $\ddot{k}_\iota$ ციფრის შესაბამისი ოთხნიშნა ორობითი $\ddot{Q}_\iota$ რიცხვი;

5. ათობით-ორობით რიცხვში არსებული თითოეული $\mathbf{Q}_i$ ტეტრადა შევცვალოთ ოთხ-ნიშნა ორობითი $\ddot{Q}_\iota$ რიცხვით;

6. ალგორითმის დასასრული.

ცხრ. 5.4. **8421** კოდის **9**-მდე დამატებად კოდად გარდაქმნის ცხრილი

	შესასვლელები (8421 კოდი)				გამოსასვლელები (9-მდე დამატებადი კოდი)			
	A_3	A_2	A_1	A_0	d_3	d_2	d_1	d_0
0	0	0	0	0	1	0	0	1
1	0	0	0	1	1	0	0	0
2	0	0	1	0	0	1	1	1
3	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	1
5	0	1	0	1	0	1	0	0
6	0	1	1	0	0	0	1	1
7	0	1	1	1	0	0	1	0
8	1	0	0	0	0	0	0	1
9	1	0	0	1	0	0	0	0

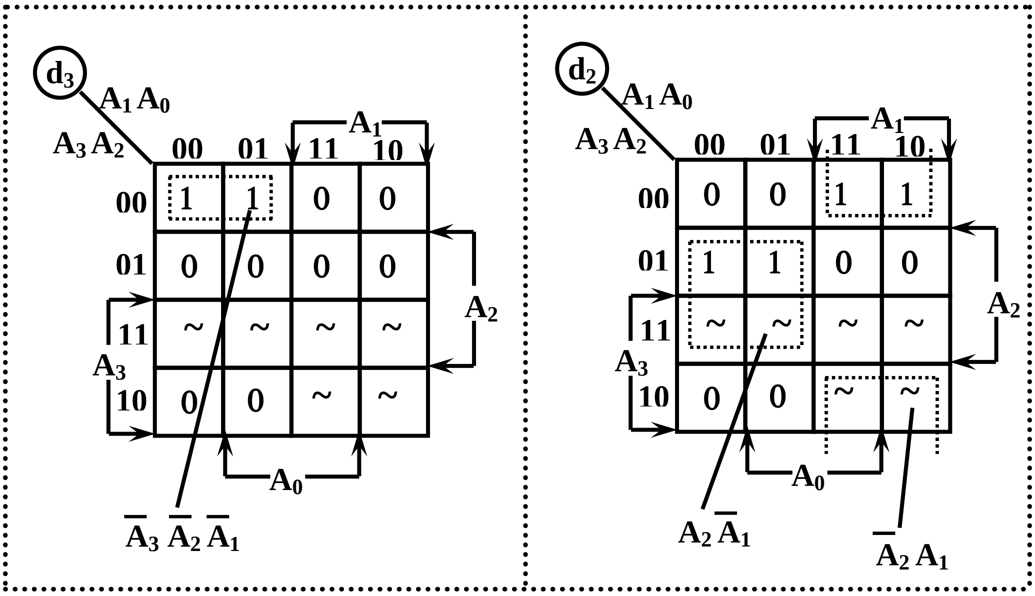
ზემოთ ფორმულირებული ალგორითმით განსაზღვრულ ოპერაციას შემოკლებით **9-მდე დამატებადი 8421 კოდის გარდაქმნის ოპერაცია** ეწოდება და ჭეშმარიტობის სათანადო ცხრილის შესაბამისად სრულდება (ცხრ. **5.4**).

5.4 ცხრილის სვეტების ერთმანეთთან შედარებით ჩანს, რომ:

$$d_0 = \overline{A}_0 ; \quad d_1 = A_1 . \tag{5.12}$$

დარჩენილი d_2, d_3 პარამეტრები შეიძლება მივიღოთ **5.11** ნახაზზე მოყვანილი კარნოს ბარათების დახმარებით. მათი შედგენის დროს გათვალისწინებულია ის ფაქტი, რომ $k = 10 \div 15$ კომბინაციები არასოდეს გამოიყენება. კარნოს აღნიშნული ბარათებიდან შეიძლება დავწეროთ:

$$d_3 = \overline{A}_3 \overline{A}_2 \overline{A}_1 = \overline{A_3 + A_2 + A_1} ; \qquad d_2 = A_2 \overline{A}_1 + \overline{A}_2 A_1 = A_2 \oplus A_1 . \tag{5.13}$$



ნახ. 5.11. კარნოს ბარათები **9**-მდე დამატებად **8421** კოდის გარდამქმნე-
ლის გამოსასვლელი $\mathbf{d_3}$ და $\mathbf{d_3}$ სიგნალების მიღებისათვის

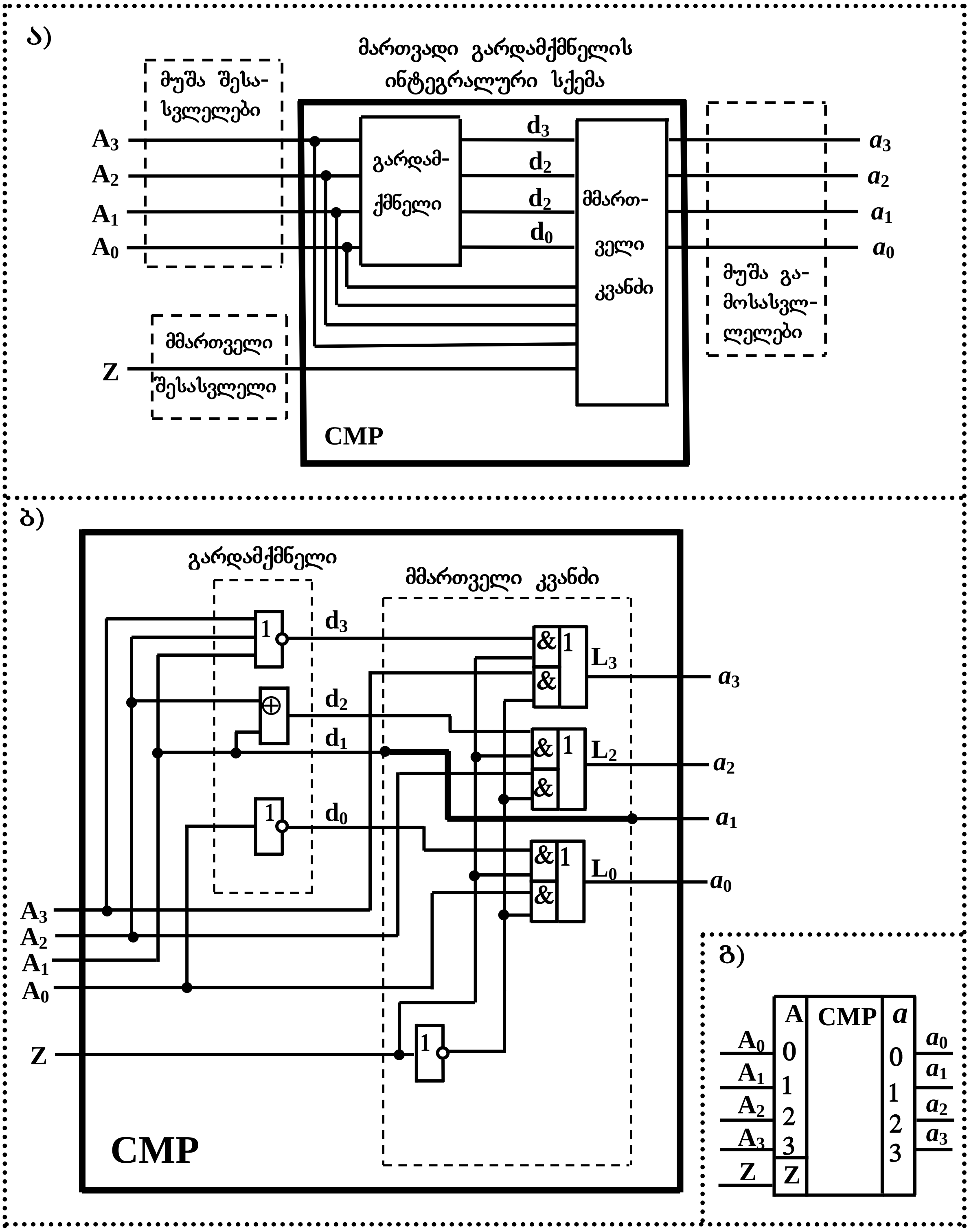
(**5.12**) და (**5.13**) ფორმულების გამოყენებით აგებულ მოწყობილობას, რომელიც პირდაპირ **8421** კოდს გარდაქმნის **9**-მდე დამატებით **8421** კოდად, შემოკლებით **გარდამქნელი** ვუწოდოთ. გარდამქმნელი წარმოადგენს გამომკლები სუმატორის ერთ-ერთ მთავარ კვანძს.

ინტეგრალური სქემის გამოყენებით აგებული გამომკლები სუმატორის ბლოკ-სქემა **5.12,ა** ნახაზზეა ნაჩვენები. მის მუშა შესასვლელს მიეწოდება შესასვლელი ოპერანდის ცალკეული $\mathbf{A_3,A_2,A_1,A_0}$ თანრიგების სიგნალები; ოთხი სიგნალის ერთობლობა წარმოქმნის $\mathbf{A_3A_2A_1A_0}$ ტეტრადას, რომელიც გადაეცემა როგორც გარდამქმნელის შესასვლელებს, ასევე მმართველი კვანძის **ქვედა შესასვლელებს**.

გარდამქმნელი (**5.12**) და (**5.13**) ფორმულების შესაბამისად გამოიმუშავებს **9**-მდე დამატებადი **8421** კოდის $\mathbf{d_3d_2d_1d_0}$ ტეტრადას (კოდურ სიტყვას), რომელიც გადაეცემა მმართველი კვანძის **ზედა შესასვლელებს.**

საბოლოოდ მმართველი კვანძის ზედა შესასვლელებზე მოდებული აღმოჩნდება $\mathbf{d_3d_2d_1d_0}$, ხოლო ქვედა შესასვლელებზე - $\mathbf{A_3A_2A_1A_0}$ ტეტრადა (იხ. ნახ. **5.12,ა**). მმართველი კვანძის დანიშნულებაა ამ ორი ტეტრადიდან ერთ-ერთი გადასცეს გამომკლები სუმატორის მუშა $\boldsymbol{a_3a_2a_1a_0}$ გამოსასვლელს. ტეტრადის ამორჩევის ბრძანებას გასცემს **მმართველი შესასვლელი**, რომელზედაც **Z** სიგნალი ფორმირდება; კერძოდ:

- თუ **Z=0,** მაშინ მმართველი კვანძი სუმატორის მუშა გამოსასვლელისაკენ გზას უხსნის $\mathbf{A_3A_2A_1A_0}$ და გზას უკეტავს $\mathbf{d_3d_2d_1d_0}$ ტეტრადას;
- თუ **Z=1,** მაშინ მმართველი კვანძი სუმატორის მუშა გამოსასვლელისაკენ გზას უხსნის $\mathbf{d_3d_2d_1d_0}$ და გზას უკეტავს $\mathbf{A_3A_2A_1A_0}$ ტეტრადას.



ნახ. 5.12. პირდაპირი 8421 კოდის დამატებით კოდად **მართვადი გარდამქმნელის** ბლოკური (ა), პრინციპული (ბ) სქემა და პირობითი გამოსახულება (გ).

გარდამქმნელისა და მმართველი კვანძის ერთობლიობით მიღებულ მოწყობილობას **მართვადი გარდამქმნელი** ეწოდება. მისი პრინციპული სქემა **5.12,ბ** ხოლო პირობითი გამოსახულება – **5.12,გ** ნახაზზეა მოყვანილი. ლიტერატურაში მართვამდი გარდამქმნელის აღსანიშნავად იყენებენ აბრევიატურას **CMP** (**Co-MP**lement – ''დამატებითი'').

5.11,ბ ნახაზზე ზემოთ აღნიშნული გარდამქმნელი და მმართველი კვანძი პუნქტირული ხაზებითაა შემოფარგლული. გარდამქმნელის სტრუქტურა (**5.12**) და (**5.13**) გამოსახულების შესაბამისადაა აგებული; განვიხილოთ მმართველი კვანძის სტრუქტურის აგების პრინციპები.

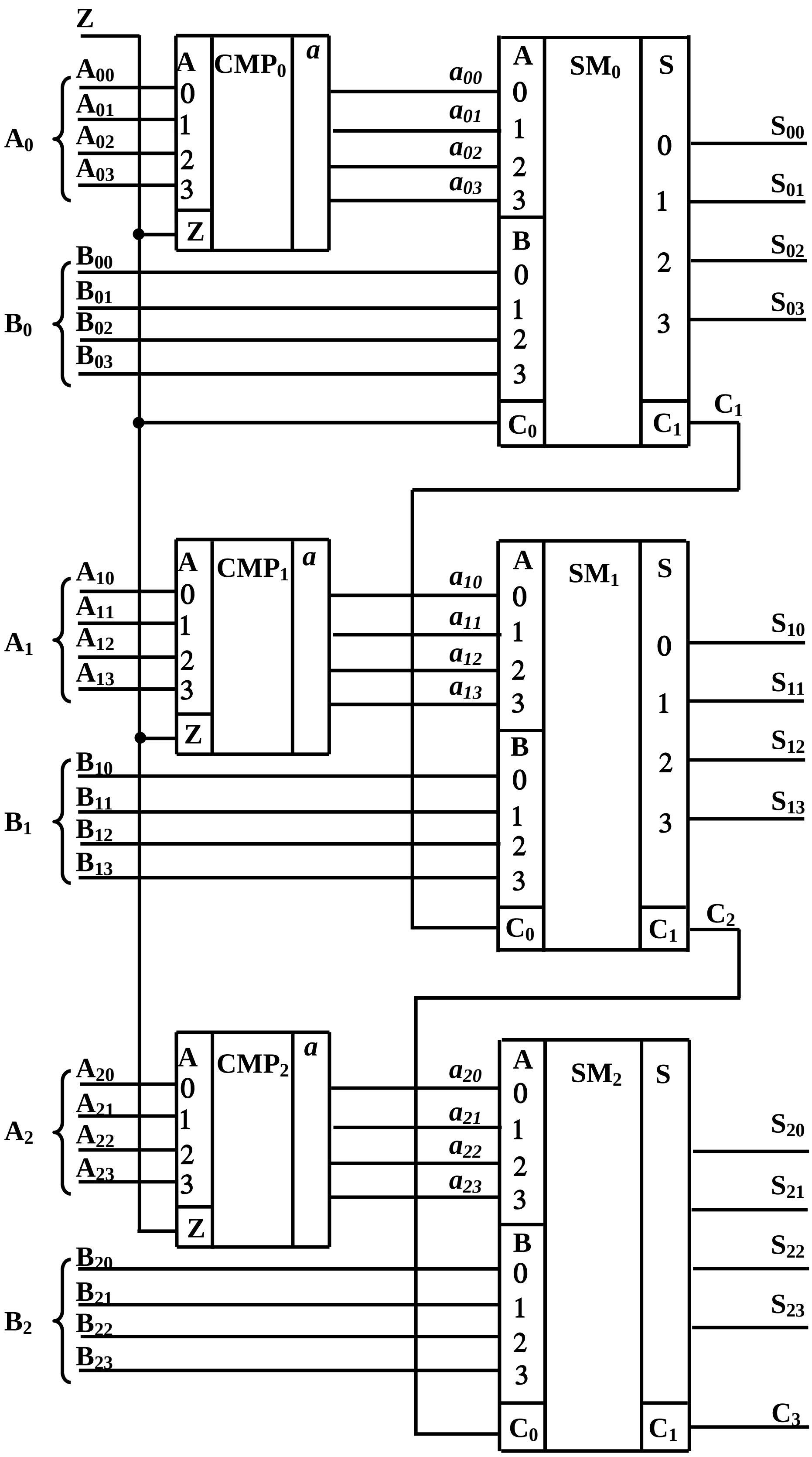
(**5.12**) გამოსახულების თანახმად $d_1 = A_1$, რის გამოც მართვადი გარდამქმნილის a_1 გამოსასვლელს უშუალოდ უნდა მიუერთდეს გარდამქმნელის d_1 გამოსასვლელი; d_1 და a_1 გამოსასვლელების შემაერთებელი სადენი ნახაზზე კონტურული ხაზით არის ნაჩვენები.

შესასვლელი ოპერანდის თანრიგის A_0 სიგნალი და გარდამქმნელის d_0 სიგნალი a_0 გამოსასვლელს უკავშირდება ორი კონიუნქტორისა (**და**-ელემენტისა) და ერთი დიზიუნქტორისაგან (**ან**-ელემენტისაგან) შემდგარი ლოგიკური L_0 ელემენტის საშუალებით; ასეთ ელემენტს **2და-არა** ტიპის ელემენტი ეწოდება. მისი ზედა **და**-ელემენტი ახდენის d_0 და Z სიგნალების, ხოლო ქვედა **და**-ელემენტი – A_0 და $\overline{Z}$ სიგნალების ლოგიკურ გამრავლებას.

შესასვლელი ოპერანდის A_2 სიგნალი და გარდამქმნელის d_2 სიგნალი a_2 გამოსასვლელს უკავშირდება **2და-არა** ტიპის ლოგიკური L_2 ელემენტის საშუალებით; მისი ზედა კონიუნქტორი ახდენს d_2 და Z სიგნალების, ხოლო ქვედა კონიუნქტორი – A_2 და $\overline{Z}$ სიგნალების ლოგიკურ გამრავლებას.

შესასვლელი ოპერანდის A_3 სიგნალი და გარდამქმნელის d_3 სიგნალი a_3 გამოსასვლელს უკავშირდება **2და-არა** ტიპის ლოგიკური L_3 ელემენტის საშუალებით; მისი ზედა კონიუნქტორი ახდენს d_3 და Z სიგნალების, ხოლო ქვედა კონიუნქტორი – A_3 და $\overline{Z}$ სიგნალების ლოგიკურ გამრავლებას.

$Z = 0$-ის შემთხვევაში ღიაა **2და-არა** ელემენტების ქვედა კონიუნქტორები და მუშა გამოსასვლელებს მიეწოდება შესასვლელი ოპერანდის d_3, d_2, d_0 სიგნალები, ხოლო $Z = 1$-ის შემთხვევაში ღიაა **2და-არა** ელემენტების ქვედა კონიუნქტორები და მუშა გამოსასვლელებს მიეწოდებათ **9**-მდე დამატებითი **8421** კოდის A_3, A_2, A_0 სიგნალები.



ნახ.5.13. 3-თანრიგიანი გამომკლები სუმატორის სქემა

3 **3**-თანრიგიანი ათობით-ორობითი **გამომკლები სუმატორის პრინციპული სქემა 5.13** ნახაზზეა მოყვანილი. იგი შეიცავს სამ **CMP**$_i$ გარდამქმნელსა და ამდენივე ათობით-ორობით **SM**$_i$ სუმატორს (i**=1,2,3**). მმართველი **Z** სიგნალი იღებს **0**-ის ან **1**-ის ტოლ მნიშვნელობას; ამასთანავე, თუ

- **Z = 0,** მაშინ სუმატორი ასრულებს ოპერანდების **შეკრების ოპერაციას;**
- **Z = 1,** მაშინ სუმატორი ასრულებს ოპერანდების **გამოკლების ოპერაციას;**

CMP$_i$ გარდამქმნელები ასრულებს შემდეგ ფუნქციებს:

$$a_i = \begin{cases} A_i, & \text{თუ } Z = 0, \\ (9 - A_i), & \text{თუ } Z = 1, \end{cases} \quad \text{ან} \quad a_{10} = \begin{cases} A_{10}, & \text{თუ } Z = 0, \\ (999 - A_{10}), & \text{თუ } Z = 1, \end{cases}$$

სადაც $a = a_2 a_1 a_0 = a_2\,10^2 + a_1\,10^1 + a_0\,10^2$, $\mathbf{A} = \mathbf{A}_2\mathbf{A}_1\mathbf{A}_0 = \mathbf{A}_2\,10^2 + \mathbf{A}_1\,10^1 + \mathbf{A}_0\,10^2$, a_i, $\mathbf{A}_i$ - მართვად გარდამქმნელთა შესასვლელებსა და გამოსასვლელებზე არსებული **4**-თანრიგიანი ოპერანდები (ტეტრადები): $a_i = a_{i3}\,a_{i2}\,a_{i1}\,a_{i0}$, $A_i = A_{i3}\,A_{i2}\,A_{i1}\,A_{i0}$, $i = 0,1,2.$

ორობით-ათობითი **SM**$_i$ სუმატორები გამოითვლის შემდეგ ჯამს:

$$S_i = B_i + a_i + C_i\,, \quad \text{ან} \quad S_{10} = \begin{cases} B_{10} + A_{10}\,, & \text{თუ } Z = 0, \\ 1000 + (B_{10} - A_{10}), & \text{თუ } Z = 0, \end{cases}$$

სადაც $S_i = S_{i3}\,S_{i2}\,S_{i1}\,S_{i0}$; $B_i = B_{i3}\,B_{i2}\,B_{i1}\,B_{i0}$; C_i არის i–ური სუმატორის შესასვლელზე მოსული გადატანის სიგნალი, ამასთანავე $C_i = \mathbf{Z}$; $\mathbf{B}_{10}$, $\mathbf{A}_{10}$, $\mathbf{S}_{10}$ – შესაბამისად წარმოადგენს შესასვლელ ოპერანდებსა და ჯამს თვლის ათობით სისტემაში.

გამოკლების დროს (**Z=1**) ჯამი წარმოიდგინება ათობითი დამატებითი კოდის სახით და მას შეიძლება ჰქონდეს დადებითი ან უარყოფითი მნიშვნელობა. ჯამის მნიშვნელობა და ნიშანი გადატანის C_3 სიგნალის მნიშვნელობით განისაზღვრება და მას აქვს შემდეგი სახე:

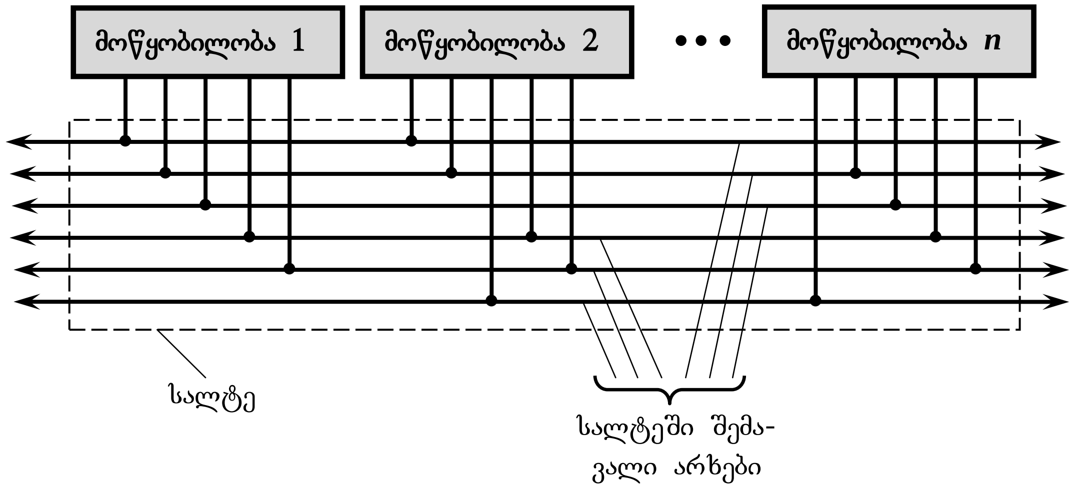
$$S = \begin{cases} \mathbf{B} - \mathbf{A}\,, \text{ თუ } (\mathbf{B} - \mathbf{A}) \geq 0; & \text{ამ დროს } C_3 = 1, \\ 1000\text{-მდე დამატებას, თუ } (\mathbf{B} - \mathbf{A}) < 0; & \text{ამ დროს } C_3 = 0. \end{cases}$$

ნიშნის თანრიგს წარმოადგენს **C**$_3$ გადატანა. შეკრების დროს თანრიგობრივი ბადე რომ არ გადაივსოს, საჭიროა სრულდებოდეს პირობა: **B + A ≤ 999.**

5.5. მულტიპლექსირება და მისი გამოყენება ლოგიკური ფუნქციების რეალიზებისათვის. მულტიპლექსორი და დემულტიპლექსორი

1 კომპიუტერული სისტემის ფუნქციონირებისათვის მასში შემავალ ცალკეულ მოწყობილობებს უნდა შეეძლოს ერთმანეთს შორის სიგნალებისა და კოდების ურთიერთგაცვლა. ამისათვის აუცილებელია მათ შორის სპეციალური კავშირების ორგანიზება. არსებობს შემდეგი ორი სტრუქტურის მქონე კავშირები:

- **კლასიკური სტრუქტურის კავშირები.** ამ დროს თითოეული მოწყობილობა ნებისმიერ სხვა მოწყობილობასთან კავშირის საკუთარი ხაზითაა დაკავშირებული და მათ შორის სიგნალებისა და კოდების გაცვლა ამ ხაზით ხდება;



ნახ. **5.14**. სალტური სტრუქტურის კავშირები.

● **სალტური სტრუქტურის კავშირები.** ამ დროს დამოუკიდებლად არსებობს კავშირის ხაზები, რომლებსაც უერთდება ცალკეული მოწყობილობები (ნახ. **5.14**). მოწყობილობებს შორის გასაცვლელი ყველა სიგნალი და კოდი კავშირის ერთი და იგივე ხაზით, ოღონდ დროის სხვადასხვა მომენტში გადაიცემა; ამასთანავე, კავშირის ნებისმიერ ხაზში გადაცემა ორივე მიმართულებით ხდება, ე.ი. გვაქვს **ორმხრივი გადაცემა.** ამის შედეგად ხაზების რაოდენობა მნიშვნელოვნად მცირდება, ხოლო ინფორმაციის გაცვლის წესები (პროტოკოლები) მარტივდება.

კავშირის ხაზს ხშირად სიგნალების ან კოდების ნაკადების ‘‘გასადინებლად’’ განკუთვნილ **არხად** განიხილავენ. ერთი არხით რამდენიმე ინფორმაციის გადაცემას არხის **შემჭიდროება** ეწოდება. სიგნალების ან კოდების გადასაცემად განკუთვნილი არხების ჯგუფს **სალტე** (ინგლ. **bus**) ეწოდება.

სხვა სიტყვებით რომ ვთქვათ, **კლასიკური სტრუქტურის შემთხვევაში კავშირის არხებია მიერთებული მოწყობილობებთან, ხოლო სალტური სტრუქტურის დროს პირიქით – კავშირის არხებთანაა მიერთებული მოწყობილობები.**

მიღებულია შემდეგი განსაზღვრება:

მულტიპლექსირება (ინგლ. ***multiplexing, muxing*)** წარმოადგენს კავშირგაბმულობის ხაზის შემჭიდროებას, ე.ი. ერთ ხაზში ინფორმაციის რამდენიმე ნაკადის ერთდროულად გადაცემას, რაც ზრდის არხის გამტარობის უნარს.

სიგნალებისა და კოდების გადაცემის ზემოთ აღწერილ ხერხს **მულტიპლექსური გადაცემა** ეწოდება. მულტიპლექსური გადაცემისათვის განკუთვნილი ხაზების ერთობლიობა ე.წ. **სალტეს** წარმოქმნის (იხ. ნახ. **5.14**) და ამიტომ კავშირების მოცემულმა სტრუქტურამ **სალტური სტრუქტურის** სახელწოდება მიიღო

კლასიკური სტრუქტურის კავშირების დროს საჭიროა დიდი რაოდენობის ხაზების გამოყენება და, გარდა ამისა, ნებისმიერი ორი მოწყობილობა ინფორმაციას ერთმანეთს სპეციალური პროტოკოლის დაცვით უცვლის, ე.ი. მრავალფეროვანია აღნიშნული პროტოკოლების სახეები. კავშირების ხაზების რაოდენობისა და ინფორმაციის გადასაცემად საჭირო პროტოკოლების მრავალფეროვნების შემცირება **სალტური სტრუქტურის კავშირების** გამოყენებითაა შესაძლებელი და ამიტომ იგი გამოიყენება კომპიუტერულ სისტემებში.

2

არსებობს მულტიპლექსირების შემდეგი სახეები:

- მულტიპლექსირება **სიხშირული დაცალკევების გზით** (ინგლ. **FDM**, **F**requency **D**ivision **M**ultiplexing);
- მულტიპლექსირება **დროითი დაცალკევების გზით** (ინგლ. **TDM**, **T**ime **D**ivision **M**ultiplexing);
- მულტიპლექსირება **ტალღის სიგრძის მიხედვით დაცალკევების გზით** (ინგლ. **WDM**, **W**avelength **D**ivision **M**ultiplexing)

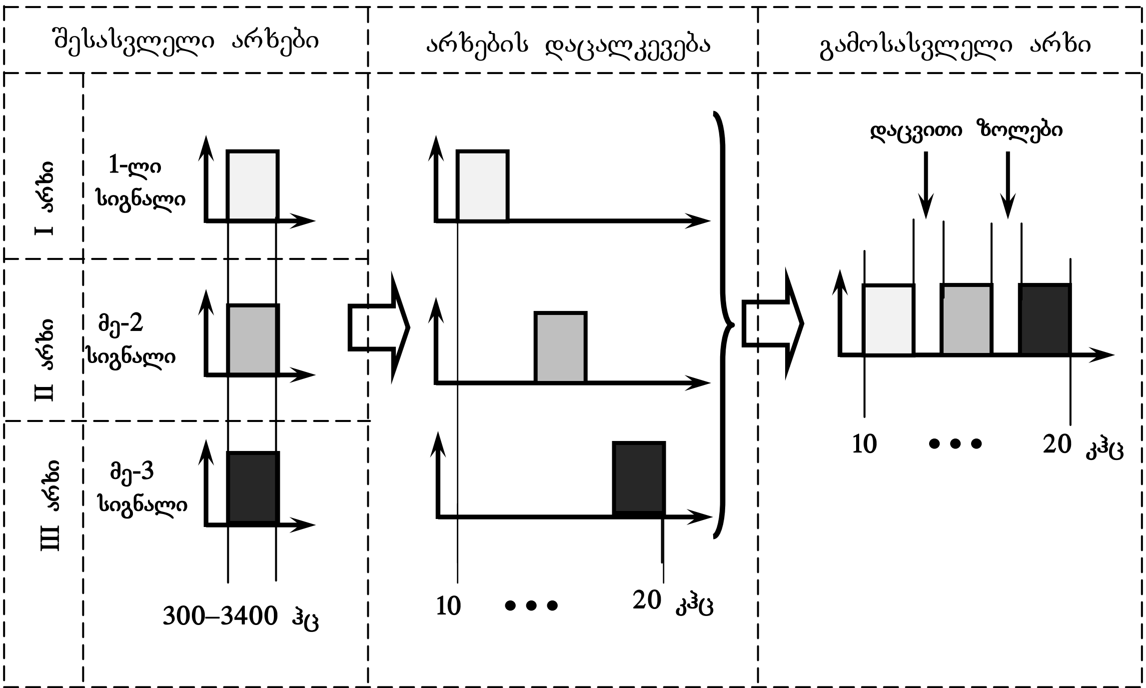
მოკლედ განვიხილოთ თითოეული მათგანი.

სიხშირული დაცალკევების გზით მულტიპლექსირების დროს ფართო სიხშირული ზოლის მქონე არხის ფარგლებში ხდება ვიწრო სიხშირული ზოლის მქონე რამდენიმე არხის განთავსება.

დროითი დაცალკევების გზით მულტიპლექსირების დროს მონაცემები კადრებად გადაიცემა; ამ დროს ნაკლები სიგანის (გადაცემის უნარის) მქონე არხებიდან დიდი სიგანის არხზე გადასვლის გამო ამ უკანასკნელის ერთი კადრის ფარგლებში თავისუფლდება რეზერვი ნაკლები მოცულობის რამდენიმე კადრის გადასაცემად.

ტალღის სიგრძის მიხედვით დაცალკევების გზით მულტიპლექსირების ტექნოლოგიას საფუძვლად უდევს ის ფაქტი, რომ სხვადასხვა სიგრძეების მქონე ტალღები ერთმანეთისაგან დამოუკიდებლად ვრცელდება და მათი საშუალებით ერთ ოპტიკურ-ბოჭკოვან კაბელში რამდენიმე არხის ორგანიზებაა შესაძლებელი.

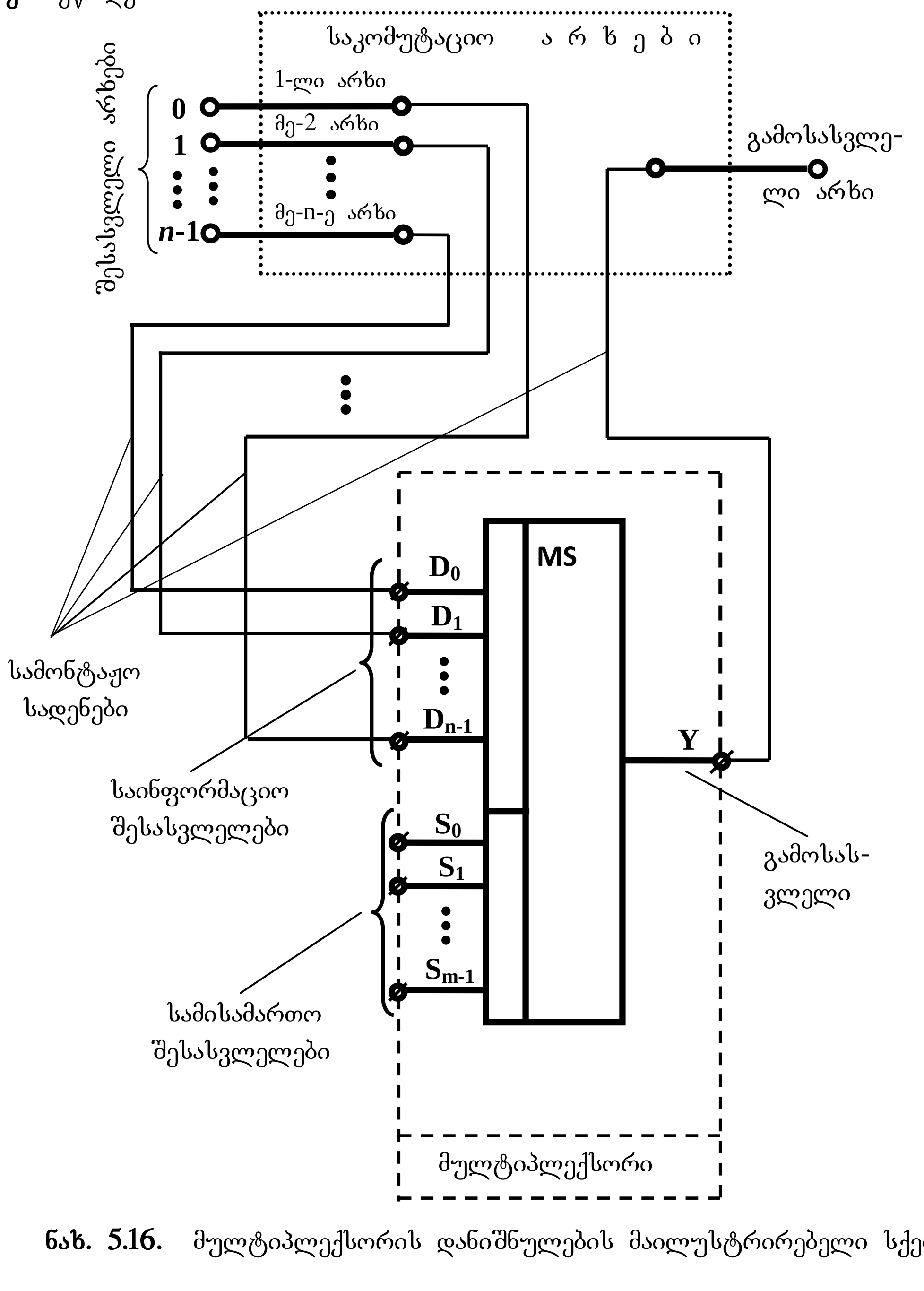
მულტიპლექსირების ზემოთ ჩამოთვლილი სახეების განხილვა ცდება ჩვენი დამხმარე სახელმძღვანელოს განხილვის ფარგლებს, მაგრამ თვალსაჩინოებისათვის მოკლედ შევეხებით სიხშირული დანაწევრების გზით სამი არხის მულტიპლექსირების კერძო შემთხვევას (**ნახ. 5.15**).



ნახ. 5.15. სამი არხის მულტიპლექსირება სიხშირული დაცალკევების გზით

მრავალარხიანი კავშირგაბმულობის განვითარების დასაწყისში კავშირგაბმულობის საქალაქთაშორისო არხებით მხოლოდ სატელეფონო ინფორმაცია გადიცემოდა; შემდგომში კავშირგაბმულობის სხვა სახეებიც იქნა ფორმირებული, მაგრამ მათ შორის ყველაზე მრავალრიცხოვანი დღესაც სატელეფონო კავშირი რჩება. აღნიშნულიდან გამომდინარე, გადაცემის მრავალარხიანი სისტემების აპარატურით წარმოშობილი **კავშირგაბმულობის არხის ძირითად ტიპად** სალაპარაკო (ბგერითი) სიგნალების გადაცემის უზრუნველმყოფი არხი მიიჩნევა. ასეთი ტიპის არხს ეწოდება **ტონალური სიხშირის არხი.**

ტონალური სიხშირის არხი ეწოდება ტექნიკური საშუალებების ერთობლიობას, რომელიც უზრუნველყოფს **300**-დან **3400** ჰერცამდე სიხშირეთა ზოლში მოთავსებული სიგნალების გადაცემას, ვინაიდან სწორედ ასეთ ზოლშია მოთავსებული სალაპარაკო ბგერები. **5.15** ნახაზზე ასეთი სამი არხია ნაჩვენები, რომლებსაც **შესასვლელი არხები ეწოდება.** მასზე ილუსტრირებულია შეზღუდული გამტარობის (**300-3400** ჰც-ის) მქონე ამ სამი არხიდან გაზრდილი გამტარობის (**10-20** კილოჰერცი) მქონე **გამოსასვლელ არხად** წოდებული ერთი არხის ფორმირების პროცესი. აღნიშნულ პროცესს **სიხშირული მულტიპლექსირება** ეწოდება.



ნახ. 5.16. მულტიპლექსორის დანიშნულების მაილუსტრირებელი სქემა

გამოსასვლელ არხში შესასვლელი არხებისათვის გამოყოფილია სიხშირის საკუთარი ზოლები, რომლებიც ერთმანეთისაგან სპეციალური დაცვითი ზოლებითაა დაცალკევებული. რომელიმე შესასვლელი არხიდან ინფორმაციის გადაცემის **საჭიროების წარმოშობ-ბისთანავე**, აღნიშნული არხი **მულტიპლექსორად** წოდებული სპეციალური მოწყობი-ლობით დაუყოვნებლად მიუერთდება საერთო არხს. მაშასადამე **მულტიპლექსორი** სხვა არაფერია, თუ არა **არხების** სპეციფიკური სახის **კომუტატორი,** რომელიც რამდენიმე შესასვლელი არხიდან ერთ-ერთ მათგანს საერთო გამოსასვლელ არხთან აერთებს.

3 მულტიპლექსორი შეიძლება რეალიზებული იქნეს:
- აპარატურულად;
- პროგრამულად.

გამოსასვლელ არხთან ***n*** რაოდენობის შესასვლელი არხებიდან ნებისმიერი საჭირო მათგანის კომუტირებისათვის განკუთვნილი მულტიპლექსორის პირობითი გამოსახულება **5.16** ნახაზზეა ნაჩვენები. მას აქვს:
- ***n*** რაოდენობის საინფორმაციო $\mathbf{D_0, D_1, \dots, D_{n-1}}$ შესასვლელები;
- ***m*** რაოდენობის $\mathbf{S_0, S_1, \dots, S_{m-1}}$ სამისამართო შესასვლელი;
- ერთი **Y** გამოსასვლელი.

ათობით ***n*** და ***m*** რიცხვებს შორის არსებობს შემდეგი დამოკიდებულება:

$$\mathbf{m =]\log_2 n[\ ,} \qquad \textbf{(5.14)}$$

სადაც $\mathbf{]\log_2 n[}$ აღნიშნავს $\mathbf{\log_2 n}$–ზე არანაკლებ მთელ დადებით რიცხვს.

5.16 ნახაზის შესაბამისად **სამონტაჟო სადენების** დახმარებით მულტიპლექსორის:
- **Y გამოსასვლელი** მიერთებულია **გამოსასვლელ არხთან**;
- $\mathbf{D_i}$, $i \in (0,1, \dots, n\text{-}1)$ **შესასვლელი** მიერთებულია ათობითი ***i* რიცხვით დანო-მრილ შესასვლელ არხთან**, რომელიც ამ არხის **ათობით მისამართად** ითვლება.

არხების დანომვრა და მულტიპლექსორის საინფორმაციო არხების ინდექსირება იმ-გვარადაა მოხდენილი, რომ სამართლიანია შემდეგი მტკიცება:

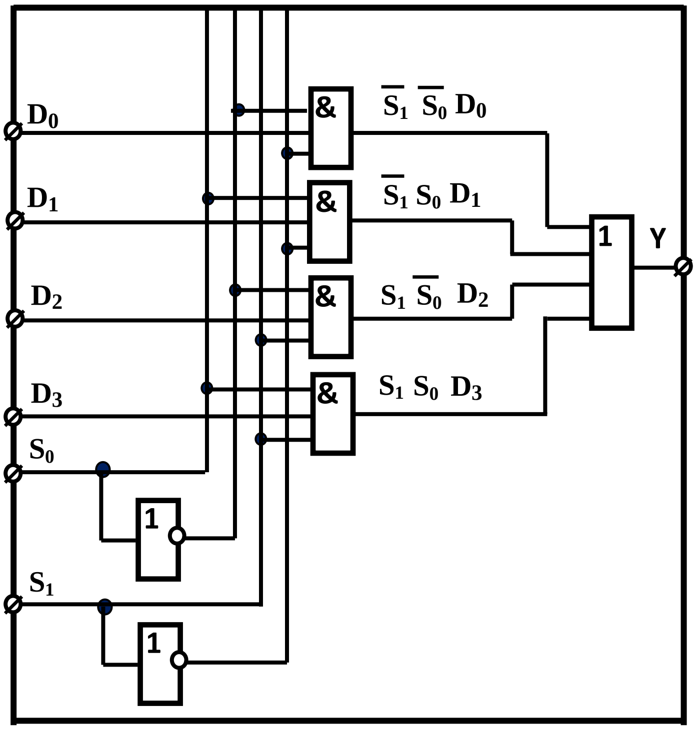
> მულტიპლექსორის საინფორმაციო $\mathbf{D_i}$, $i \in (0,1, \dots, n\text{-}1)$ შესასვლელის ინდექსი გვიჩ-ვენებს ამ შესასვლელთან მიერთებული შესასვლელი არხის ათობით მისამართს. ათობითი ***i*** რიცხვის შესატყვისი **m** ბიტურ ორობით რიცხვს $\mathbf{D_i}$ შესასვლელთან მიერთებული **შესა-სვლელი არხის ორობითი მისამართი** ეწოდება.

შესასვლელი ***i***-ური არხის **საჭიროების წარმოშობა** (იხილეთ მოცემული პარაგრაფის ქვეპუნქტი 2) მულტიპლექსორის **სამისამართო შესასვლელზე** ამ არხის ორობითი მისამართის მიწოდების ტოლფასია. სამისამართო შესასვლელზე ნებისმიერი შე-სასვლელი არხის ორობითი მისამართის მიწოდებისთანავე მულტიპლექსორი ამ არხს გა-მოსასვლელ არხთან მიაერთებს.

4 **n=4** რაოდენობის საინფორმაციო შესასვლელის მქონე მულტიპლექსორისა-თვის (**ნახ. 5.16**) სამისამართო შესასვლელების **m** რაოდენობაა:

$$\mathbf{m =]\log_2 4[\ = 2} \qquad \textbf{(5.15)}$$

აღნიშნულიდან გამომდინარე **n=4** რაოდენობის საინფორმაციო შესასვლელის მქონე მულტიპლექსორს გააჩნია (**ნახ.5.16**):



ნახ. 5.16. n=4 რაოდენობის საინფორმაციო შესა-სვლელებიანი მულტიპლექსორის პი-რობითი გამოსახულება

- საინფორმაციო შესასვლელები $\mathbf{D_0}$, $\mathbf{D_1}$, $\mathbf{D_2}$, $\mathbf{D_3}$;
- სამისამართო შესასვლელები $\mathbf{S_0}$, $\mathbf{S_1}$;
- გამოსასვლელი **Y**.

საინფორმაციო შესასვლელების ორობითი მისამართები შემდეგია:

- $\mathbf{D_0}$ შესასვლელისათვის **00**;
- $\mathbf{D_1}$ შესასვლელისათვის **01**;
- $\mathbf{D_2}$ შესასვლელისათვის **10**;
- $\mathbf{D_3}$ შესასვლელისათვის **00**

საინფორმაციო შესასვლელზე ზემოთ ჩამოთვლილი რომელიმე მათ-განის წარმოშობის შემთხვევაში გა-მოსასვლელს მიუერთდება ამ მისამ-ართის შესაბამისი საინფორმაციო შე-სასვლელი, რომელთანაც, როგორც **5.16** ნახაზიდან ჩანს, სათანადო შე-სასვლელი არხია დაკავშირებული.

განხილული მულტიპლექსორის ფუნქციონირება **5.5** ცხრილშია ილუსტრირებული.

სამისამართო $\mathbf{S}_i$, $i \in (0;1)$ შესასვლელს თუ მიეწოდება:

- ლოგიკური **0** სიგნალი, მაშინ იგი აღვნიშნოთ ინვერსირების ნიშნით, ე.ი როგორც $\mathbf{S}_i$;
- ლოგიკური **1** სიგნალი, მაშინ იგი უინვერსიოდ, ე.ი. $\mathbf{S}_i$ სახით აღვნიშნოთ.

ასეთი აღნიშვნების შემოღების შემთხვევაში შეგვიძლია დავასკვნათ, რომ განხილული მულტიპლექსორის საშუალებით რეალიზდება შემდეგი ლოგიკური ფუნქცია:

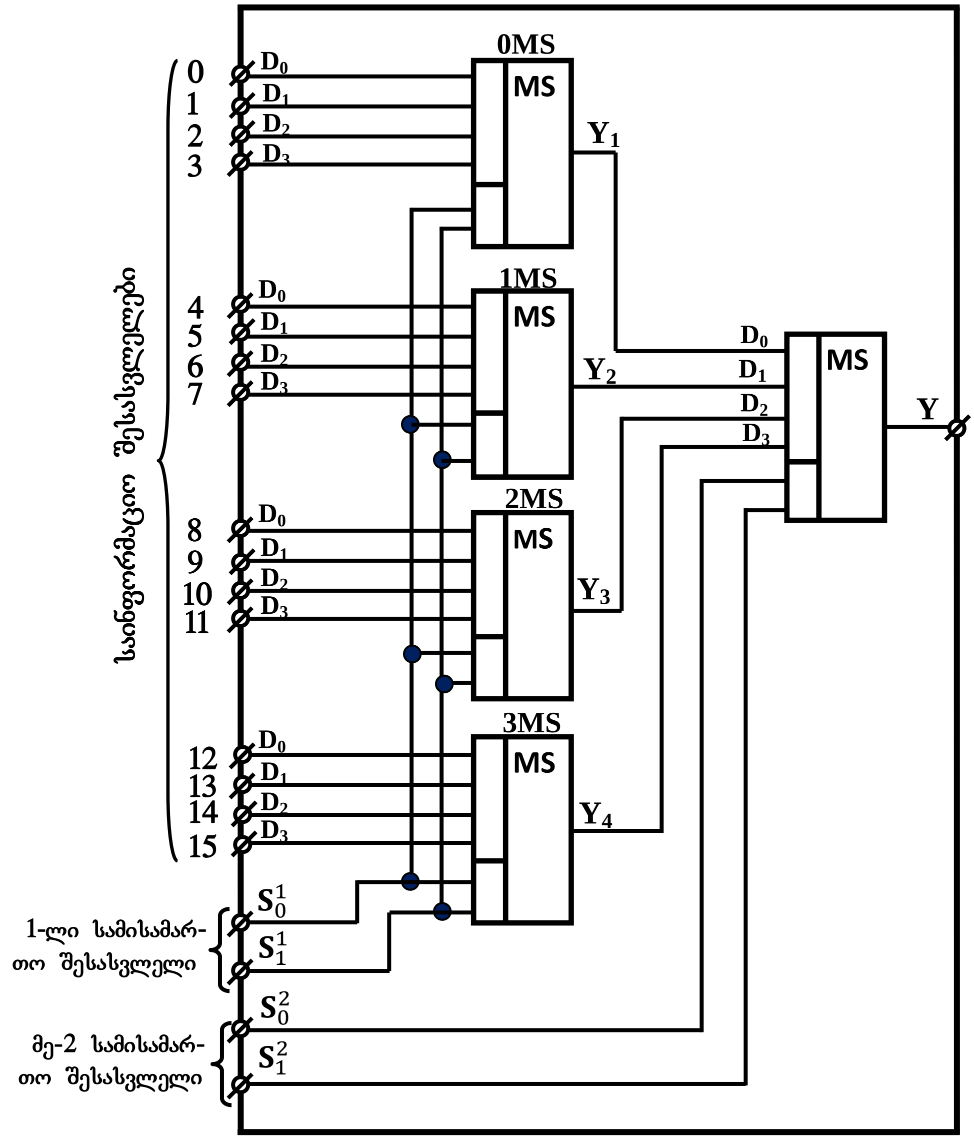
$$\mathbf{Y} = \overline{S_1}\,\overline{S_0}\, D_0 + \overline{S_1} S_0\, D_1 + S_1 \overline{S_0}\, D_2 + S_1 S_0\, D_3\,, \qquad (5.16)$$

სადაც პლუსის ნიშნებით დიზიუნქციის ლოგიკური ოპერაცია, ხოლო გამრავლების ნიშნებით – კონიუნქციის ლოგიკური ოპერაციებია აღნიშნული.

ცხრ. 5.5. n=4 რაოდენობის საინფორმაციო შესას-ვლელებიანი მულტიპლექსორის ფუნქციონირებ-მაილუსტრირებელი ცხრილი

მისამართები

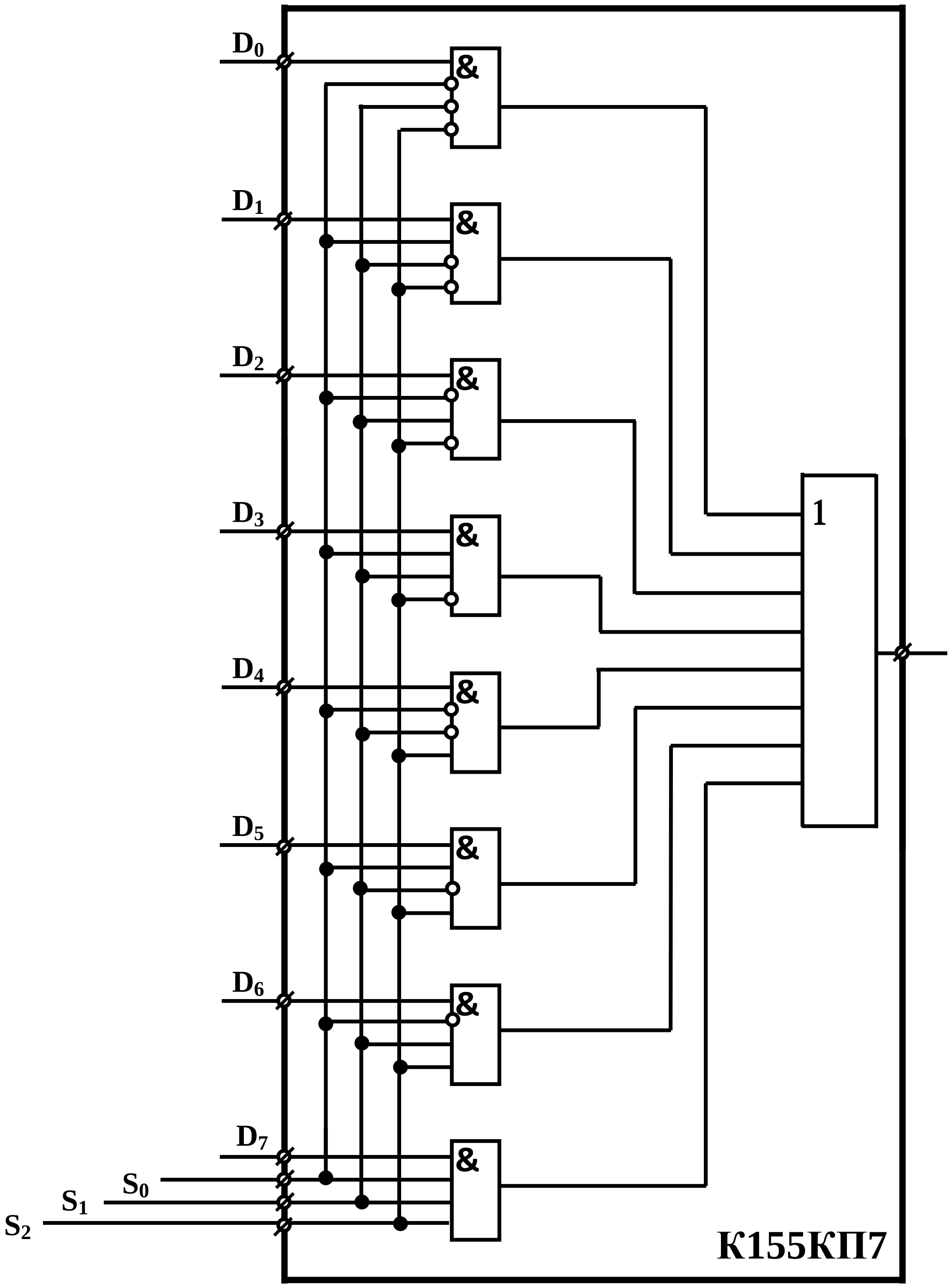
სამისამართო შესასვლელები		მისამართით გან-სანსაზღვრული i ინდექსი	**Y**-თან მიერ-თებული $\mathbf{D}_i$ შესასვლელი
S_1	S_0		
0	0	0	D_0
0	1	1	D_1
1	0	2	D_2
1	1	3	D_3



ნახ. 5.17. მულტიპლექსორული ხე

ზემოთ განვიხილეთ ოთხი საინფორმაციო შესასვლელის მქონე მულტიპლექ-სორი. უფრო მეტი რაოდენობის საინფორმაციო შესასვლელების მქონე მულ-ტიპლექსორის მისაღებად აიგება **მულტიპლექსორული ხე**. **5.17** ნახაზზე ნაჩვენებია **4** საინფორმაციო შესასვლელის მქონე **0MS, 1MS, 2MS, 3MS** მულტი-პლექსორისაგან აგებული მულტიპლექსორული ხე, რომელიც წარმოადგენს **16** საინ-ფორმაციო შესასვლელის მქონე მულტიპლექსორს. გამოსასვლელზე მისაერთებელი შესა-სვლელი **0, 1, ..., 15** არხიდან ერთ-ერთი არხის ამოსარჩევად გამოიყენება ორი სამისა-მართო შესასვლელი; **1**-ლი სამისამართო შესასვლელით **0MS, 1MS, 2MS, 3MS** მულ-ტიპლექსორებიდან ამოირჩევა ის მულტიპლექსორი, რომელთანაც მიერთებულია ამოსარ-ჩევი არხი, ხოლო მე-**2** სამისამართო შესასვლელის საშუალებით ამორჩეული მულტი-პლექსორის საინფორმაციო შესასვლელებთან მიერთებული არხებიდან ამოირჩევა საძებნი არხი.

მაგალითად, დავუშვათ, რომ საჭიროა მულტიპლექსორის გამოსასვლელს მიუერთდეს მე-**11** შესასვლელი არხი. როგორც **5.17** ნახაზიდან ჩანს, აღნიშნული არხი დაკავშირებუ-ლია **2MS** მულტიპლექსორის **D3** საინფორმაციო შესასვლელთან.



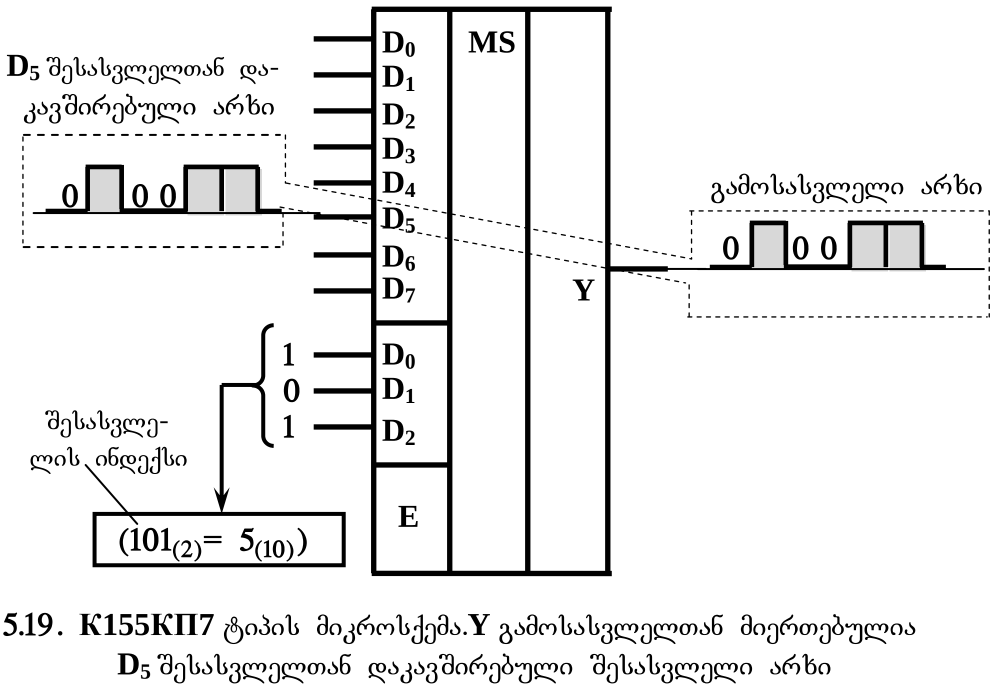
ნახ. 5.18. K155КП7 ტიპის მიკროსქემის შინაგანი სტრუქტურა

ზემოთ აღნიშნული ამოცანის გადასაწყვეტად **1-ლ** სამისამართო შესასვლელზე უნდა მივაწოდოთ ორობითი მისამართი **10, რის შედეგადაც ამოირჩევა 2MS** მულტიპლექსორი; შემდეგ მე-**2** საინფორმაციო შესასვლელზე უნდა მივაწოდოთ ორობითი **11** მისამართი, რის შედეგადაც ამოირჩევა **2MS** მულტიპლექსორის $\mathbf{D_3}$ საინფორმაციო შესასვლელი და იგი მიუერთდება **16** შესასვლელიან მულტიპლექსორის **Y** გამოსასვლელს. ვინაიდან $\mathbf{D_3}$ საინფორმაციო შესასვლელი საძებნ მე-**11** არხთანაა დაკავშირებული, ამიტომ ზემოთ დასმული ამოცანა წარმატებით იქნება გადაწყვეტილი. საბოლოოდ შეიძლება ვთქვათ, რომ **მე-11** არხის ჯამური ორობითი მისამართი კონკრეტული განხილული შემთხვევის დრო **1011**-ის ტოლია.

6 მოცემული პარაგრაფის მე-3 ქვეპუნქტის დასაწყისში აღვნიშნეთ, რომ მულტი-პლექსორი შეიძლება რეალიზებული იქნეს როგორც აპარატურულად ასევე პროგრამულადაც.

მულტიპლექსორის აპარატურულად რეალიზებისათვის სერიულად გამოდის სხვადა-სხვა ტიპის მიკროსქემები. ერთ-ერთი ასეთია **К155КП7** ტიპის მიკროსქემა, რომელიც წარმოადგენს **8** საინფორმაციო არხის მქონე მულტიპლექსორს. მისი შინაგანი სტრუქტუ-რა **5.18** ნახაზზე, ხოლო პირობითი გამოსახულება – **5.19** ნახაზზეა ნაჩვენები. მოცემულ ნახაზზე ნაჩვენებია შემთხვევა, როდესაც სამისამართო შესასვლელზე მიწოდებულია ორ-ობითი მისამართი **101;** ამიტომ მულტიპლექსორის მიერ **გამოსასვლელ არხთან** მიერთებუ-ლია საინფორმაციო $\mathbf{D_5}$ არხთან დაკავშირებული **შესასვლელი არხი** $(\mathbf{101_{(2)}= 5_{(10)}})$.

7 გარდა კავშირგაბმულობის არხების შემჭიდროებისა, მულტიპლექსორი შე-იძლება წარმატებით გამოვიყენოთ უნივერსალურ დამპროგრამებელ ელემენ-ტადაც, რომლის მეშვეობითაც შევძლებთ მოვახდინოთ ლოგიკური ფუნქციების რეალიზ-ება. მისი არსის გასაგებად ვისარგებლოთ **ინდუქციის მეთოდით**, რომლის დროსაც კერძო ფაქტებიდან, ცალკეული დებულებებიდან ხდება ზოგადი დასკვნის გამოყვანა (მისი საპი-რისპიროა **დედუქციური მეთოდი**, რომლის დროსაც ზოგადი დებულებიდან მიიღება კერძო დასკვნა). ამისათვის განვიხილოთ **5.20,ა** ნახაზზე მოყვანილი ჭეშმარიტების ცხრილის შესაბამისი სამ ცვლადზე დამოკიდებული ლოგიკური $\boldsymbol{y = f(x_0\ x_1, x_2)}$ ფუნქციის რეალიზება **8** საინფორმაციო შესასვლელის მქონე მულტიპლექსორის (**ნახ.5.20,ბ**) საშუალებით. ამ დროს გამოყენებული მეთოდიკა სამართლიანი იქნება მულტიპლექსორის დახმარებით ნე-ბისმიერ ცვლადზე დამოკიდებული ლოგიკური ფუნქციის რეალიზებისათვისაც, ოღონდ ამ შემთხვევაში განხილული ლოგიკური ფუნქციის არგუმენტების მნიშვნელობათა ნაკრებე-ბის რაოდენობა არ უნდა აღემატებოდეს აღნიშნული ფუნქციიის რეალიზებისათვის გამო-ყენებული მულტიპლექსორის საინფორმაციო შესასვლელების რაოდენობას.



ნახ. 5.19. **К155КП7** ტიპის მიკროსქემა.**Y** გამოსასვლელთან მიერთებულია $\mathbf{D_5}$ შესასვლელთან დაკავშირებული შესასვლელი არხი

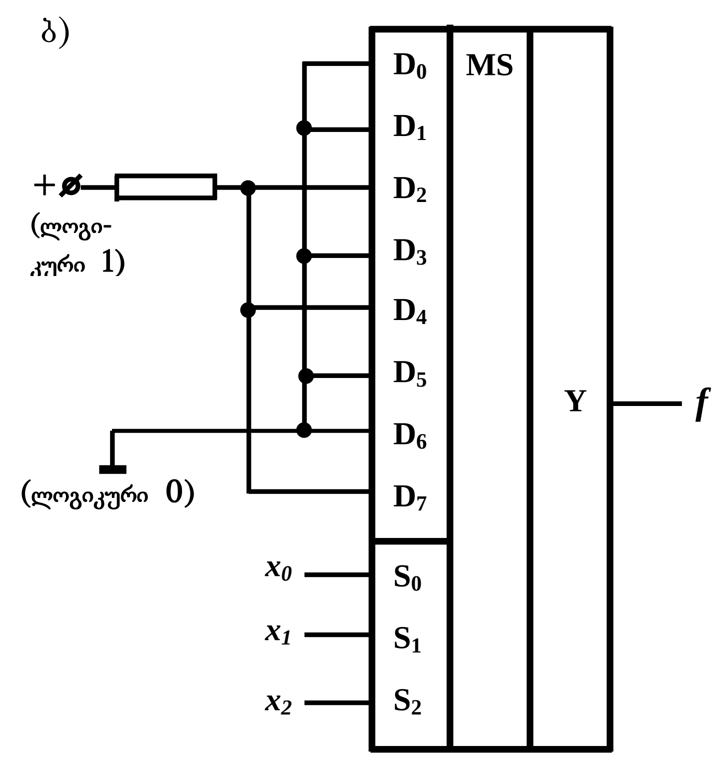
ლოგიკური $\boldsymbol{y = f(x_0, x_1, x_2)}$ ფუნქციის არგუმენტების მნიშვნელობათა ნაკრებები და-ნომრილია ათობითი $\boldsymbol{i\in(0,1,\ \dots\ ,\ 7)}$ რიცხვებით (**ნახ.5.20,ა**), რომლებიც წარმოადგენს მულტიპლექსორის საინფორმაციო $\mathbf{D}_i$ შესასვლელების ინდექსებს.

ლოგიკური ფუნქციის არგუმენტების მნიშვნელობათა იმ $\boldsymbol{i}$ ნაკრებებს, რომლებზედაც ფუნქცია იღებს ლოგიკური **1**-ის ტოლ მნიშვნელობებს, **ერთეულოვანი ნაკრებები** ვუწო-

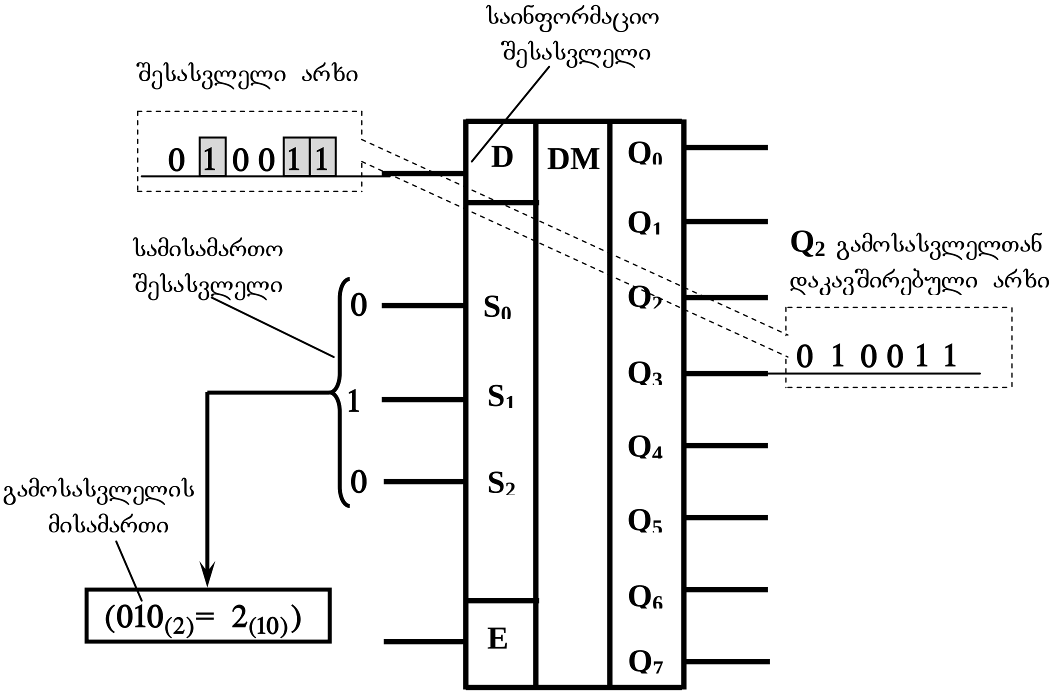
ა)

i	x_0	x_1	x_2	f
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

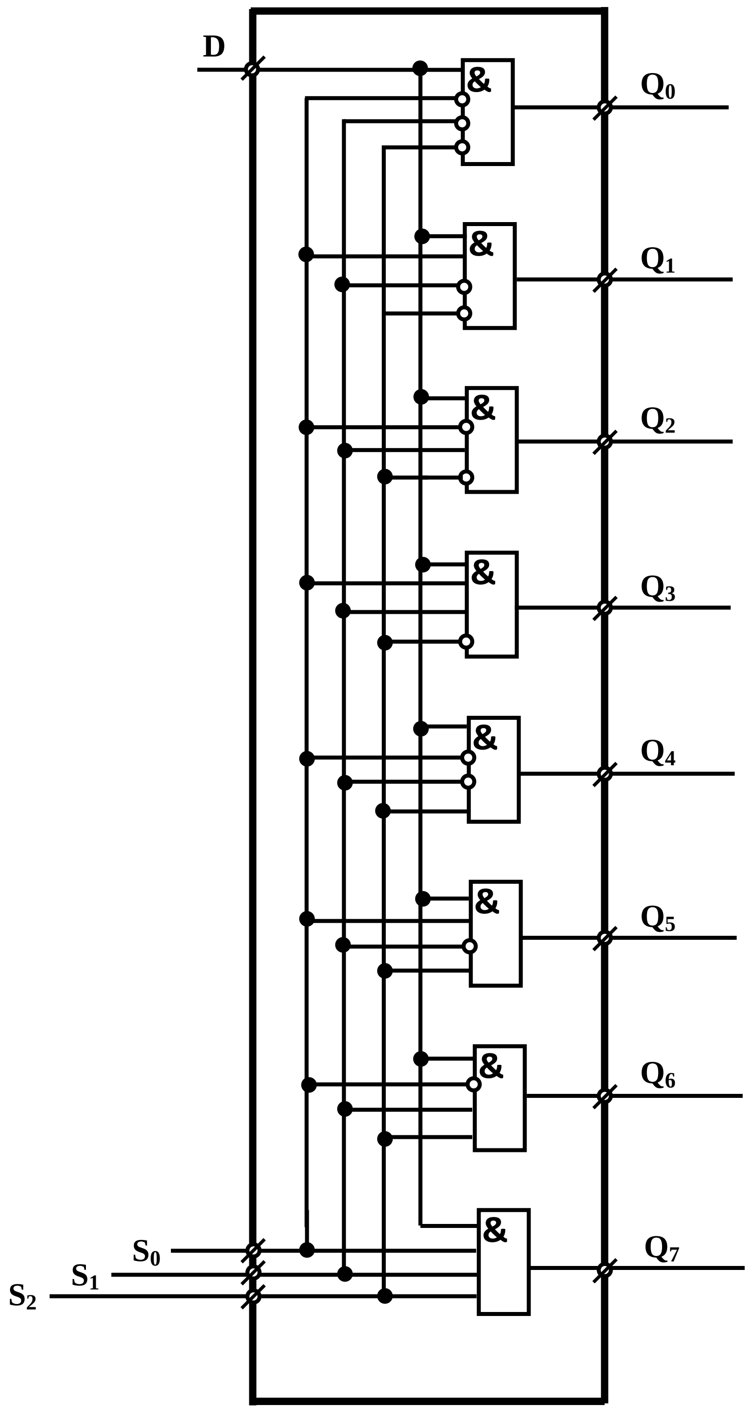
ბ)



ნახ.5.20. ლოგიკური ფუნქციის რეალიზება მულტიპლექსორის გამოყენებით:
ა) ცხრილური ფორმით წარმოდგენილი ლოგიკური ფუნქცია;
ბ) ცხრილური ფორმით წარმოდგენილი ლოგიკური ფუნქციის რე-ალიზება მულტიპლექსორის მეშვეობით.



ნახ. 5.21. დემულტიპლექსორი. შესასვლელი არხი მიერთებულია Q_2 გამოსასვლელთან დაკავშირებულ არხთან



ნახ. 5.22. დემულტიპლექსორის სქემა

დოთ, ხოლო რომლებზედაც ფუნქცია იღებს ლოგიკური **0**-ის ტოლ მნიშვნელობებს – **ნულოვანი ნაკრებები.**

ერთეულოვანი ნაკრებების ნომრების ქვესიმრავლე აღვნიშნოთ $I^{(1)}$, ხოლო ნულოვანი ნაკრებების ნომრების სიმრავლე – $I^{(0)}$ სიმბოლოთი. **5.20,ა** ნახაზზე ნაჩვენები ლოგიკური ფუნქციის შემთხვევაში $I^{(1)}$ **={2;4;7},** ხოლო $I^{(1)}$ **={0;1;3,6}.**

მულტიპლექსორის საინფორმაციო D_i **,** $i \in I^{(1)}$**={2;4;7}** შესასვლელებზე თუ მივაერთებთ კვების წყაროს (ლოგიკური 1-ის ტოლ სიგნალს), $i \in I^{(1)}$**={0;1;3;6}** შესასვლელებზე – "მიწას" (ლოგიკური 0-ის ტოლ სიგნალს), ხოლო სამისამართო შესასვლელებზე – ლო-

გიკურ x_0, x_1, x_2 ცვლადებს, მაშინ აღნიშნული მულტიპლექსორი მოახდენს განხილული ლოგიკური $y = f(x_0, x_1, x_2)$ ფუნქციის რეალიზებას.

დემულტიპლექსორი (**ნახ.5.21**) წყვეტს მულტიპლექსორის შებრუნებულ ამოცანას – მიმდევრობით მონაცემებს (ინფორმაციას) ერთადერთი საინფორმაციო **D** შესასვლელთან დაკავშირებული არხიდან გადასცემს რამდენიმე $Q_0, Q_1, \dots, Q_7$ გამოსასვლელთან დაკავშირებული არხებიდან იმ ერთადერთ Q_i არხს, რომლის i ინდექსის შესაბამისი ორობითი რიცხვი მიწოდებულია სამისამართო Q_2, Q_1, Q_0 შესასვლელებზე.

5.21 ნახაზზე გამოსახულია შემთხვევა, როდესაც სამისამართო S_2, S_1, S_0 შესასვლელებზე მიწოდებულია $i = 101_{(2)} = 2_{(10)}$ რიცხვი; ამიტომ საინფორმაციო **D** შესასვლელთან დაკავშირებულ ერთადერთი **შესასვლელი არხიდან** მიმდევრობითი 010011 მონაცემები (ინფორმაცია) გადაეწოდება Q_2 გამოსასვლელთან დაკავშირებულ **გამოსასვლელ არხს**.

განხილული დემულტიპლექსორის შინაგანი სტრუქტურა 5.22 ნახაზზეა ნაჩვენები.

მულტიპლექსორის ანალოგიურად დემულტიპლექსორიც შეიძლება იქნეს რეალიზებული როგორც პროგრამულად, ასევე აპარატურულადაც. ზემოთ განხილული დემულტიპლექსორის აპარატურულად რეალიზებისათვის, მაგალითად, შეიძლება **К155ИД7** ტიპის მიკროსქემა გამოვიყენოთ.

გამოყენებული ლიტერატურა

1. **დოლიძე თ.** ციფრული გამომთვლელი მანქანები: არითმეტიკულ-ლოგიკური საფუძვლები და ელემენტთა სისტემა: **სპი**-ს გამოცემა. **1985.** -112 გვ.

2. **დუნდუა ა. საპოჟნიკოვი ვ., საპოჟნიკოვი ვლ.** დისკრეტულ მოწყობილობათა თეორიის საკითხები. – თბ.: საქართველოს ტექნიკური უნივერსიტეტი, **1990** წ. – 119 გვ.

3. **დუნდუა ა.** კომპიუტერი, მეცნიერება და აზროვნების ლინგვისტურ-ფილოსოფიური ასპექტები. – "მეცნიერება და ტექნიკა", **1995**, №540. – გვ. 69 – 82.

4. **დუნდუა ა.** კიბერნეტიკა ადამიანის გონის ფრაქტალური ბუნების შინაგანი ლოგიკის შესახებ – "მეცნიერება და ტექნიკა", **1996**, №541. – გვ. 37 – 43.

5. **დუნდუა ა.** პერსონალურ კომპიუტერებში გამოყენებული **32**-თანრიგიანი მიკროპროცესორების არქიტექტურის ევოლუციის საკითხები. – "მეცნიერება და ტექნოლოგიები", **2002** №7-9. – გვ. 16 – 21.

6. **მაჭარაძე თ., წვერაიძე ზ.** კომპიუტერები და კომპიუტერული ტექნოლოგიები. – თბ.: "ტექნიკური უნივერსიტეტი", **2009** წ. – 363 გვ.

7. **Акулов О.А., Медведев Н.В.** Информатика: базовый курс. М.: «Омега–Л», **2009**. -574 с.

8. Введение в микроЭВМ / **Майоров С.А., Кириллов В.В., Приблуда А.А.** Л.: «Машиностроение», **1988.** -304 с.

9. **Гивоне Д., Россер Р.** Микропроцессор и микрокомпьютеры. М.: Мир, **1983**. – 464 с.

10. **Голдсорт Б.** Проектирование цифровых логических устройств. – М.: Машиностроение. **1985**. -322 с.

11. **Дергачова Л.М.** Решение типовых экзаменационных задач по информатике – М.: «БИНОМ. Лаборатирия знаний, **2012** . -360 с.

12. **Дмитриев В.И.** Прикладная теория информации. М.: «Высшая школа», **1989** - 332 с.

13. **Дундуа А.А.** Формализация некоторых этапов программной реализации поледовательностных схем на микропроцессорах / Проблемы разработки, внедрения и эксплуатации микроэлек-тронных систем железнодорожной автоматики и телемеханики: сб. науч. тр. – СПб.: Петербургский государственный университет путей сообщения, **2005**. Стр. 88 – 98.

14. **Дундуа А.А.** Координатный метод перехода от языка заказчика к языку разработчика цифро-вых систем управления / Автоматика и телемеханика железных дорог России. Техника, технология, серти-фикация: сб. науч. тр. – СПб.: Петербургский государственный университет путей сообщения, **2008**. Стр. 51 – 58.

15. **Ершова Э.Б., Рогинский В.Н., Маркин И.П.** Основы дискретной автоматики в электросвязи. – М.: Связь, **1980.** -238 с.

16. **Калабеков Б.А., Мамзалов И.А.** цифровые устройства и микропроцессорные системы. М.: Радио и связь, **1987**. – 400 с.

17. **Колдуэл С.** Логический синтез релейных устройств. – М.: Изд. Иностранной литературы, **1962. -737 с.**

18. **Кузнецова О.С**. Краткий курс по информатике. - М.: «Окей-книга», 2011. – 174 с.

19. Информатика и ИКТ. Подготовка к ЕГЕ 2011. Типрвые задачи/под ред. Проф. **Макаровой Н.В**. – СПб.: Питер, 2011. -464 с.

20. **Новожилов О.П.** Информатика . М.: «Юрайт», **2012**. -564 с.

21. Информатика **/** Под ред. **Симоновича С.В.** – СПб.:Питер, **2010.** – 640 с.

22. Информатика **/ Соболь Б.В., Галин А.Б., Панов Ю.В., Рашидова Е.В., Садовой Н.Н. –** Ростов-на Дону: «Феникс», **2010**. – 446 с.

23. Основы информатики**/ Ляхович В.Ф., Крамаров С.О., Шамараков И.П. –** Ростов н/Д: «Феникс», **2010.** -715 с.

24. **Кирилов В.И., Старченко А.А.** Логика. – М.: «Юрист», **1999**. -256 с.

25. **Клини С.К.** Представление событий в нервных сетях и конечных автоматах. - Сб. «Автоматы» под редакцией **Шеннона К. Э.** и **Макарти Дж.**, М.: ИЛ, **1956**. Стр. 15-67.

26. **Мак-каллок Уорен С., Питтс В. -** Сб. «Автоматы» под редакцией **Шеннона К. Э.** и **Макарти Дж.**, М.: ИЛ, **1956**. Стр. 362 - 384.

27. **Сапожников В.В., Кравцов Ю., Сапожников Вл.В.** Дискретные устройства железнодорожной автоматики, телемеханики и связи. М.: Транспорт, **1988.** – 255 с.

28. **Сапожников В.В., Сапожников Вл.В., Борисенко Л.И.** Какими должны быть микропроцессорные системы железнодорожной автоматики и телемеханики**,** «Автоматика, телемеханика и связь», **1988,** №5, с. 32-34 .

29**. Сапожников В.Вл., Кононов В.А., Куренков С.А., Лыков А.А., Наседкин О.А., Никитин А.Б., Прокофьев А.А., Трясов М.С.** Микропроцессорные системы централизации - ГОУ «Учебно-методический центр по образованию на железнодорожном транспорте», ***2008.*** – 398 стр.

30. **Степанов А.Н.** Информатика. СПб.: Питер, **2010** – 720 с.

31. **Стрельников А.И., Стрельникова Л.Л.** Беседы о неизвестном. – Амрита. 2012. -288 с.

32. **Федотова Е.Л., Федотов А.А.** Информатика. Курс лекций. –М.: ИД «Форум»: ИНФРА-М. **2011**. – 480 с.

33. Стахов А.П. -Троичный принцип Брусенцова, система счисления Бергмана и «золотая» троичная зеркально-симметричная арифметика-
- **http://www.trinitas.ru/rus/doc/0232/003a/02320001.htm.**

34. **Стахов Алексей**. Роль системы счисления в истории компьютеров. **- http://baxrefer.ru/07/dok.php?id=rk253.**

35. **Стахов Алексей**. Система счисления Бергмана и новые свойства натуральных чисел; **- http://www.trinitas.ru/rus/doc/0232/004a/02321068.htm.**

36. **Таненбаум Э., Остин Т.** Архитектура компьютераю 6-е изд. – СПб.: Питер, **2013, -816 стр.**

37. **Хлебников А.А.** Информационные технологии. М.: КНОРУС, **2014.** – 472 с.

38. **http://school497.ru/download/u/02/les10/les.html.**

39. **http://consumer.nm.ru/charhist.htm.**

40. **http://book.kbsu.ru/theory/chapter4/1_4_13.html.**

41 **http://www.3dmax.ru/modeling/lessons/292.html (**Правила "Орнаментостроения", часть 1)

42. **http://habrahabr.ru/post/112953/.**

43. **http://www.csu.ac.ru/~yan/informat/tutor1/flp_files/nflp_1.htm.**

44. **http://kuzelenkov.narod.ru/mati/book/inform/inform5.html.**

45. **http://more-it.ru/view_post.php?id=12**.

46. **http://www.rusedu.info/Article620.htmlинформации»**.

47. **http://www.5byte.ru/11/0011.php.**

48. **http://book.kbsu.ru/theory/chapter4/1_4.html.**

საგნობრივი საძიებელი

ს ა რ ჩ ე ვ ი

КРАТКОЕ СОДЕРЖАНИЕ

В работе рассмотрены следующие вопросы:

Александр Аквсентьевич Дундуа

ТЕОРЕТИЧЕСКИЕ ОСНОВЫ КОМПЬЮТЕРНЫХ СИСТЕМ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИИ

Учебное пособие
(На грузинском языке)

Редактор Н. Сухиташвили
Рецензеты: профессор К. Н. Камкамидзе
профессор М. А.Гоцадзе
Компьютерная верстка А. А. Дундуа

Издательство «Технический университет»
2014