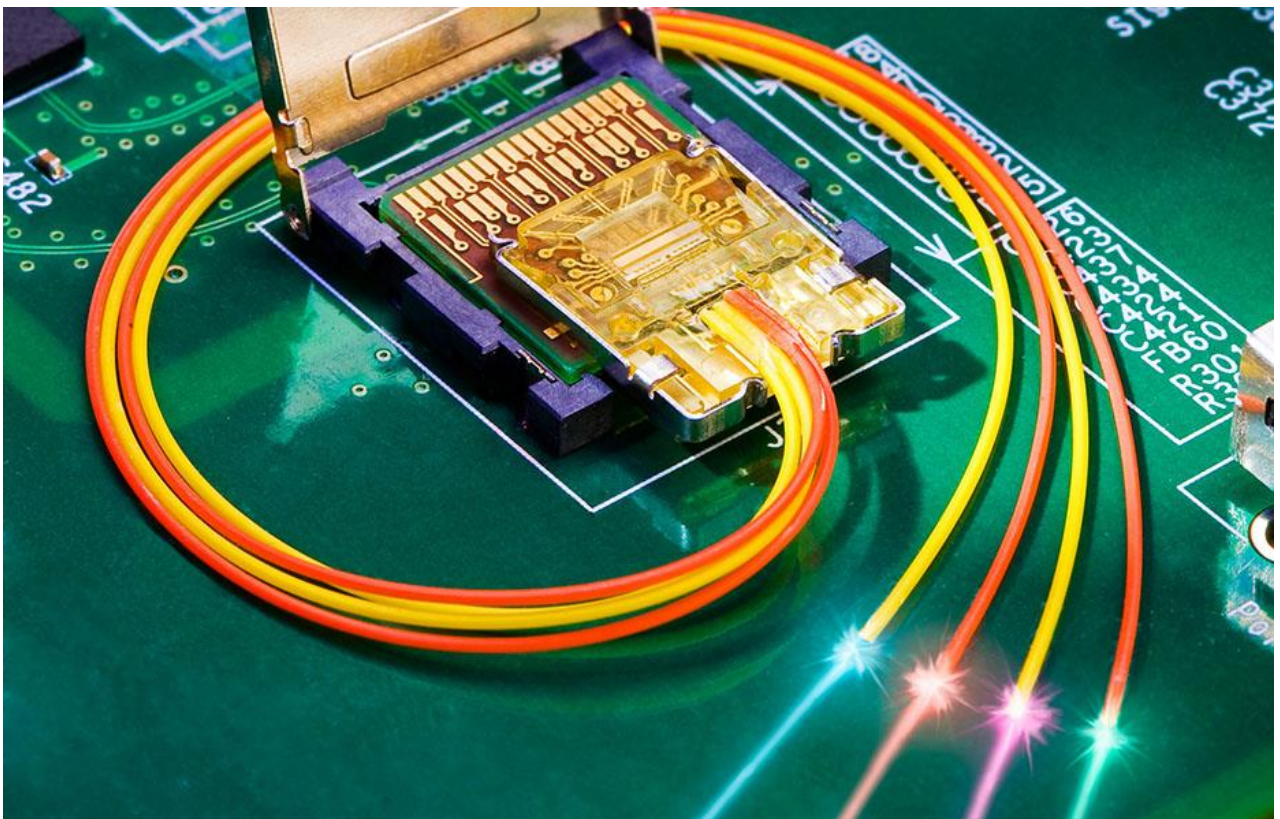


დ. ვეფხვაძე

საბოროტო კომპიუტერები

და

ინფორმაციული სისტემები



თბილისი 2023

<https://doi.org/10.52340/lc.2025.10.09>

საქართველოს საავიაციო უნივერსიტეტი

დ. ვეფხვაძე

საბორტო კომპიუტერები და ინფორმაციული
სისტემები

ლექციების კურსი

დამტკიცებულია საქართველოს საავიაციო უნივერსიტეტის
საინჟინრო ფაკულტეტის საბჭოს მიერ



თბილისი 2023

ლექციების კურსში განხილულია: საბორტო კომპიუტერების განვითარების ისტორია, დანიშნულება, სტრუქტურა, ფუნქციონირებისა და ტექნიკური რეალიზაციის ძირითადი პრინციპები, ექსპლუატაციის განსაკუთრებულობა, მუშაობისუნარიანობის კონტროლის პროგრამული და სქემური მეთოდები, პროცესორების, ოპერატიული მეხსიერების და ინტერფეისული მოწყობილობების ძირითადი მახასიათებლები და ორგანიზაციის ხერხები; თანამედროვე სპეციალიზებული საბორტო კომპიუტერები; საბორტო ინფორმაციულ სისტემების მაგისტრალებში მონაცემების შეტანის, შენახვის და გამოტანის პროცედურები; საბორტო კომპიუტერების და სისტემების საიმედოობის ამაღლების სტრუქტურული, ინფორმაციული მეთოდები და საშუალებები; მათი განვითარების პერსპექტივები და ტენდენციები.

ლექციების კურსი განკუთვნილია საქართველოს საავიაციო უნივერსიტეტის საინჟინრო სპეციალობის ბაკალავრიატისა და მაგისტრატურის სტუდენტებისთვის.

გამოდის პროფესორ სერგო ტეფნაძის რედაქციით

საქართველოს საავიაციო უნივერსიტეტი, 2023

შესავალი

დღეს წარმოდგენილია თანამედროვე საჰაერო ხომალდების ფრენის მართვის თუ კონტროლის ამოცანების ოპტიმალური გადაწყვეტა ფრენის მართვის ავტომატურ და ავტომატიზებულ სისტემებში საბორტო კომპიუტერების გამოყენების გარეშე.

საბორტო კომპიუტერები როგორც მმართველი კომპიუტერები ფრენის მართვის კონკრეტული ამოცანის გადაწყვეტისთვის ახდენენ აუცილებელი საფრენოსნო ინფორმაციის შეკრებას, დამუშავებას და მმართველი სიგნალების გამომუშავებას. საბორტო კომპიუტერების გამოყენებამ შესაძლებელი გახადა რეალიზებულ ყოფილიყო ფრენის მართვის ოპტიმიზაციის ალგორითმები, ამაღლებულიყო საჰაერო ხომალდების საბორტო სისტემების საიმედოობა და გაზრდილიყო ზოგადად საჰაერო ხომალდების ფრენის უსაფრთხოება. ეს არასრული ჩამონათვალიც კი საკმარისია იმისათვის, რომ ავღნიშნოთ, თუ რაოდენ დიდი ყურადღება უნდა მიექცეს დღეს საბორტო კომპიუტერების დამუშავებას, შესწავლას, აგებას და გამოყენებას.

წინამდებარე სალექციო კურსის ამოცანას წარმოადგენს სტუდენტებს შეასწავლოს: საბორტო კომპიუტერების აგების მათემატიკური და სქემოტექნიკური საფუძვლები, ელემენტური ბაზა, დანიშნულება, სტრუქტურა, ფუნქციონირებისა და ტექნიკური რეალიზაციის ძირითადი პრინციპები, ექსპლუატაციის განსაკუთრებულობა; საბორტო კომპიუტერების მუშაობისუნარიანობის კონტროლის პროგრამული და სქემური მეთოდები; პროცესორების, ოპერატიული მეხსიერების და ინტერფეისული მოწყობილობების ძირითადი მახასიათებლები და ორგანიზაციის ხერხები; ექსპლუატაციის ხისტ პირობებში მომუშავე თანამედროვე სპეციალიზებული საბორტო კომპიუტერები; ციფრული ინფორმაციულ-გამოთვლითი-მმართველი სისტემების სტრუქტურული სქემები და მუშაობის პრინციპები; საბორტო ინფორმაციულ სისტემების მაგისტრალებში მონაცემების შეტანის, შენახვის და გამოტანის პროცედურები; საბორტო კომპიუტერების და სისტემების საიმედოობის ამაღლების სტრუქტურული, ინფორმაციული მეთოდები და საშუალებები; საბორტო კომპიუტერების განვითარების პერსპექტივები და ტენდენციები (ოპტიკური, კვანტური, ნეიროკომპიუტერები).

ლექცია 1

საბორტო კომპიუტერების განვითარების ისტორია, დანიშნულება, შემადგენლობა, მუშაობის პრინციპი, სტრუქტურა და ფუნქციონირება

1.1. საბორტო კომპიუტერების განვითარების ისტორია

საბორტო მოწყობილობებში ციფრული გამოთვლითი საშუალებები გამოჩნდა გასული საუკუნის 60-იანი წლების ზღურბლზე და პრაქტიკულად ძალზე მცირე დროის განმავლობაში მათ შეცვალეს ადრე საბორტო მოწყობილობებში გამოყენებული ანალოგური გამომთვლელები, რითაც მიღწეულ იქნა ამოცანების გადაწყვეტის უფრო მაღალი სიზუსტე, ისინი ხასიათდებოდნენ გამოყენების დიდი უნივერსალურობით და გააჩნდათ ფართო ლოგიკური შესაძლებლობები.

საბორტო ციფრული გამოთვლითი მანქანის (სცგმ), იგივე საბორტო კომპიუტერის ეს თვისებები საშუალებას იძლევა გამოვიყენოთ ისინი საჰაერო ხომალდის საბორტო მოწყობილობების ყველა ქვესისტემაში, რითაც მიიღწევა მათი მდგრადობა მართვის ალგორითმების გართულებასთან მიმართებაში და საშუალებას იძლევა გამოყენებულ იქნეს უფრო რთული ე.ი საჰაერო ხომალდის და მისი ქვესისტემების მართვის უფრო სრულყოფილი ალგორითმები. მათ გამოყენებამ შესაძლებელი გახდა განხორციელებულიყო ინფორმაციული ურთიერთქმედება საბორტო მოწყობილობის ცალკეულ ქვესისტემებს შორის (ადრე ასეთი ურთიერთქმედება უშუალოდ შეუძლებელი იყო) და შესაძლებელი გახდა შექმნილიყო საბორტო მოწყობილობის ერთიანი კომპლექსი (სმეკ), რაც საბოლოო ჯამში იძლევა საშუალებას ამაღლდეს საფრენოსნო დავალების შესრულების ეფექტურობა და ფრენის უსაფრთხოება.

საბორტო კომპიუტერების გამოყენებამ მოითხოვა თვითმფრინავის საბორტო რადიოელექტრონული მოწყობილობის განსაზღვრული უნიფიკაცია, რის შედეგადაც შესაძლებელი გახდა საბორტო მოწყობილობის დამუშავების ვადების შემცირება, ასევე შემცირდა დანახარჯები მის დამუშავებაზე, მოდერნიზაციაზე და ექსპლუატაციაზე.

საჰაერო ხომალდების საბორტო მოწყობილობის შემადგენლობაში ციფრული გამოთვლითი ტექნიკის საშუალებების ეფექტური გამოყენებისათვის აუცილებელი იყო გადაჭრილიყო შემდეგი პრობლემები:

- დამუშავებულიყო საბორტო კომპიუტერები, რომლებიც დააკმაყოფილებდნენ მაქსიმალურად კონკრეტული გამოყენების შემთხვევაში მოთხოვნილ ყველა პარამეტრს;

- შექმნილიყო ინფორმაციის გაცვლის სისტემები რომლებსაც შეეძლებოდათ საბორტო აპარატურასთან (გადამწოდებთან და შემსრულებელ ორგანოებთან) არა მარტო შეუღლება, არამედ ასევე, თვით საბორტო კომპიუტერებს შორის ურთიერთქმედების განხორციელება საბორტო გამოთვლითი სისტემების შემადგენლობაში;

- დამუშავებულიყო პროგრამული უზრუნველყოფა, რომელშიც ასახული უნდა ყოფილიყო პროგრამების შექმნისა და გამართვის საშუალებები საბორტო კომპიუტერის გამოთვლითი პროცესის და ფუნქციონირების უზრუნველყოფით.

პირველი ორი პრობლემის გადაწყვეტის აუცილებლობა განსაზღვრული იყო თავიდანვე, ხოლო რაც შეეხება მესამე პრობლემას - პროგრამული უზრუნველყოფის შექმნას, თითქოს ის იყო მეორეხარისხოვანი ამოცანა. ამიტომ ციფრული ავიონიკის განვითარების ადრეულ სტადიებზე ძირითადი ყურადღება გამახვილებული იყო საბორტო კომპიუტერებისა და მისი საბორტო აპარატურასთან შეუღლების საშუალებების დამუშავებაზე. პროგრამული უზრუნველყოფის შექმნის პრობლემა რთულდებოდა როგორც საბორტო კომპიუტერის სტრუქტურის გართულების, გადასაწყვეტი ამოცანების წრის გაფართოებით, ასევე საბორტო გამოთვლითი სისტემების წარმოშობისა და განვითარების გამო. ამჟამად დანახარჯები პროგრამული უზრუნველყოფის დამუშავებაზე მნიშვნელოვნად აჭარბებს დანახარჯებს - გაწეულს აპარატული საშუალებების დამუშავებასა და შექმნაზე.

საბორტო მოწყობილობებისთვის ციფრული გამოთვლითი ტექნიკის საშუალებების დამუშავება დაიწყო გასული საუკუნის 50-იანი წლების მეორე ნახევარში. ამ პერიოდიდან დაიწყო ციფრული გამოთვლითი მოწყობილობების დამუშავება ხომალდების საბორტო მოწყობილობებისთვის, ხოლო შემდეგ რაკეტებისთვისაც. 60-იანი წლების დასაწყისში შეიქმნა საბორტო ციფრული გამოთვლითი მოწყობილობები სპეციალიზებული და უნივერსალური დანიშნულებით, რის შედეგადაც დამუშავებულ იქნა მთელი რიგი საბორტო გამოთვლითი ციფრული მოწყობილობები და ავტონომიური საბორტო კომპიუტერები.

მიკროპროცესორების სერიულად გამოშვების შემდეგ დაიწყო სხვადასხვა საბორტო ციფრული გამოთვლითი მანქანების (შემდეგში საბორტო კომპიუტერების)

ფართო ფრონტით დამუშავება, თუმცა მათ (საბორტო კომპიუტერები „ინტეგრაცია“, D-175, „სიგმა“, „ალისა“ და სხვა) ვერ ჰპოვეს პრაქტიკული გამოყენება.

გასული საუკუნის 90-იან წლებში მიკროპროცესორული კომპლექტის i80386 და i80486-ის ბაზაზე შეიქმნა საბორტო კომპიუტერების სერია, რომელთა შორის შეიძლება გამოიყოს საბორტო კომპიუტერების ოჯახები სცგმ-386/486, სცგმ-მბ-5.

გასული საუკუნის 70-იან წლებიდან საბორტო კომპიუტერები გამოიყენებოდა პრაქტიკულად საბორტო მოწყობილობის ერთიანი კომპლექსის თითქმის ყველა ქვესისტემაში, თუმცა მათი დამუშავება ხდებოდა ავიაციის სხვადასხვა უწყებებსა და დაწესებულებებში, რაც გამორიცხავდა მათი ერთიანი არქიტექტურის ქონას, პროგრამული უზრუნველყოფის შემუშავების ერთიან სისტემას და საბოლოო ჯამში იწვევდა საბორტო კომპლექსების შექმნის ვადების გაზრდას, იმავდროულად მათ დამუშავებასა და ექსპლუატაციაზე გაწეული დანახარჯების ამაღლებასაც.

გასული საუკუნის 70-იან წლების ბოლოსა და 80-იანი წლების დასაწყისში ჩატარებული ანალიზის საფუძველზე დამუშავდა პროგრამა უნიფიცირებული საბორტო კომპიუტერების ოჯახის შექმნის შესახებ. ამ პროგრამის დამტკიცების შემდეგ გაიშალა ფართო მუშაობა უნიფიცირებული საბორტო კომპიუტერების შესაქმნელად. ამ პროგრამის შესაბამისად განხორციელდა მთელი რიგი სამუშაოები ასეთი საბორტო კომპიუტერების დამუშავებისათვის, თუმცა სამართლიანობა მოითხოვს აღინიშნოს, რომ ამ მიმართულებით მნიშვნელოვანი წარმატება მოიპოვა მხოლოდ „ბაგეტი“-ს ტიპის საბორტო კომპიუტერების ოჯახმა, რომლის შემადგენლობაში შედის მისი სხვადასხვა ტიპის მოდიფიკაციები.

იმავდროულად, პარალელურად მიმდინარეობდა სამუშაოები სპეციალიზებული პროცესორების შექმნისთვის სიგნალების დამუშავებისათვის. ამ მიმართულებით შექმნილ იქნა მთელი რიგი სამრეწველო პროგრამირებადი სასიგნალო პროცესორები „II 200“, „II 300“, „ბაგეტი-55-02“. ამ უკანასკნელს შეუძლია ოპერაციების შესრულება სიჩქარით 40 მილიონი ოპერაცია წამში. ამჟამად მიმდინარეობს უფრო სრულყოფილი სასიგნალო პროცესორის „II-600“-ის დამუშავება.

მთელი სამი ათეული წლის განმავლობაში საბორტო კომპიუტერები ხარისხობრივად შეიცვალნენ. მათი სწრაფმედეხა მნიშვნელოვნად გაიზარდა და მიაღწია ათობით მილიონ ოპერაციას წამში, ხოლო დამხსომეხელი მოწყობილობების

ტევადობამ 8-16 მეგაბაიტს, ამავდროულად შემცირდა მათი წონა და ენერგომოხმარება. ეს კი განპირობებული იყო საბორტო კომპიუტერების ბაზის, არქიტექტურის და სტრუქტურული ორგანიზაციის სრულყოფით.

თუ დასაწყისში საბორტო კომპიუტერების აგებისთვის გამოიყენებოდა დისკრეტული კომპონენტები, შემდგომ ელემენტური ბაზის განვითარების შედეგად ისინი შეიცვალა ჯერ ინტეგრალური სქემებით (ის), შემდეგ სქემებით საშუალო და დიდი ინტეგრაციით და ბოლოს დიდი ინტეგრალური სქემების მიკროპროცესორული კომპლექტებით.

დისკრეტული ელემენტების შეცვლამ დიდი ინტეგრალური სქემებით (დის) შესაძლებელი გახადა გაზრდილიყო საბორტო კომპიუტერების სწრაფქმედება, იმავდროულად წონისა და ენერგომოხმარების საგრძნობი შემცირებით. მიკროპროცესორების სრულყოფამ 80-იან წლებსა 90-იანი წლების დასაწყისში შესაძლებელი გახადა მნიშვნელოვნად გაზრდილიყო საბორტო კომპიუტერების სწრაფქმედება, ასევე გაუმჯობესებულიყო მათი შიგა და გარე ინტერფეისები.

პირველ საბორტო კომპიუტერებში გამოიყენებოდა არაუნიფიცრებული სალტები, რომლებიც უზრუნველყოფდნენ ურთიერთქმედებას კომპიუტერის ცალკეულ ბლოკებს შორის და გარდამქმნელები „კოდი-ანალოგი“, „ანალოგი-კოდი“ აბონენტებთან კავშირისათვის. საბორტო კომპიუტერების სტრუქტურის განვითარების შესაბამისად დასაწყისში შიგა ინტეფეისის სახით გამოიყენებოდა ინტერფეისი „საერთო სალტ“, ხოლო შემდეგ სისტემური სალტე VMF, რომელიც უკვე წარმოადგენდა ღია ინტერფეისს. ანალოგური გარე ინტერფეისი დამატებით აღჭურვილია რადიალური არხებით თანმიმდევრობითი კოდების გადაცემისთვის გატარების უნარით 48 კილობიტი/წამში, შემდეგ გარე ინტერფეისის შემადგენლობაში გამოყენებულ იქნა მულტიპლექსური არხები გატარების უნარით 1 მეგაბიტი/წამში. შემდგომში სტრუქტურის სრულყოფასთან ერთად შესაძლებელია გამოყენებულ იქნეს AS4074 ტიპის ქსელური ინტერფეისები და SCI-ის ტიპის ინტეფეისები (რეალური დროის სისტემებისათვის) გატარების უნარით 1 გიგაბიტი/წამში.

მოკლედ განვიხილოთ საბორტო კომპიუტერების განვითარების ისტორია მათი თაობების ქრონოლოგიის მიხედვით.

პირველი თაობის საბორტო კომპიუტერებისთვის დამახასიათებელი იყო შედარებით დაბალი საინფორმაციო-გამოთვლითი პოტენციალი და საკმაოდ პრიმიტიული (დღევანდელი გადასახედიდან) სტრუქტურა, რომელიც რეალიზებული იყო დისკრეტული კომპონენტების გამოყენებით. ეს საბორტო კომპიუტერები მოიცავდა დამხსომებელ მოწყობილობებს, მართვის მოწყობილობას და კვების მეორად წყაროს. კომპიუტერის მეხსიერება იყო ერთდონიანი. ამ კომპიუტერების დახურული არქიტექტურა აგებული იყო ფუნქციური ბლოკების საფუძველზე, რომლებიც დამუშავებული იყო სპეციალურად თითოეული კონკრეტული შემთხვევისათვის. მისი ნებისმიერი მოდერნიზაცია მოითხოვდა ფუნქციური ბლოკებისთვის დამატებით დამუშავებას საჭირო მახასიათებლებით. მათში დაპროგრამება ხორციელდებოდა მანქანურ კოდებში, ხოლო პროგრამების შემუშავებისთვის გამოიყენებოდა მაინტერპრეტირებელი სისტემები და კონტროლისა და ინდიკაციის პულტები.

პირველი თაობის საბორტო კომპიუტერებისთვის დამახასიათებელი იყო: სწრაფქმედება 62 ათასი ოპერაცია წამში; ოპერატიული მეხსიერების ტევადობა 256 16 თანრიგიანი სიტყვა; მუდმივი დამხსომებელი მოწყობილობა 8კილო x 16 ბიტი; მტყუნებამდე ნამუშევარი 200 სთ, მასა 330 კგ. მოთხოვნილი სიმძლავრე 2000 ვატი.

მეორე თაობის საბორტო კომპიუტერების სტრუქტურაში გამოყენებულ იქნა კონვეიერიზაციის ელემენტები, რომლებიც უზრუნველყოფდნენ ოპერაციების შესრულებაში პროცესორების ჩართვას გამრავლების, გაყოფის და ელემენტარული ფუნქციების ოპერაციების შესრულებისათვის უფრო სრულყოფილი ამჯამავებისა და სპეციალური მოწყობილობების ჩართვას. მათი სტრუქტურა რეალიზებული იყო ინტეგრალურ სქემებზე, მაგრამ მათთვის ნიშანდობლივია დეტერმინირება და ძნელად მოდერნიზების შესაძლებლობა, ე.ი სხვა სიტყვებით რომ ვთქვათ, ისინი არსებითად წარმოდგენდნენ ჩაკეტილ სისტემებს. ამ თაობის კომპიუტერებისთვის პროგრამების დასაწერად იყენებდნენ ასემბლერის დონის ენებს, ხოლო მათი დამუშავებისთვის სპეციალურ გამმართავ კომპლექსებს, რომლებიც საბორტო კომპიუტერებს აერთიანებდნენ ინსტრუმენტულ გამოთვლით მანქანასთან.

პირველი და მეორე თაობის საბორტო კომპიუტერების აბონენტებთან ურთიერთქმედებისთვის გამოიყენებოდა შეუღლების მოწყობილობა, რომელიც შეიცავდა გარდამქმნელების აუცილებელ ნაკრებს „ანალოგი - ციფრები“ და „ციფრები - ანალოგი“, რაც გამოწვეული იყო იმით, რომ საბორტო აპარატურა იყენებდა ძირითადად

ანალოგურ ინტერფეისს. შეუღლების ეს მოწყობილობა კონსტრუქციულად შესრულებული იყო ან ავტონომიური ბლოკის სახით (რომელიც ურთიერთქმედებდა კომპიუტერთან ციფრული არხის მეშვეობით) ან ინტეგრირებული იყო საბორტო კომპიუტერთან და ქმნიდა შეტანა-გამოტანის მოწყობილობას. მეორე თაობის საბორტო კომპიუტერებში თანმიმდევრობითი კოდებისა და ერთჯერადი ბრძანებების გაცვლისთვის გამოიყენება არხები ARINC-429.

მეორე თაობის საბორტო კომპიუტერების ტიპური წარმომადგენლები არიან საბორტო კომპიუტერები: „ორბიტა-10“ „ორბიტა-20“ „არგონ-15“ და მისი მოდიფიკაციები. მეორე თაობის კომპიუტერებისთვის ძირითადად დამახასიათებელია შემდეგი ტექნიკური მახასიათებლები: სწრაფქმედება 500 ათასი ოპერაცია წამში, მეხსიერების საერთო ტევადობა 80 კილო სიტყვა (ოპერატიული მეხსიერება 5 კილო, მუდმივი მეხსიერება 80 კილო, ენერგოდამოუკიდებელი დამხსომებელი მოწყობილობა 1 კილოსიტყვა), მტყუნებამდე ნამუშევარი 5000 საათი, წონა 16,6 კგ.

მესამე თაობის საბორტო კომპიუტერებში გამოყენებულია იერარქიული მეხსიერება, რომელიც შეიცავს ზეოპერატიულ მეხსიერებას, წყვეტის მრავალდონიან სისტემას, მეხსიერებასთან პირდაპირი დაშვების არხებს, ასევე ინფორმაციის დაცვის მექანიზმებს არასანქცირებული დაშვებისაგან. მესამე თაობის საბორტო კომპიუტერის სტრუქტურაში გამოყენება ჰპოვა მულტიპროცესორების მხარდაჭერის საშუალებებმა. ამ თაობის საბორტო კომპიუტერებს ძირითადად გააჩნიათ მაგისტრალურ-მოდულური ორგანიზაცია და მათთვის დასაშვებია თავიანთი მახასიათებლების ფართო საზღვრებში ცვლილება შესაბამისი მოდულების აუცილებელი რაოდენობის გამოყენების გზით, რაც იმას ნიშნავს რომ მათთვის დამახასიათებელია გარკვეული გახსნილობა. უნიფიცირებული მოდულების აგებისთვის გამოიყენება სქემები ინტეგრაციის საშუალო და მაღალი დონეებით, ასევე დიდი ინტეგრალური სქემების მიკროპროცესორული კომპლექტებით. რაც შეეხება ინფორმაციის გაცვლას, გარე ინტერფეისის შემადგენლობაში გარდა გარდამქმნელებისა „ანალოგი-ციფრი“ და „ციფრი-ანალოგი“ გამოიყენება ერთჯერადი ბრძანებები, თანმიმდევრობითი კოდები და მულტიპლექსური არხები დადგენილი სტანდარტების შესაბამისად.

მესამე თაობის საბორტო კომპიუტერების ტექნიკური და პროგრამირებადი უზრუნველყოფის პარამეტრები მნიშვნელოვნად გაუმჯობესდა მეორე თაობის საბორტო

კომპიუტერებთან შედარებით. მათი დამუშავების პერიოდი მოიცავს გასული საუკუნის 80-იანი წლების დასასრულისა და 90-იანი წლების დასაწყის პერიოდს.

საბორტო კომპიუტერები მეოთხე თაობა ხასიათდება ღია არქიტექტურით. ამ თაობის საბორტო კომპიუტერებს გააჩნიათ ინტეგრირებული სტრუქტურა, რომლის შემადგენლობაში საერთო დანიშნულების პროცესორთან ერთად შედის ასევე სპეციალიზებული პროცესორები, ხოლო გარე ინტერფეისის სახით მათში გამოიყენება ორი ტიპის არხი ერთი ერთჯერადი ბრძანებების გაცვლისთვის, მეორე კი ძირითადი ინფორმაციის გაცვლისთვის.

ამ თაობის საბორტო კომპიუტერების სტრუქტურის შედარებით სრულყოფილება საშუალებას იძლევა გარე ინტერფეისის შემადგენლობაში ჩაირთოს სპეციალური არხები და ქსელური ინტერფეისები (მაგალითად AS4074 ტიპის ქსელური ინტერფეისი). ღია უნიფიცირებული ინტერფეისების გამოყენება საშუალებას იძლევა განხორციელდეს ღრმა უნიფიკაცია, რითაც მოცული იქნება კომპიუტერის ყველა კომპონენტი: აპარატული მოდულები, კონსტრუქციული შესრულება და პროგრამული უზრუნველყოფა.

მე-4 თაობის საბორტო კომპიუტერების ტიპური წარმომადგენლებია საბორტო კომპიუტერების „ბაგეტის“ ოჯახის ტიპის კომპიუტერები, რომელთა დანიშნულებას, სტრუქტურას, ტექნიკურ მახასიათებლებს და მუშაობის პრინციპს ჩვენ დაწვრილებით განვიხილავთ მოცემულ სალექციო კურსში.

მე-5 თაობის საბორტო კომპიუტერების დამუშავებისას მომავალში, მათ სტრუქტურაში აუცილებელი იქნება უნიფიცირებული შიგა და გარე (თანმიმდევრობითი) ინტერფეისების, ინტელექტუალური პროცესორების მოდულების, ასევე ინტელექტუალური ინტერფეისის, განვითარებული ოპერაციული სისტემების და ინსტრუმენტული საშუალებების არსებობა მათი სასიცოცხლო ციკლის პროგრამების დამუშავების ყველა ეტაპზე.

1.2. საბორტო კომპიუტერების დანიშნულება და მათი მუშაობის განსაკუთრებულობები

თანამედროვე საფრენი აპარატი წარმოადგენს ტექნიკური საშუალებების კომპლექსს, რომლებიც ფუნქციურად არიან დაკავშირებული ერთმანეთთან და ემსახურებიან მის (საფრენი აპარატის) წინაშე დასმული ამოცანების გადაწყვეტას. თითოეული საფრენი აპარატის (მაგალითად გამანადგურებლის) მიერ თავისი დავალებების შესრულება დამოკიდებულია თვითმფრინავის საფრენოსო - ტექნიკურ და ტაქტიკურ მონაცემებზე, მფრინავის გამოცდილებაზე, ასევე მიწისზედა სამსახურის მუშაობის მოქნილ მუშაობაზე, მათ რიცხვში რა თქმა უნდა იგულისხმება რადიო ელექტრონული მოწყობილობებისა და სისტემების სპეციალისტები. ამიტომ საავიაციო კომპლექსი მიეკუთვნება რთულ ადამიანურ-მანქანურ სისტემას, რომელიც შედგება საფრენი აპარატისგან, მასთან ერთად მიწისზედა ტექნიკური საშუალებების ერთობ-ლიობისა და მათი მართვის განმხორციელებელი ადამიანებისგან (სპეციალისტებისგან).

ნებისმიერი ამოცანის შესრულება მოითხოვს ამ ამოცანის გადაწყვეტისათვის საჭირო ინფორმაციის შეკრების და დამუშავების ოპერაციების შესრულებას შესაბამისი ალგორითმების მიხედვით, რომლებშიც გათვალისწინებულია მართვის ობიექტის მდგომარეობის და მისი ფრენის პირობები გარემოში. კონკრეტული სიტუაციის შესაბამისად ხდება საფრენი აპარატის მართვის სიგნალების გამომუშავება, კერძოდ კი მართვის პროცესის მართვა .

ამის შესაბამისად განისაზღვრება საავიაციო კომპლექსის ძირითად საშუალებათა აუცილებელი შემადგენლობა:

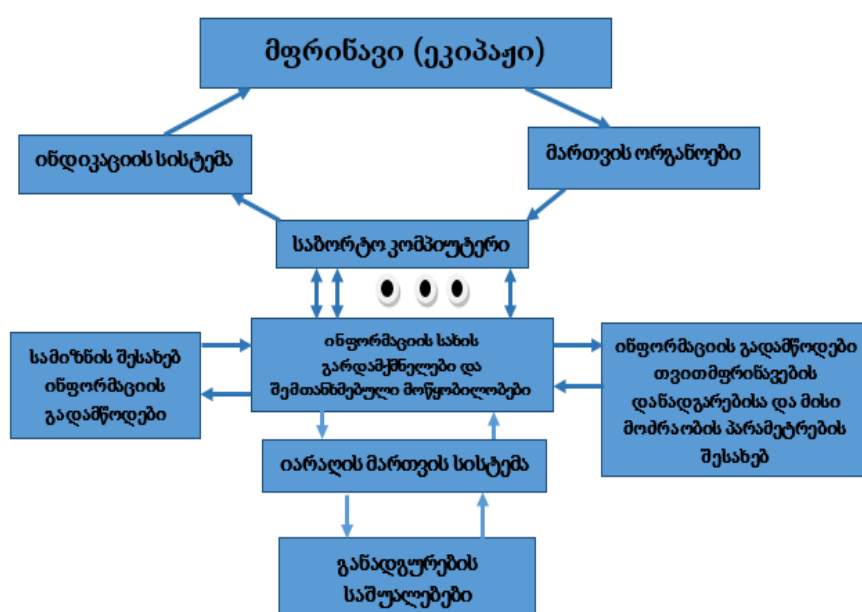
- საინფორმაციო საშუალებები - პირველადი ინფორმაციის წყაროები ;
- ინფორმაციის გადაცემის ხაზები ;
- ინფორმაციის დამუშავების საშუალების, გადაწყვეტილების მიღებისა და მართვის ბრძანებების ფორმირების საშუალებები ;
- მართვის ბრძანებების ფორმირების საშუალებები.

ამ ჩამონათვალიდან ცენტრალური ადგილი უკავია ინფორმაციის დამუშავების, გადაწყვეტილების მიღებისა და მართვის ბრძანებების ფორმირების საშუალებებს.

ეს თავის მხრივ წარმოადგენს სხვადასხვა ფუნქციური საშუალებების გამოთვლითი მოწყობილობების ერთობლიობას, რომლებიც დაკავშირებული არიან ერთმანეთთან და მთლიანობაში შეადგენენ გამოთვლით სისტემას (გს).

გამოთვლითი სისტემის ცნების ქვეშ იგულისხმება ერთმანეთთან ურთიერთ-დაკავშირებული „აპარატულ - პროგრამული საშუალებების ერთობლიობა, რომლებიც ემსახურება ციფრული ინფორმაციის გადაცემის, შენახვისა და გადამუშავების ოპერაციების შესრულებას დასმული ამოცანების გადაწყვეტისათვის შემავალი მონაცემების გამომავალ მონაცემებად გარდაქმნის თვალსაზრისით.

საჭაერო სამიზნეების განადგურების ძირითად საშუალებებს წარმოადგენს ზებგერითი მაღალმანევრირებადი გამანადგურებლები, რომელთა საბრძოლო მოქმედებების ძირითად სახეს წარმოადგენს საჭაერო ბრძოლა. თანამედროვე საჭაერო ბრძოლა ხასიათდება მაღალი სიჩქარეებით, ინტენსიური მანევრირებით, დიდი ინფორმაციული გაჯერებით, რაც მთლიანობაში განაპირობებს ეკიპაჟზე მაღალი ფსიქოფიზიკურ დატვირთვების ზემოქმედებას. ასეთ პირობებში მიუხედავად იმისა, რომ განუხრელად მიმდინარეობს მფრინავების მომზადების ხარისხის სრულყოფა, მაინც ისინი არასრულად იყენებენ თვითმფრინავის პოტენციურ შესაძლებლობებს. გამოსავალს ამ მდგომარეობიდან წარმოადგენს ინფორმაციის დამუშავების ავტომატიზაცია და გამანადგურებლის მართვა საბორტო კომპიუტერის დახმარებით. ასეთი ავტომატიზებული სისტემის განზოგადებული სტრუქტურული სქემა მოცემულია ნახ.1.1-ზე.



ნახ.1.1

სტრუქტურული სქემის შემადგენლობაში შედის ინფორმაციის გადამწოდები სამიზნეების შესახებ, ასევე ინფორმაცია განმანდგურებლის დანადგარების მდგომარეობის და მისი მოძრაობის პარამეტრების შესახებ. ზოგადად ეს შეიძლება იყოს როგორც ცალკეული მოწყობილობები (მაგალითად შეტევის კუთხის გადამწოდი) ან მთელი სისტემები. უკანასკნელთაგან ერთ-ერთი მაგალითი შეიძლება იყოს საბორტო რადიოლოკაციური სისტემა. იმდენად რამდენადაც, ინფორმაციის წარმოდგენის ფორმა ყოველთვის არ არის მისაღები მის დასამუშავებლად საბორტო კომპიუტერში (ე.ი ინფორმაციას არ გააჩნია ორობით კოდებში წარმოდგენილი სახე), ამდენად ინფორმაციის სახის და მისი მიწოდების სიჩქარის შესათანხმებლად გამოიყენება შეტანა-გამოტანის მოწყობილობა. შეტანა-გამოტანის მოწყობილობა კონკრეტული შესრულების ზოგიერთ ვარიანტებში შეიძლება უშუალოდ შედიოდეს საბორტო კომპიუტერის შემადგენლობაში.

ზოგადად განმანდგურებლის საბრძოლო ამოცანა მდგომარეობს საჭაერო ობიექტის განადგურებაში. ამ ამოცანის გადაწყვეტისათვის აუცილებელია აღმოჩენილ იქნეს ობიექტი, აღქმულ იქნეს იგი როგორც სამიზნე და განისაზღვროს პარამეტრები განადგურების საშუალებების მომზადებისათვის და შემდგომ მათი გამოყენებისათვის. მთელი ჩამოთვლილი ეტაპებისთვის საბორტო კომპიუტერის მათემატიკური უზრუნველყოფის შემადგენლობაში არსებობს შესაბამისი პროგრამები. ინფორმაცია რომელიც მიიღება ამ პროგრამების შედეგად, მიეწოდება მფრინავს ინდიკაციის სისტემაზე, ასევე იარაღის მართვის სისტემას და მათი ფუნქციონირების რეჟიმების ცვლილებებისათვის გადამწოდებს.

მფრინავი მართვის შესაბამისი ორგანოების დახმარებით იყენებს რადიოელექტრონული კომპლექსის ყველა სისტემის მუშაობის აუცილებელ რეჟიმებს და ასევე უზრუნველყოფს საბორტო კომპიუტერის გამომავალი პროცესების მართვას.

საბორტო კომპიუტერის მუშაობის განსაკუთრებულობას განმანდგურებლის მართვის კონტურში წარმოადგენს: საბორტო კომპიუტერის შეუღლება (შეპირაპირება) სხვადასხვაგვარი ფიზიკური ხასიათის აპარატურასთან; ალგორითმების მრავალჯერადი გამეორება; დროის რეალური მასშტაბი; ალგორითმების რეალიზაციის შეზღუდვა დროში; საიმედოობის ამაღლება.

პირველი დამახასიათებელი განსაკუთრებულობა მდგომარეობს იმაში, რომ ფიზიკური სიდიდეები, რომლებიც იზომებიან გადამწოდების მიერ თავისი ბუნებით წარმოადგენენ უწყვეტ სიდიდეებს რის გამოც საბორტო კომპიუტერებში მათ შესაყვანად

საჭიროა მათი ციფრულ ფორმაში გარდაქმნა. ასეთი გარდაქმნა შესაძლებელია განხორციელდეს უშუალოდ გადამწოდების ან ანალოგურ-ციფრული გარდამქნელების საშუალებით.

როგორც ცნობილია შემსრულებელი მოწყობილობების უმრავლესობა მუშაობს უწყვეტი სიგნალებით, ამიტომ საბორტო კომპიუტერებში მიღებული დისკრეტული შედეგები გარდაქმნილ უნდა იქნეს ანალოგურ სიდიდეებში მათი შემდგომი გამოყენების მიზნით. საჭირო ინფორმაციის სხვადასხვაგვარი ფიზიკური ხასიათი იწვევს მთელი სისტემის მნიშვნელოვან გართულებას, მისი მასის, გაბარიტების და ღირებულების გაზრდას. გარდა ამისა, მაღალი სიზუსტის გარდამქმნელების შექმნა თავისთავად ქმნის განსაზღვრულ პრობლემებს და სიძნელებებს მათი შექმნის თვალსაზრისით.

მეორე განსაკუთრებულობა გამანადგურებლის კონტურში მდგომარეობს იმაში რომ ამოცანების გადასაწყვეტი ალგორითმები, რომლებიც აღწერილია მანქანურ ენაზე არ იცვლება საფრენი აპარატის ექსპლუატაციის მთელი პეროოდის პროცესში და შენახულ უნდა იქნეს მუდმივ დამხსომებელ მოწყობილობაში. ნებისმიერი ყოველი ალგორითმის შესრულება მიმდინარეობს მისი მრავალჯერადი გამეორებით. ალგორითმის ყოველი გამეორებისას იცვლება მხოლოდ საწყისი მონაცემები. ამა თუ იმ ალგორითმის გამეორების სიხშირე დამოკიდებულია საწყისი მონაცემების ცვლილების სიჩქარისგან, შედეგების გაცემის დასაშვები დისკრეტულობით, რომლებიც შესაბამისობაში უნდა იყოს რადიოელექტრონული კომპლექსის გამოყენების პირობებთან.

მესამე განსაკუთრებულობა მდგომარეობს დროის რეალური მასშტაბის რეალიზაციაში. ეს განსაკუთრებულობა მდგომარეობს იმაში რომ გამოსასვლელი ინფორმაციის დისკრეტული ხასიათის შემთხვევაში უზრუნველყოფილ უნდა იქნეს რადიოელექტრონული კომპლექსის ქვესისტემების უწყვეტი მართვა ამოცანის გადაწყვეტის მოთხოვნილი სიზუსტის შენარჩუნებით. ამასთან ერთად გამოთვლების შედეგის მიღება უნდა მოხდეს არაუგვიანეს იმისა, ვიდრე საგრძნობლად შეიცვლებიან ალგორითმის მუშაობისათვის საჭირო საწყისი მონაცემები.

მეოთხე განსაკუთრებულობა მდგომარეობს თითოეული ალგორითმის რეალიზაციისათვის საჭირო დროის მოთხოვნილი ინტერვალის წარმოდგენაში, რომელიც განსაზღვრავს საბორტო კომპიუტერის და რადიოელექტრონული კომპლექსის ქვესისტემების მუშაობის ციკლს .

მეხუთე განსაკუთრებულობა ხაზს უსვამს ისეთ მნიშვნელოვან მახასიათებელს, როგორიცაა საიმედოობა, რომელიც თავის მხრივ განისაზღვრება არა მარტო საბოროტო კომპიუტერის ელემენტებისა და მისი დამზადების ტექნოლოგიით, არამედ პროგრამული უზრუნველყოფის საიმედოობითაც. ამ უკანასკნელის საიმედოობა განისაზღვრება ფაქტორების შემდეგი ერთობლიობით: ამოცანის გადაწყვეტის მდგრადი მეთოდების არჩევა; ალგორითმების პროგრამული დაცვის ეფექტურობა; ალგორითმების რეალიზაციის სისწორის კონტროლის პროგრამული მეთოდების განვითარების ხარისხი და ა.შ.

საბოროტო კომპიუტერის მუშაობის ყველა ზემოთ ჩამოთვლილი განსაკუთრებულობა, მის მიერ გადასაწყვეტი ამოცანების დიაპაზონი და როლი რადიოელექტრონული კომპლექსის შემადგენლობაში განაპირობებს სპეციფიკურ მოთხოვნათა ფართო სპექტრს გამოთვლითი სისტემის ძირითადი მახასიათებლების და მისი პროგრამული და აპარატული უზრუნველყოფის შემადგენლობის მიმართ.

1.3. საბოროტო კომპიუტერების კლასიფიკაცია, შემადგენლობა და მუშაობის პრინციპი

ინფორმაციის წარმოდგენის ფორმის მიხედვით კომპიუტერები იყოფიან ორკლასად: ანალოგური და ციფრული (დისკრეტული). პირველი კომპიუტერები წარმოადგენენ ანალოგურ გამომთვლელებს. ამჟამად საავიაციო რადიოელექტრონულ მოწყობილობებში (კავშირის სისტემები, რადიოლოკაცია, სიტყვიერი სიგნალების დამუშავება და ა.შ) ფართო გამოყენებას პოულობენ ციფრული კომპიუტერები, რომელთაც გააჩნიათ მთელი რიგი უპირატესობები ანალოგურ კომპიუტერებთან შედარებით. კერძოდ: უფრო მაღალი საიმედოობა; მადესტაბილიზებული ფაქტორების ზემოქმედებისას პარამეტრების სტაბილურობა; ინფორმაციის დამუშავების მაღალი სიზუსტე; შრომატევადობის მნიშვნელოვანი შემცირება, რეგულირების და აწყობის ოპერაციების გამარტივება; ძალზე მაღალი ინტეგრაციის ხარისხის მიკროსქემების შექმნის შესაძლებლობა.

აღნიშნულიდან გამომდინარე, ჩვენ განვიხილავთ მხოლოდ ციფრული საბოროტო კომპიუტერების შემადგენლობას და მუშაობის პრინციპს.

საბოროტო ციფრული კომპიუტერი წარმოადგენს დისკრეტული ინფორმაციის ავტომატურ გარდამქმნელს, რომელიც აგებულია პროგრამული მართვის პრინციპით.

პროგრამული მართვის ძირითადი დებულებებია:

1. საბორტო კომპიუტერებში ინფორმაციის კოდირება ორობით ფორმაში.

ინფორმაციის მანქანურ ერთეულს წარმოადგენს სიტყვა. ის კომპიუტერში მუშავდება როგორც ერთი მთლიანი. მაგრამ, ისე როგორც ჩვეულებრივ პრაქტიკაში, მანქანური სიტყვა შესაძლებელია დაიყოს მარცვლებად, რომლებიც შედგებიან ასოებისგან. ორობითი კოდირების დროს ინფორმაციის ანბანი შედგება ორი ციფრისგან: 0 და 1. ადგილი, რომელსაც იკავებს ციფრი მანქანური სიტყვის ჩანაწერში იწოდება თანრიგად. ერთი თანრიგი შეიცავს ინფორმაციის ერთ ბიტს. თანამედროვე საბორტო კომპიუტერები მუშაობენ სიტყვებით, რომლებთაგან თითოეულის სიგრძე ჯერადაა 8 ბიტის, ე.ი. ბაიტის.

გამოთვლების მიმდინარეობის დროს კომპიუტერში ცირკულირებს ორი სახის ინფორმაცია:

- რიცხვითი, რომელიც მიეკუთვნება გადასაწყვეტი ამოცანების მონაცემებს (საწყისი სიდიდეები, საშუალო და საბოლოო შედეგები);
- მმართველი, რომელიც მიეკუთვნება გადასაწყვეტი ამოცანების ალგორითმებს.

მანქანური სიტყვა წარმოადგენს ფორმის მიხედვით, რომელიც მიეკუთვნება მმართველ ინფორმაციას, შეუძლებელია განსხვავდებოდეს რიცხვისგან იმდენად რამდენადაც ისინი კოდირდებიან ერთნაირად. რიცხვითი და მმართველი ინფორმაციის ასეთი ერთიანობა საშუალებას იძლევა საბორტო კომპიუტერში დამუშავდეს არა მარტო რიცხვები, არამედ მმართველი სიტყვაც, ე.ი. თვით ალგორითმის ელემენტები.

2. რიცხვითი და მმართველი ინფორმაცია ალგორითმის რეალიზაციის მთელი დროის განმავლობაში ინახება საბორტო კომპიუტერის მეხსიერებაში. მანქანური ინფორმაციის ერთეულის შენახვისთვის (მანქანური სიტყვის ან ბაიტის შენახვა) გამოიყენება მეხსიერების უჯრედი. საბორტო კომპიუტერებში გამოყენებულია მანქანურ სიტყვასთან მიმართვის სამისამართო პრინციპი. ამასთან მეხსიერების ყველა უჯრედი ინომრება. მოცემული უჯრედის ნომერი იწოდება მის მისამართად. მეხსიერების უჯრედის მისამართი ასრულებს მანქანური იდენტიფიკატორის (სახელის) როლს, მასში შენახული სიტყვისათვის. იმისათვის რომ მოცემულ უჯრედში შეტანილ იქნეს სიტყვა საჭიროა მითითებული იყოს მისი მისამართი. ანალოგიურად მეხსიერებიდან ამა თუ იმ სიტყვის ამოღებისთვის (ამორჩევისთვის) საჭიროა მიეთითოს შესაბამისი უჯრედის მისამართი.

3. თითოეული ამოცანის გადაწყვეტის ალგორითმი კომპიუტერში წარმოდგება მმართველი სიტყვების თანმიმდევრობით - ბრძანებებით. ბრძანება იძლევა საბორტო კომპიუტერის ყველა მოქმედებას გამოთვლითი პროცესის ერთ ბიჯზე. სახელდობრ, განსაზღვრავს ამ ბრძანების შესაბამის ოპერაციას და მასში მონაწილე ოპერანდებსაც. ამის შესაბამისად ბრძანების ორობითი კოდი უნდა შეიცავდეს ორ ნაწილს: ოპერაციულს და ოპერანდულს.

ბრძანების ოპერაციულ ნაწილში ჩაწერილია იმ ოპერაციის კოდი, რომელიც უნდა შესრულდეს მოცემული ბრძანებით.

ბრძანების ოპერანდულ ნაწილში მოთავსებულია ცნობები ოპერანდების შესახებ. ამ ცნობების სახით შეიძლება წარმოდგენილი იქნეს შესაბამის ოპერანდების მისამართები. ამიტომ ბრძანების ოპერანდულ ნაწილს ასევე უწოდებენ მის სამისამართო ნაწილს.

ალგორითმი, რომელიც წარმოდგენილია ბრძანებების ტერმინებში, იწოდება პროგრამად.

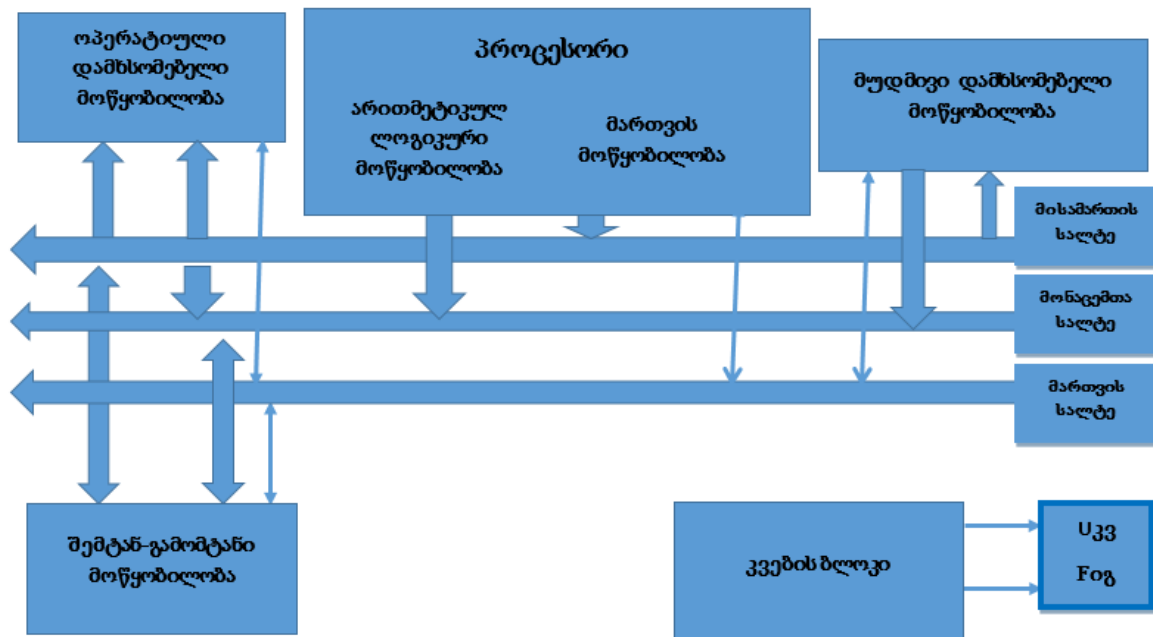
4. კომპიუტერის პრინციპულ განსაკუთრებულობას, რომელიც განასხვავებს მას გამოთვლების ავტომატიზაციის სხვა საშუალებებისგან, წარმოადგენს ის, რომ ამოცანის გადაწყვეტის პროგრამა თავსდება მეხსიერებაში.

კომპიუტერში გამოთვლების შესრულება, რომელიც გაწერილია ალგორითმით, დაიყვანება ბრძანებების რეალიზაციის თანმიმდევრობაზე. ბრძანებების შესრულების რიგი ცალსახად განისაზღვრება პროგრამით. ყველაზე პირველად შესრულებადი ბრძანება, მოიცემა პროგრამის გამშვები მისამართით. გამშვები მისამართი (ჩვეულებრივ ეს არის მოცემული პროგრამის პირველი ბრძანების მისამართი) მოიცემა ან კომპიუტერის მართვის პულტიდან ან ფორმირდება აპარატულად.

მომდევნო ბრძანების მისამართი ცალსახად განისაზღვრება მიმდინარე ბრძანების შესრულების მიმდინარეობის დროს. გამოთვლითი პროცესი გრძელდება მანამ, სანამ არ შესრულდება ბრძანება, რომელსაც მიწერილი ექნება გამოთვლის დასრულება.

1.4. საბორტო კომპიუტერების სტრუქტურა და ფუნქციონირება

საბორტო კომპიუტერის დანიშნულებაა ამოცანათა ფიქსირებული წრის გადაწყვეტა, რაც განსაზღვრავს მის მნიშვნელოვან სპეციალიზაციას. სპეციალიზებული კომპიუტერებისათვის დამახასიათებელია განსაკუთრებულობების არსებობა შემადგენლობისა და ფუნქციონირების პრინციპების თვალსაზრისით. კომპიუტერის სტრუქტურული სქემების მრავალსახეობიდან ზოგადი საკითხების განხილვისათვის გამოვიყენოთ გავრცელებული სტრუქტურული სქემა, რომელიც მოცემულია ნახ.1.2-ზე.



ნახ.1.2

მასზე აღნიშნულია: შეტანა-გამოტანის მოწყობილობა; ოპერატიული დამხსომბელი მოწყობილობა; მუდმივი დამხსომბელი მოწყობილობა; არითმეტიკულ-ლოგიკური მოწყობილობა; მართვის მოწყობილობა; კვების ბლოკი; მისამართის, მონაცემების და მართვის სალტეები.

შეტანა-გამოტანის მოწყობილობის დანიშნულებაა კომპიუტერის დაკავშირება მართვის ობიექტთან. კონსტრუქციულად შემტან-გამოტანის მოწყობილობა, შეიძლება შესრულებული იყოს საბორტო კომპიუტერისგან ცალკე.

საბორტო კომპიუტერი, რომელიც შედის საავიაციო კომპლექსის შემადგენლობაში, ასრულებს ამოცანების გადაწყვეტას, რომელიც ძირითადად დაკავშირებულია საბორტო სიტემების მართვასთან. საბორტო კომპიუტერი ინფორმაციას იღებს სხვადასხვა სახის გადაწყვეტილებიდან, ხოლო გადაწყვეტის შედეგებს გასცემს კომპლექსის შემსრულებელ და ინდიკატორულ მოწყობილობებზე. ამასთან ინფორმაციის შეტანა, ამოცანების გადაწყვეტა და გადაწყვეტის შედეგების გაცემა სრულდება ისეთი სიხშირით, რომ

საჯერო ხომალდის ეკიპაჟმა და საბორტო სისტემებმა აღიქვან ეს შედეგები როგორც მართვის უწყვეტი პროცესი დასაშვები დინამიკური შეცდომებით.

საბორტო კომპიუტერის კავშირი გადამწოდებთან, შემსრულებელ და ინდიკატორულ მოწყობილობებთან ხორციელდება ინფორმაციის გადაცემის არხების საშუალებით, რომლებიც იმართებიან შეტანა-გამოტანის სპეციალიზებული მოწყობილობებით.

ზოგად შემთხვევაში საბორტო კომპიუტერის შეტანა-გამოტანის სპეციალიზებული მოწყობილობის შემადგენლობაში შედის:

- ინფორმაციის შეტანის არხები შესასვლელი გარდამქმნელებით;
- ინფორმაციის გამოტანის არხები გამოსასვლელი გარდამქმნელებით;
- მართვის მოწყობილობა.

შესასვლელი არხების დახმარებით და მათი გარდამქმნელების საშუალებით ხორციელდება გადამწოდების მიერთება და ინფორმაციის გარდაქმნა, რომელიც შემოდის შესასვლელი არხებიდან და შემდეგ გარდაიქმნება საბორტო კომპიუტერის კოდებში.

გამოსასვლელი არხები და მათი გარდამქმნელები ახორციელებენ კოდების გარდაქმნას (რომლებიც მიიღება საბორტო კომპიუტერში ამოცანების გადაწყვეტის შემდეგ) ინფორმაციაში, რომელიც უზრუნველყოფს კომპლექსის შემსრულებელი და ინდიკატორული მოწყობილობების მართვას.

შესასვლელი და გამოსასვლელი ინფორმაციის წარმოდგენის ფორმები საკმარისად სხვადასხვაგვარია. ეს მოითხოვს განსაზღვრულ მოთხოვნებს გარდამქმნელების სქემური რეალიზაციის თვალსაზრისით.

იმდენად რამდენადაც ინფორმაციის გადამწოდების, შემსრულებელი და ინდიკატორული მოწყობილობების რიცხვმა შეიძლება მიაღწიოს ათეულს ან შეიძლება ასეულსაც კი, ამიტომ შესასვლელი და გამოსასვლელი გარდამქმნელების უმრავლესობა სრულდება მრავალარხული სახით. ასეთი გარდამქმნელი თანმიმდევრობით უერთდება რამდენიმე გადამწოდს ან შემსრულებელ და ინდიკატორულ მოწყობილობებს, რითაც უზრუნველყოფილია ინფორმაციის რიგისობით შეტანა ან გამოტანა.

მართვის მოწყობილობა მიღებული პრიორიტეტის შესაბამისად ორგანიზებას უკეთებს შესასვლელი ან გამოსასვლელი არხების მიერთებას საბორტო კომპიუტერთან ინფორმაციის შეტანის ან გამოტანისთვის. ანაწილებს კომპიუტერის ოპერატიული დამხსომელებელი მოწყობილობის მეხსიერების უჯრედებს - კოდების მიღების და გაცემის

შემთხვევაში, უზრუნველყოფს მუშაობის მოთხოვნილი რეჟიმისათვის გარდამქნელების აწყობას, ასევე ორგანიზებას უკეთებს შეტანა-გამოტანის მოწყობილობის მუშაობის რეჟიმის კონტროლს. გარდა ამისა, შეტანა-გამოტანის მოწყობილობის მართვის მოწყობილობა უზრუნველყოფს სისტემებიდან შემოსული წყვეტის სიგნალების მიღებას და გაშიფვრას. თითოეული ასეთი სიგნალი მოითხოვს მოცემულ მომენტში შესრულებადი პროგრამის შეწყვეტას და კომპიუტერის გადაყვანას იმ პროგრამის შესრულებაზე, რომელიც მოთხოვნილია მოცემული სისტემის მხრიდან.

შეტანა - გამოტანის მოწყობილობის მართვის მოწყობილობა შეცავს ასევე დროის მთვლელს (ტაიმერი), რომელიც გსცემს ზუსტი დროის სიგნალებს. ჩვეულებრივ ასეთი სიგნალები ფორმირდებიან ასეულ ჰერცამდე სიხშირით. ეს ინფორმაცია გამოიყენება კომპიუტერის მიერ გამოთვლითი პროცესის ორგანიზაციისათვის. ამ სიგნალებით კომპიუტერი ასრულებს სპეციალურ პროგრამას, რომელიც იწოდება დისპეტჩერად.

ოპერატიული დამხსომებელი მოწყობილობის დანიშნულებაა ძირითადად იმ ინფორმაციის დამახსოვრება, რომლებიც მიეწოდება გადამწოდებიდან, ასევე საშუალებო და საბოლოო შედეგების შენახვა. ოპერატიულ დამხსომებელ მოწყობილობაში ხდება როგორც ინფორმაციის ჩაწერა, ისე იქიდან მისი ამოკითხვაც.

მუდმივი დამხსომებელი მოწყობილობის დანიშნულებაა ექსპლუატაციის პროცესში არაცვალებადი ინფორმაციის შენახვა. ასეთ ინფორმაციას მიეკუთვნება საბორტო კომპიუტერის მუშაობის პროგრამები და კონსტანტები, რომლებიც გამოიყენებიან ალგორითმების რეალიზაციისას. ინფორმაცია მუდმივ დამხსომებელ მოწყობილობაში იწერება ქარხანაში მისი დამზადებისას და არ ექვემდებარება ცვლილებას ექსპლუატაციის პირობებში. ინფორმაციის შენახვისათვის, რომლის მნიშვნელობები არ იცვლება ერთი ან რამდენიმე გაფრენისთვის, გამოიყენება გრძელვადიანი დამხსომებელი მოწყობილობა, რომელიც იძლევა როგორც ჩაწერის, ისე ამოკითხვის საშუალებას. ასეთი დამხსომებელი მოწყობილობის დამზადება ძალზე შრომატევადია, ამიტომ მისი ტევადობა ჩვეულებრივ მნიშვნელოვნად მცირეა ოპერატიული და მუდმივი დამხსომებელი მოწყობილობების ტევადობასთან შედარებით.

პროცესორი - კომპიუტერის ძირითადი მოწყობილობაა, რომლის დანიშნულებაა ოპერაციათა თანმიმდევრობის ავტომატური შესრულება, რომელიც გათვალისწინებულია ამოცანის გადაწყვეტის პროგრამით.

საბორტო კომპიუტერის პროცესორი ფუნქციურად შედგება არითმეტიკულ-ლოგიკური მოწყობილობისა და მართვის მოწყობილობისაგან. არითმეტიკულ-ლოგიკური მოწყობილობა გათვლილია არითმეტიკული და ლოგიკური ოპერაციების შესასრულებლად ორობით რიცხვებზე. მართვის მოწყობილობის დანიშნულებაა გამოთვლითი პროცესის ორგანიზაცია მოცემული პროგრამის შესაბამისად საბორტო კომპიუტერის ყველა მოწყობილობის მუშაობის იძულებითი კოორდინაციის გზით მმართველი სიგნალების დახმარებით.

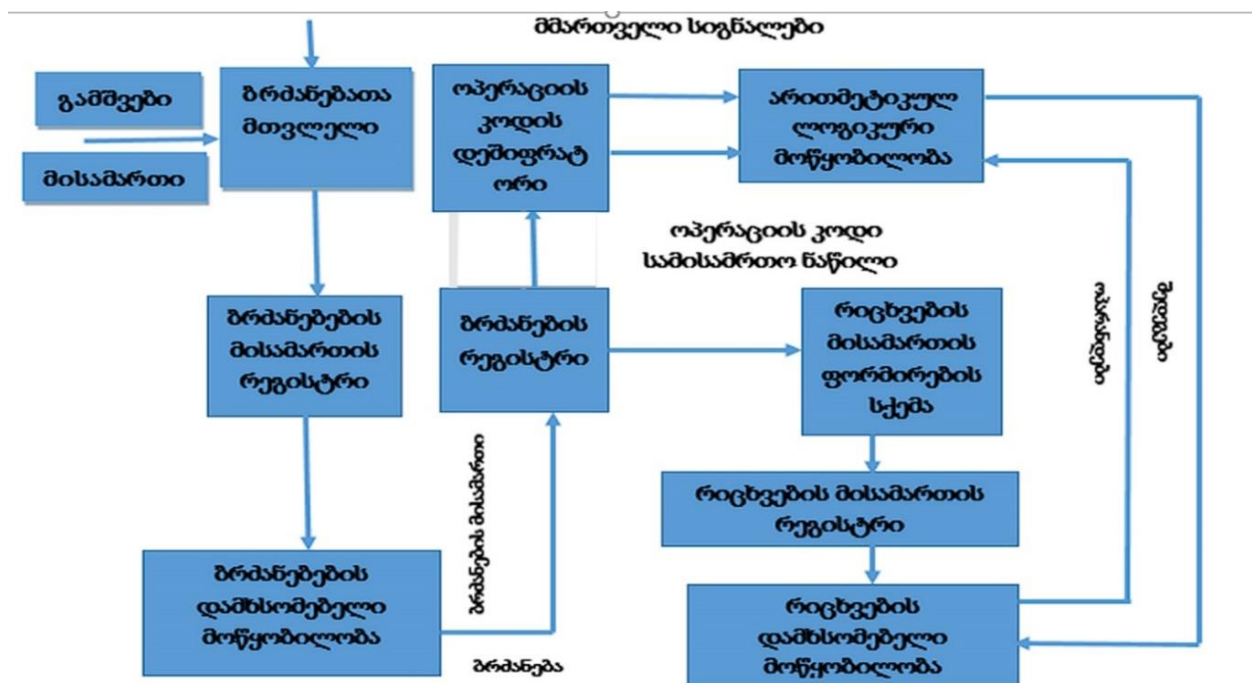
კვების ბლოკი უზრუნველყოფს კვების ძაბვების ფორმირებას, რომლებიც აუცილებელია საბორტო კომპიუტერის ყველა მოწყობილობების მუშაობისთვის, ასევე მასინქრონიზებელი სიგნალების სერიის გენერაციისათვის.

ნახ1.3-ზე მოყვანილია საბორტო კომპიუტერებში ბრძანებათა შესრულების გამარტივებული სქემა, რომელიც აჩვენებს პროგრამული მართვის პრინციპის ძირითად დებულებებს. ამ სქემის შესაბამისად პროგრამების შესრულების წესი ასეთია.

დროის საწყის მომენტში (კვების ჩართვის მომენტი ან მართვის პულტიდან შესაბამისი ბრძანების მიწოდების მომენტი) ბრძანებათა მრიცხველში, რომლის დანიშნულებაა მიმდინარე ბრძანების მისამართის ფორმირება, მიეწოდება პროგრამის გამშვები მისამართი. ამ მისამართის შენახვა ბრძანების შესრულების მთელი დროის განმავლობაში ხორციელდება ბრძანებათა მისამართის რეგისტრში (ბმრ). ბრძანების მისამართი ბრძანებათა მისამართის რეგისტრიდან მიეწოდება ბრძანებათა დამხსომებელ მოწყობილობაში. ამ მისამართით დამხსომებელი მოწყობილობიდან ამოიკითხება ბრძანება და ჩაიწერება ბრძანებათა რეგისტრში (ბრ), რომელშიც ინახება ბრძანების შესრულების მთელი დროის განმავლობაში.

ბრძანებათა რეგისტრიდან ოპერაციული და სამისამართო ნაწილები ცალ-ცალკე შესაბამისად მიეწოდებიან ოპერაციის კოდის დემიფრატორს და რიცხვის მისამართის ფორმირების სქემას. ოპერაციის კოდის დემიფრატორი აფორმირებს მმართველი სიგნალების ნაკრებს, რომლებიც აუცილებელია ბრძანების რეალიზაციისათვის. ეს სიგნალები შემდეგში მიეწოდება ყველა მოწყობილობას, რომლებიც მონაწილეობას იღებს ბრძანების შესრულებაში. მისამართის ფორმირების სქემა ბრძანების სამისამართო ნაწილის მიხედვით და სხვა ნიშან-თვისებებით, რომლებიც ახასიათებენ მისამართის ფორმირების ხერხს, უზრუნველყოფს რიცხვის მისამართის მიღებას. სამისამართო ნაწი-

ლის ზომისგან დამოკიდებულებით შესაძლებელია რამდენიმე ოპერანდის მისამართის ერთდროული ფორმირება. მაგალითად, პირველი შესაკრების მისამართი, მეორე შესაკრების მისამართი და შედეგის მისამართი. ფორმირებული მისამართის შენახვა ბრძანების შესრულების პროცესში ხორციელდება რიცხვის მისამართის რეგისტრში (რმრ).

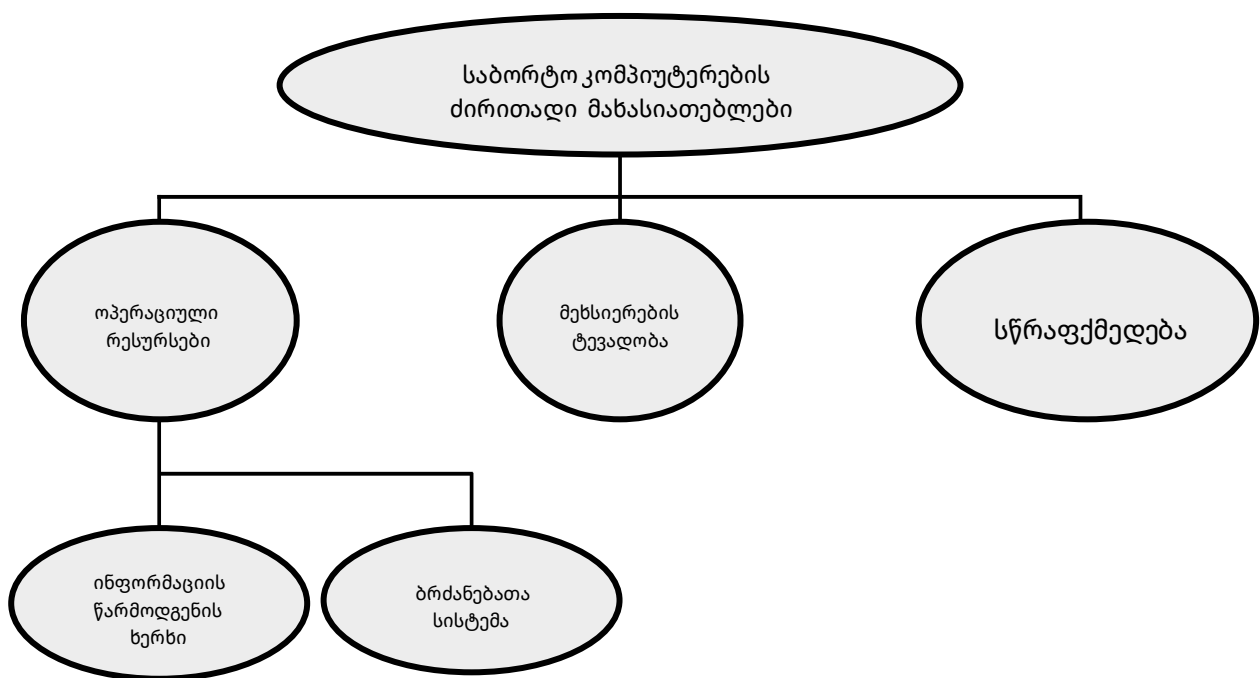


1.5. საბორტო კომპიუტერების ძირითადი მახასიათებლები

საბორტო კომპიუტერის შესაძლებლობები, როგორც გამოთვლითი საშუალებებისა განისაზღვრება მისი ტექნიკური მახასიათებლების ერთობლიობით, რომელთა შორის ძირითადია: ოპერაციული რესურსები, მეხსიერების ტევადობა და სწრაფაქმედება (ნახ.1.4).

ოპერაციული რესურსები ძირითადად განისაზღვრება ინფორმაციის წარმოდგენის ხერხით და ბრძანებათა სისტემით.

ინფორმაციის წარმოდგენის ხერხი - ეს არის ფორმათა სიმრავლე, რომლებიც გამოიყენება კომპიუტერში რიცხვების, ლოგიკური სიდიდეების და სხვათა წარმოდგენისთვის. ბრძანებების სისტემა - ეს ბრძანებათა ნუსხაა, რომლებიც განსაზღვრავენ მანქანური (კომპიუტერული) ოპერაციების ნომენკლატურის და დამისამართების ხერხებს (ოპერანდებისა და ბრძანებების მისამართების ფორმირების ხერხები).



ნახ.1.4

განვიხილოთ არსებული მახასიათებლებიდან ზოგიერთი, რომლებიც საბორტო კომპიუტერის ოპერაციულ რესურსებს შეადგენენ.

ინფორმაციის დამუშავებისა და გადაცემის ხერხისგან დამოკიდებულებით განასხვავებენ პარალელურ და თანმიმდევრობით კომპიუტერებს. პარალელურ კომპიუ-

ტერებში ინფორმაციის დამუშავება პროცესორებში, ასევე მისი გადაცემა კომპიუტერის მოწყობილობებს შორის წარმოებს ერთდროულად ყველა თანრიგების მიხედვით. თანმიმდევრობით კომპიუტერებში თითოეული მანქანური სიტყვა პროცესორში მუშავდება და გადაიცემა კომპიუტერის მოწყობილობებს შორის თანრიგობრივად. ცხადია, რომ სხვა ტოლი პირობების შემთხვევაში პარალელური კომპიუტერების სწრაფმედება უფრო მაღალია, ვიდრე თანმიმდევრობითი კომპიუტერებისა, თუმცა თანმიმდევრობითი კომპიუტერები ხასიათდებიან მოწყობილობათა მცირე მოცულობით.

საბორტო კომპიუტერის პროცესორში ოპერაციათა შესრულების მართვის ხერხის მიხედვით საბორტო კომპიუტერები იყოფიან სინქრონულად და ასინქრონულ საბორტო კომპიუტერებად. სინქრონულ საბორტო კომპიუტერებში თითოეული ოპერაციის შესრულებისთვის გამოიყოფა დრო, რომელიც განსაზღვრულია ოპერანდების დამუშავების ყველაზე უფრო უარესი ვარიანტებისთვის დროის თვალსაზრისით. სინქრონულ საბორტო კომპიუტერებში შემდეგი ოპერაცია არ დაიწყება მანამ, სანამ არ ამოიწურება ის დრო, რომელიც განკუთვნილია მიმდინარე ოპერაციის შესრულებისათვის, მიუხედავად იმისა დამთავრდა თუ არა ის ფაქტობრივად.

თითოეული ოპერაციის შესრულების დრო დაკავშირებულია ტაქტური (მასინქრონიზებული) იმპულსების რაოდენობის განსაზღვრასთან, რომლებიც თავის მხრივ გამომუშავდებიან შიგა ან გარე გენერატორით.

ასინქრონულ საბორტო კომპიუტერებში მანქანური ოპერაციების შესრულების ბლოკებს გააჩნიათ საშუალებები (ხელსაწყოები) რომლებიც ახდენენ მიმდინარე ოპერაციის დასრულების მომენტის ფიქსაციას და გამოიმუშავებენ სიგნალს მომდევნო ოპერაციის შესრულებისთვის. თანამედროვე საბორტო კომპიუტერები სინქრონული კომპიუტერებია.

საბორტო კომპიუტერებში გამოიყენება ბრძანებათა სხვადასხვა ფორმატები: ერთმისამართიანი და მრავალმისამართიანი, ფიქსირებული და ცვლადი დამისამართებით. გამოიყენება ასევე ოპერანდებისა და ბრძანებების მისამართების ფორმირების სხვადასხვა ხერხები: პირდაპირი დამისამართება, ირიბი და ფარდობითი დამისამართება.

საბორტო კომპიუტერის მნიშვნელოვან მახასიათებელს წარმოადგენს მისი ოპერატიული დამხსმომეხელი მოწყობილობა და მუდმივი დამხსმომეხელი მოწყობილობა. მუდმივი დამხსმომეხელი მოწყობილობის ტევადობა განისაზღვრება პროგრამებისა და კონსტანტების მოცულობით, რომელთა შენახვა აუცილებელია მასში.

მუდმივი დამხსმომებელი მოწყობილობების ტევადობა შემოსაზღვრულია ზემოდან შემდეგი ფაქტორით. პროცესორის დამუშავება ხდება იმ გაანგარიშებით რომ მან უნდა იმუშაოს მუდმივი დამხსმომებელი მოწყობილობის I სიგრძის თანრიგებიანი მისამართებით. ამასთან მუდმივი დამხსმომებელი მოწყობილობის ზღვრული ტევადობა, რომელიც დაშვებადია პროცესორისთვის შეადგენს სიდიდეს

$$E_{\text{მდმ}} = 2^{r_{\text{მდმ}}}.$$

ოპერატიული დამხსმომებელი მოწყობილობის ტევადობა განისაზღვრება გადასაწყვეტი ამოცანების ხასიათით და საბორტო კომპიუტერში პროგრამების შესრულების მიმდინარეობის ტემპით. ოპერატიული დამხსმომებელი მოწყობილობის ტევადობა ტოლია

$$E_{\text{ოდმ}} = 2^{r_{\text{ოდმ}}},$$

სადაც I -ოდმ - ოპერატიული დამხსმომებელი მოწყობილობის მისამართის თანრიგობაა.

საბორტო კომპიუტერის სწრაფქმედება, ხასიათდება სიჩქარით, რითაც წარმოებს ინფრომაციის დამუშავება. ის განისაზღვრება ან ოპერაციათა რიცხვით, რომლებიც სრულდება დროის ერთეულში, ან ერთი ოპერაციის შესრულების დროით.

ლექცია 2

საბოროტო კომპიუტერებში ბრძანებებისა და რიცხვების წარმოდგენის ფორმატები და მათი ბრძანებების სისტემა

2.1. საბოროტო კომპიუტერში ბრძანებების და რიცხვების წარმოდგენის ფორმატები

ფორმატად იწოდება ორობითი კოდის თანრიგების რაოდენობა, რომელიც გამოიყენება საბოროტო კომპიუტერის ბრძანებების, რიცხვების და მისამართების ჩაწერისათვის. საბოროტო კომპიუტერის დანიშნულებაა მმართველი პროგრამების შესრულება, რომლებთაგან თითოეული წარმოადგენს ბრძანებათა თანმიმდევრობას. კომპიუტერის ბრძანება უნდა ფლობდეს ინფორმაციას იმის შესახებ თუ რა უნდა გაკეთდეს და რაზე უნდა განხორციელდეს ეს მოქმედებები. ამიტომ ბრძანების სტრუქტურა უნდა შეიცავდეს ოპერაციულ და ოპერანდულ ნაწილებს. ორადგილიანი ოპერაციის შესრულებისას, მაგალითად, ისეთის როგორიცაა შეკრების ოპერაცია, ოპერანდული ნაწილი უნდა შეიცავდეს ინფორმაციას პირველი და მეორე ოპერანდების განლაგების ადგილის შესახებ ან ამ ოპერანდების მნიშვნელობების შესახებ. ამავდროულად, ოპერაციის შედეგი (ჯამი) განლაგებული უნდა იქნეს მოცემული მისამართის მიხედვით, ამიტომ ზოგადად ბრძანების ფორმატი უნდა იყოს სამ მისამართიანი:

ოკ	A1	A2	A3

სადაც ოკ - ოპერაციის კოდია (ბრძანების ოპერაციული ნაწილი) ;

A1, A2, A3 - შესაბამისად პირველი, მეორე და მესამე ოპერანდების მისამართი და შედეგის მისამართია.

თუ ოპერანდებიდან ერთ-ერთი არაა საჭირო შემდგომი გამოთვლებისთვის, მაშინ მისი მისამართი გამოიყენება შედეგის შენახვისთვის. ბრძანების ფორმატი ასეთ შემთხვევაში იწოდება ორ მისამართიან ფორმატად:

ოკ	A1	A2

საბორტო კომპიუტერის შესაძლებლობები ინფორმაციის დამუშავების მიხედვით უმეტესად განისაზღვრება მისი ბრძანებების სისტემით, ე.ი. იმით თუ რა სახის ბრძანებების შესრულება შეუძლია მოცემულ საბორტო კომპიუტერს.

თანამედროვე საბორტო კომპიუტერის ბრძანებების სისტემის შემადგენლობა შეიცავს 64-დან 256-მდე სხვადასხვა ბრძანებას. ამრიგად, ბრძანების ორობითი ნომერი (ოპერაციის კოდი) უნდა შეიცავდეს 6-დან 8-მდე თანრიგს. თუ მეხსიერების მოცულობა $E=64$ კილო სიტყვას, მაშინ მისამართის კოდირებისთვის მოითხოვება $n = \text{int}(\log_2 E) = 16$ თანრიგი, ამიტომ ორმისამართიანი ბრძანების შემთხვევაში საჭიროა 38-დან 40-მდე თანრიგი. იმისათვის რომ შეამცირებულ იქნეს ბრძანების მიერ დაკავებული მეხსიერების თანრიგების რაოდენობა, საბორტო კომპიუტერის პროცესორის სტრუქტურაში ერთ-ერთი ოპერანდის შენახვისათვის გამოიყენება სპეციალური რეგისტრი. „ორბიტა“ ტიპის საბორტო კომპიუტერში ეს არის რეგისტრი-დამგროვებელი (რდ), ხოლო A-15 ტიპის საბორტო კომპიუტერში გამოიყენება პირველი ამჯამავის რეგისტრი (რგა1). ასეთ შემთხვევაში ბრძანების ფორმატი შედგება ოპერაციის კოდისგან და მეორე ოპერანდის მისამართისგან:

ოკ	A1

ბრძანების თანრიგების შემდგომი შემცირება შესაძლებელია, მაშინ თუ ბრძანების ოპერანდულ ნაწილში წარმოდგენილია არა მთელი მისამართი, არამედ მხოლოდ მისი ნაწილი. ასეთ შემთხვევაში სრული მისამართი ფორმირდება იმ წესების მიხედვით, რომელიც განისაზღვრება სპეციალური ორობითი კოდით, რომელიც იწოდება დამისამართების ნიშან-თვისებად (დნთ).

ცხრილში 2.1 მოცემულია A-15A საბორტო კომპიუტერის ბრძანებათა ფორმატი.

ცხრილი 2.1

ბრძანებების თანრიგების ნუმერაცია	ნულოვანი ტრიადა			პირველი			მე-2			მე-3	მე-4			მე-5			მე-6		
	A	B	C	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ფუნქციური დანიშნულება	დამისამართ ების ნიშან- თვისება			ოპერაციის კოდი						სამისამართო ნაწილი (წანაცვლება)									

საბორტო კომპიუტერის II-100-ის ბრძანებების ფორმატი შეესაბამება სამმისამართიანი ბრძანების ფორმატს და ზოგადად შეიცავს სამ ველს, რომლებთაგან თითოეული შედგება ერთი ან რამდენიმე ბაიტისგან:

- ოპერაციის კოდის ველი ;
- ველი, რომელიც შეიცავს მონაცემების აღწერას და მითითებას შედეგის განთავსების შესახებ;
- ნაჭდევების ველი.

ოპერაციის კოდის ველი შედგება ერთი ბაიტისგან, რომელშიც ჩაიწერება ოპერაციის ნომერი (ბრძანების). ოპერაციის კოდის ერთი ბაიტი საშუალებას იძლევა დაკოდირდეს 256 ბრძანება.

საბორტო კომპიუტერში აპარატურული დანახარჯების შესამცირებლად არითმეტიკულ-ლოგიკური მოწყობილობის რეალიზაციისათვის გამოიყენება მანქანური ორობითი კოდები რიცხვების წარმოდგენისთვის. ჩვეულებრივ ასეთ დროს დადებითი რიცხვები წარმოდგება წესიერი წილადების სახით პირდაპირ კოდში, ხოლო უარყოფით რიცხვები კი დამატებით კოდში. რიცხვების კოდი ნაჩვენებია ცხრილში 2.2 .

რიცხვის ფორმატი

ცხრილი 2.2

რიცხვის თანრიგების ნუმერაცია		პირველი ტრიადა			მე-2			მე-3			მე-4			მე-5		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
დანიშნულება	ნიშანი	რიცხვის მანტისა														

უმცირესი რიცხვი, რომელიც წარმოდგენილია სათანრიგო ბადეში ტოლია $0.00...01=2^{-15}$. . სათანრიგო ბადეში წარმოდგენილი რიცხვების დიაპაზონია $(1-2^{-15})$ - დან -2^{-15} -მდე და $+2^{-15}$ -დან $+(1-2^{-15})$ - მდე. რიცხვები, რომლებიც ნაკლებია 2^{-15} - ზე სათანრიგო ბადეში არ აისახება და მიიღება 0 - ის ტოლად (რიცხვი გადის სათანრიგო ბადის საზღვრებს გარეთ მარჯვნივ). ყველა რიცხვი, რომელიც, აღემატება $1-2^{-15}$ რიცხვს, ასევე არ იქნება წარმოდგენილი მიღებულ სათანრიგო ბადეში. ასეთი რიცხვი გადის სამისამართო ბადის საზღვრიდან მარცხნივ (სათანრიგო ბადის გადავსება) და მისი უფროსი თანრიგები იკარგება (მძიმისგან მარცხნივ თანრიგები). რადგან ამ შემთხვევაში გამოთვლის შედეგი ხდება არასწორი, ამიტომ გადავსების დროს ხდება მანქანის გაჩერება ან ფორმირდება პროგრამის წყვეტის სიგნალი.

იმისათვის, რომ გამოირიცხოს სათანრიგო ბადის გადავსება საჭიროა დაპროგრამებისას ამოცანის მონაცემები, რომლებიც მონაწილეობას იღებენ გამოთვლებში აღებულ იქნეს შესაბამისი სამასშტაბო კოფიციენტებით.

სამეცნიერო-ტექნიკური გაანგარიშებების დროს მასშტაბირება ხდება უფრო მარტივი, თუ ყველა რიცხვი მოდულის მიხედვით იქნება 1-ზე ნაკლები, ე.ი. თუ მძიმე დაფიქსირებულია რიცხვის უფროსი თანრიგის წინ, ხოლო მასშტაბური კოეფიციენტები წარმოდგენილი იქნება 2^i ის სახით.

საბოროტო კომპიუტერებში რეალიზებულია რიცხვების წარმოდგენა ფიქსირებული მძიმით. მათში რიცხვების წარმოდგენა ფიქსირებული მძიმით საშუალებას იძლევა გამარტივდეს კომპიუტერის სქემა, გაიზარდოს მისი სწრაფქმედება, თუმცა ეს იწვევს სიძნელეებს დაპროგრამების თვალსაზრისით.

2.2 საბორტო კომპიუტერში ოპერანდების დამისამართების ხერხები

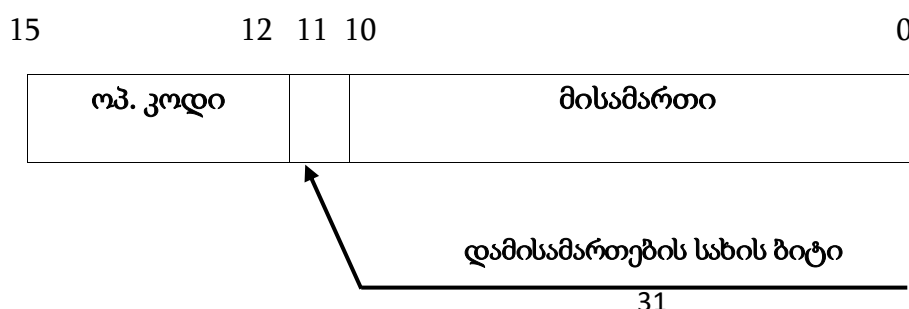
საბორტო კომპიუტერის პროცესორი აღიქვამს ბრძანებათა განსაზღვრულ ნაკადს. პროგრამის შედგენისას დამპროგრამებელი შემოსაზღვრულია ამ ბრძანებებით. ბრძანებათა რაოდენობა და ტიპი დამოკიდებულია საბორტო კომპიუტერის შესაძლებლობებსა და დანიშნულებაზე. იმის და მიხედვით, თუ რომელ ბლოკებთან ხდება ბრძანების მიმართვა ან რომელ ბლოკებზე ის მიეთითება, ბრძანებები შეიძლება დაიყოს სამ ჯგუფად: 1) ბრძანებები რომლებიც მიმართავენ ოპერატიულ ან მუდმივ მეხსიერებას (სამისამართო ბრძანებები); 2) პროცესორის შიგა რეგისტრებს (რეგისტრული ან უმისამართო); 3) შეტანა-გამოტანის რეგისტრებს.

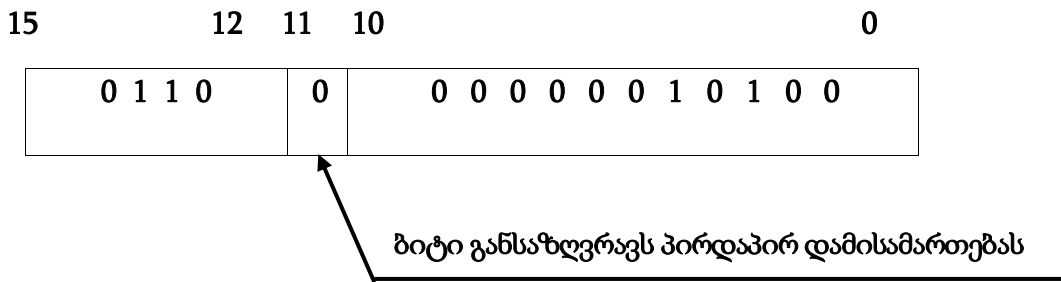
შემსრულებელი მისამართების მოცემის ხერხები ყოველ კონკრეტულ შემთხვევაში შეიძლება იყოს სხვადასხვანაირი, რაც საშუალებას იძლევა ბრძანებების ფიქსირებული ზომის შემთხვევაში მოხდეს ვარირება სამისამართო სივრცით. ვთქვათ, მაგალითად ბრძანების თანრიგობა შეადგენს 2 ბაიტს. ერთმისამართიან ბრძანებაში თანრიგების ნაწილს იკავებს ოპერაციის კოდი, თანრიგების დანარჩენი ნაწილი შეიძლება გამოყენებულ იქნეს ოპერანდის მისამართის მოცემისთვის, ე.ი. მეხსიერების უჯრედის მოცემისთვის, რომელშიც ინახება ინფორმაცია. თუ მეხსიერებას გააჩნია მცირე მოცულობა (მეხსიერების უჯრედების მცირე რაოდენობა), მაშინ გამოიყენება მეხსიერების უჯრედების პირდაპირი დამისამართება. ასეთი დამისამართების დროს ბრძანებაში მიეთითება უშუალოდ შემსრულებელი მისამართი.

მეხსიერების დიდი მოცულობის შემთხვევაში გამოიყენება ირიბი დამისამართება, რომელიც საშუალებას იძლევა არსებითად გაფართოვდეს სამისამართო სივრცე. მოვიყვანოთ მათი შესაბამისი მაგალითები.

ა) პირდაპირი დამისამართება

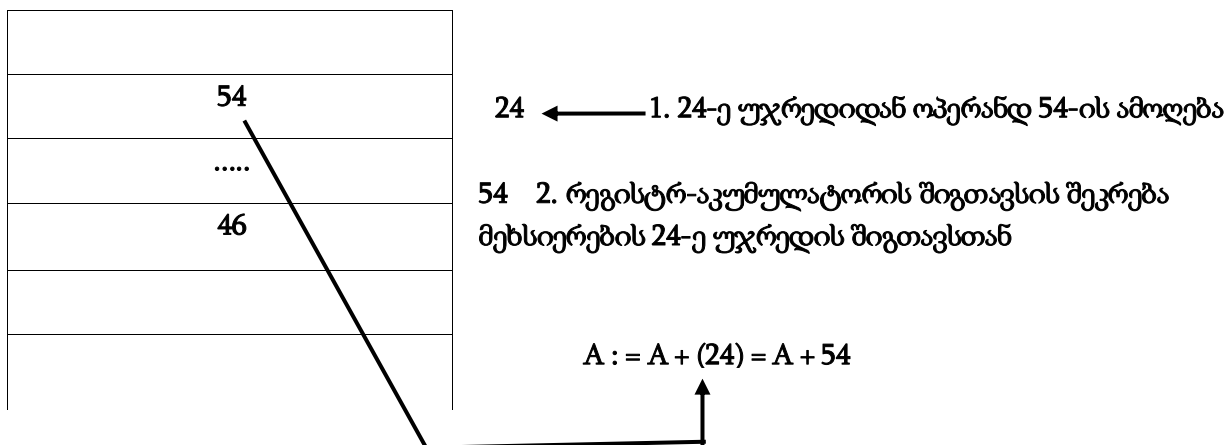
ასეთი დამისამართების ორობითი კოდი, რომელსაც შეიცავს ბრძანება 12-დან 15-ის ჩათვლით განსაზღვრავს ოპერაციის კოდს, ხოლო 0-დან 11 თანრიგის ჩათვლით კი მოიცემა შემსრულებელი მისამართი.





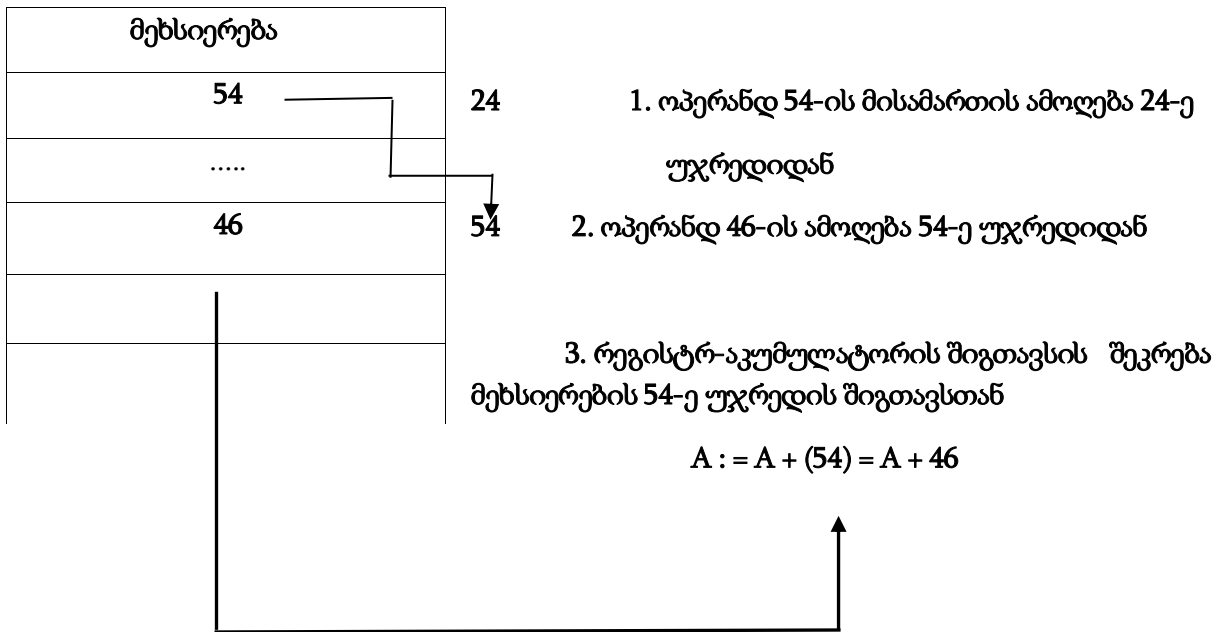
რვაობით სისტემაში ბრძანების კოდი შეიძლება ჩაიწეროს ასეთნაირად 06 0024. ვთქვათ ოპერაციის კოდია 06 და იგი განსაზღვრავს ოპერაციას „შეკრება“. მაშინ მნემონურ სახეში ბრძანების კოდი იქნება ADD 0024, სადაც ADD წარმოადგენს „შეკრება“ ოპერაციის კოდს.

მეხსიერება



ბ) ირიბი დამისამართება





როგორც უკვე ავღნიშნეთ ბრძანების კოდად იწოდება ბრძანების კოდის აგების განსაზღვრული ხერხი, რომელშიც ერთი ნაწილი გამოყოფილია ოპერაციის კოდის წარმოდგენისათვის, ხოლო მეორე კი ოპერანდის მისამართის წარმოდგენისათვის. თუ ბრძანებაში წარმოდგენილია მხოლოდ ოპერაციის კოდი, მაშინ პროგრამა იწოდება უმისამართო ბრძანებად. ასეთი ბრძანება კომპიუტერში იძლევა ისეთ მოქმედებებს, რომლებიც არ მონაწილეობენ ოპერანდის დამუშავებაში ან ოპერანდის ამორჩევაში მისი მისამართის მიხედვით. იმის და მიხედვით ბრძანების სამისამართო ნაწილში რა რაოდენობის მისამართი იქნება მითითებული განასხვავებენ ერთმისამართიან, ორმისამართიან და ა.შ. ბრძანებებს.

ბრძანებები და ოპერანდები (რიცხვები, მონაცემები) ინახება საბოროტო კომპიუტერის მეხსიერებაში განსაზღვრული თანრიგობით. ამიტომ თანრიგების რაოდენობა, რომლებიც განკუთვნილია მათი წარმოდგენისათვის, როგორც წესი, ერთნაირია. მაგალითად, ბრძანებები და ოპერანდები შესაძლებელია წარმოდგენილ იქნეს ორი ბაიტით, ე.ი 16 თანრიგით. ბრძანებებისა და ოპერანდების მოთავსება მეხსიერების რამდენიმე თანმიმდებრობით განლაგებულ უჯრედებში, საშუალებას იძლევა გაიზარდოს ბრძანების ან ოპერანდის თანრიგობა. ასეთ შემთხვევაში ყოველთვის მიეთითება მეხსიერების მხოლოდ პირველი უჯრედის მისამართი, ხოლო მეზობელი მისამართები ფორმირდებიან ავტომატურად.

საბოროტო კომპიუტერში ოპერანდების მისამართები მოიცემა ირიბად სპეციალური რეგისტრების საშუალებით, რომლებიც თავის მხრივ წარმოადგენენ პროცესორის შიგა

მეხსიერებას. ასეთი რეგისტრები მიღებულია იწოდებოდნენ საერთო დანიშნულების რეგისტრებად (სდრ). თუ საბოროტო კომპიუტერში გამოყენებული იქნება 8 რეგისტრი, მაშინ რეგისტრის ნომრის მოსაცემად ბრძანების სამისამართო ნაწილში გამოიყოფა სამთანრიგიანი ველი (კოდი 000-111). საბოროტო კომპიუტერში მიღებულია ოპერანდის მისამართის ფორმირების სხვადასხვა ხერხი. დამისამართების ამა თუ იმ ხერხის მოცემისათვის, რომელიც იწოდება დამისამართების რეჟიმად, გამოიყენება ასევე ბრძანების სამისამართო ნაწილის სამთანრიგიანი ველი (კოდები 000-111).

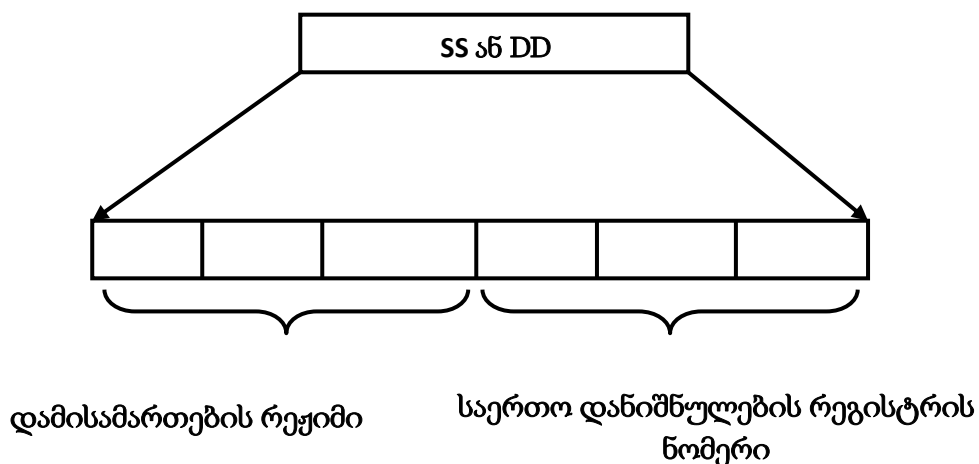
თუ მისამართი განსაზღვრავს ადგილს, საიდანაც ამოიღება ოპერანდი, მაშინ მისამართი იწოდება ინფორმაციის წყაროდ და აღინიშნება SS. თუ მისამართი განსაზღვრავს ადგილს, რომელშიც თავსდება ოპერანდი, მაშინ მისამართი იწოდება ინფორმაციის მიმღებად და აღინიშნება როგორც D

1. ორმისამართიანი ბრძანებების ფორმატები

ა) 15 12 11 6 5 0

ოპ.კოდი	SS	DD

SS და DD ველები აგებულია ისე , რომ განსაზღვრავს როგორც დამისამართების რეჟიმს, ისე შესაბამისი რეგისტრის ნომერს.



ბრძანება სრულდება შემდეგნაირად:

- აღებულ იქნეს პირველი ოპერანდი იმ მისამართის მიხედვით, რომელიც ფორმირებულია SS ველში;
- აღებულ იქნეს მეორე ოპერანდი იმ მისამართის მიხედვით, რომელიც ფორმირებულია DD ველში;

- შესრულდეს მოქმედებები კოდის ოპერაციის შესაბამისად;
- შედეგი მოთავსდეს მისამართზე, რომელიც ფორმირებულია DD ველში.

ბ) 15 12 11 6 5 0

ოპ.კოდი	R	SS ან DD

ბრძანება სრულდება შემდეგნაირად :

- აღებულ იქნეს პირველი ოპერანდი R რეგისტრიდან (და მისი მომდევნო რეგისტრისგან) ;
- აღებულ იქნეს მეორე ოპერანდი იმ მისამართის მიხედვით, რომელიც ფორმირებულია SS ან DD ველში;
- შესრულდეს მოქმედებები ოპერაციის კოდის შესაბამისად;
- შედეგი მოთავსდეს R რეგისტრში ან მისამართზე, რომელიც ფორმირებულია DD ველში.

2. ერთმისამართიანი ბრძანებების ფორმატი

ა) 15 6 5 0

ოპ. კოდი	DD

ბრძანება სრულდება შემდეგნაირად, როცა სამისამართო ველში მოცემულია DD :

- ამოღებულ იქნეს ოპერანდი იმ მისამართის მიხედვით, რომელიც ფორმირებულია DD ველში;
- შესრულდეს მოქმედებები, რომლებიც მოცემულია ოპერაციის კოდით.
- შედეგი მოთავსდეს მისამართზე, რომელიც ფორმირებულია DD ველში.

ბ)

15 8 7 0

ოპერაციის კოდი	XX

ბრძანება სრულდება სხვა ხერხით, თუ სამისამართო ველში მოცემულია XX :

- ბრძანების მთვლელის შიგთავსს, რომელიც განსაზღვრავს მომდევნო ბრძანების მისამართს, დაემატოს რიცხვი ნიშნით XX (წანაცვლება).

- მოხდეს გადასვლა ბრძანების შესრულებაზე, რომლის მისამართი დაფორმირებულია ბრძანებათა მრიცხველში.

გ)

15	3 2	0
ოპერაციის კოდი		R

ამ ბრძანების შესრულებისას ყველა მოქმედება ხორციელდება R რეგისტრის შიგთავსზე.

საბოლოო კომპიუტერში ბრძანებების სხვადასხვა ფორმატების არსებობა ახორციელებს უფრო მოქნილ დაპროგრამებას და ამცირებს ამოცანის ამოხსნის პროგრამის სიგრძეს.

განვიხილოთ ოპერანდების მისამართების ფორმირების შესაძლო ხერხები, რომლებიც განისაზღვრებიან დამისამართების რეჟიმის 3 - თანრიგა კოდით.

კომპიუტერის მეხსიერების სტრუქტურა უზრუნველყოფს ბაიტურ დამისამართებას. თუმცა მეხსიერებიდან მონაცემების ამორჩევა ხდება 2 ბაიტის სახით, რომლებიც ერთიანობაში ქმნიან სიტყვას (ამორჩევის სიგანე). დამისამართება ხორციელდება სიტყვის უმცროსი ბაიტის მიმართ, შესაბამისად, ყველა მუშა მისამართები - წყვილი რიცხვებია. დამისამართებისათვის გამოიყენება 8 რეგისტრი (საერთო დანიშნულების რეგისტრები) ნომრებით 0 დან 7 მდე , ამასთან რეგისტრი მე - 7 ნომრით ასრულებს ბრძანებათა მთვლელის როლს. ბრძანებისა და მათი შიგთავსების ჩაწერისათვის გამოიყენება ათვლის რვაობითი სისტემა.

შემსრულებელი მისამართების ფორმირების წესები და ხერხები დამისამართების რეჟიმების მონაცემის სხვადასხვა ხერხების შემთხვევაში განვიხილოთ ორმისამართიანი ბრძანების **გ ა დ ა გ ზ ა ვ ნ ა** მაგალითზე (ოპერაციის კოდის მნემონური აღნიშვნა ასემბლერში MOV ან რვაობით კოდში - 01). განხილვის სიმარტივის თვალსაზრისით დამისამართების რეჟიმები იცვლება მხოლოდ პირველი ოპერანდის ამორჩევის დროს, ხოლო მეორე ოპერანდის მისამართის ფორმირების დროს მოიცემა დამისამართების ფიქსირებული რეჟიმი, მაგალითად, გამოიყენება რეგისტრული დამისამართება.

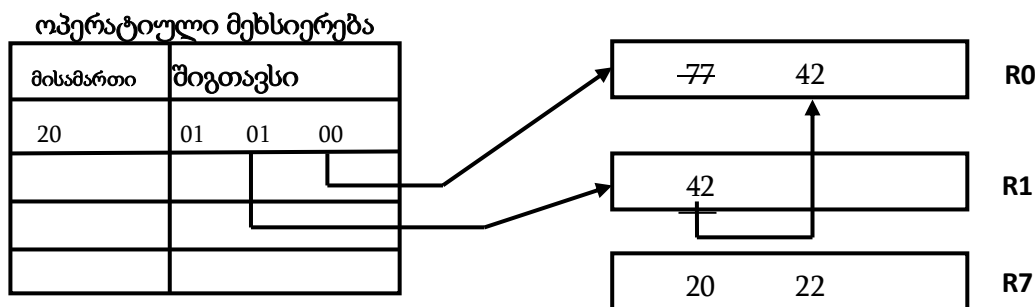
განვიხილოთ დამისამართების რამდენიმე ხერხი.

რეგისტრული დამისამართება

დამისამართების რეჟიმის კოდია (0)₈ ან (000)₂. ოპერანდი, რომლის ნომერი მითითებულია ბრძანების სამისამართო ველში განთავსდება რეგისტრში. ვთქვათ გამოიყენება R0 და R1 რეგისტრები. მაშინ ორმისამართიანი ბრძანების ჩაწერა 1-ლი და მე-2 ოპერანდებისა რეგისტრული დამისამართებით მნემონურ ფორმაში ჩაიწერება ასე

MOV R1, R0.

ნახ. 2.1 - ზე წარმოდგენილ მეხსიერების უჯრედში მისამართით 20 მოთავსებულია ბრძანების კოდი 01 01 00 (ჩაწერა 20 / 01 01 00) ათვლის რვაობით სისტემაში. თუ გადავიყვანთ კოდის მნიშვნელობას ათვლის ორობით სისტემაში თითოეული ციფრის ორობითი ტრიადად შეცვლით გარდა უფროსისა, მაშინ მივიღებთ 16 თანრიგიან კოდს. ბრძანება სრულდება შემდეგნაირად : პირველი ოპერანდი (რიცხვი 42) R1 რეგისტრიდან თავსდება R0 რეგისტრში 77 რიცხვის მაგივრად, ამასთან რიცხვი 42 R1 რეგისტრში ისევ შეინახება. ბრძანებათა მთვლელის შიგთავსი იზრდება სიტყვის სიგრძით ბაიტებში, ე.ი 2 -ით. ბრძანების შესრულების მაგალითი, რომელშიც პირველი და მეორე მისამართი მოცემულია რეგისტრული დამისამართებით მოცემულია ნახ.2.1 - ზე.



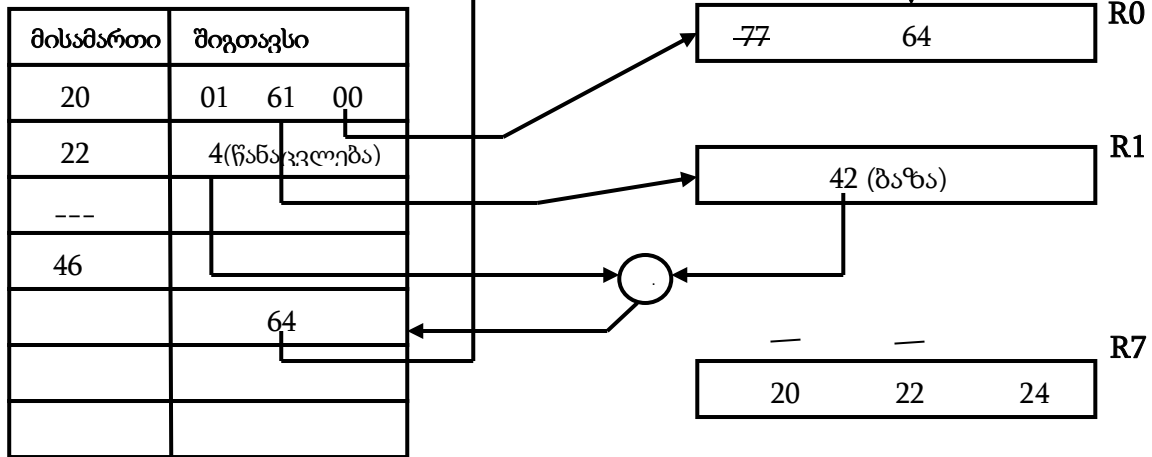
ნახ.2.1

ინდექსური დამისამართება

დამისამართების რეჟიმის კოდია (6)₈ ან (110)₂. ოპერანდი განლაგდება ოპერატიული მეხსიერების უჯრედში, რომლის მისამართი გაიანგარიშება როგორც წანაცვლებისა (ნიშნის რიცხვი) და ბაზის (დადებითი რიცხვი) ჯამი. წანაცვლება მოიცემა ბრძანებაში და განლაგდება ოპერატიული მეხსიერების ორ შემდეგ უჯრედში, ხოლო ბაზა მითითებულია რეგისტრში. ბრძანების მნემონური აღნიშვნაა: MOV Cm(R1)

R0, სადაც Cm - წანაცვლების სიდიდეა. ბრძანების შესრულების სქემა წანაცვლების და ბაზის კონკრეტული მნიშვნელობების ჩვენებით მოცემულია ნახ. 2.2-ზე.

ოპერატიული მეხსიერება

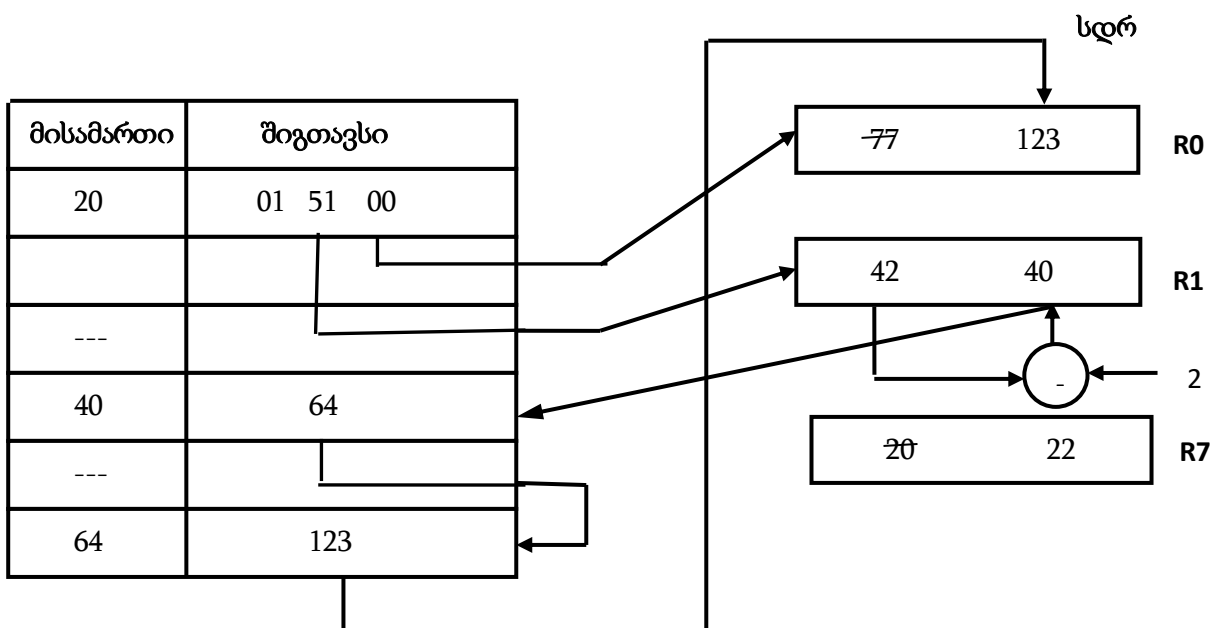


ნახ.2.2

ორიბი ავტოდეკრემენტული დამისამართება

ორიბი ავტოდეკრემენტული დამისამართების რეჟიმის კოდია (15)8 ან (101)2. ოპერანდი განლაგდება ოპერატიული მეხსიერების უჯრედში, რომლის მისამართი ფორმირდება რეგისტრის შიგთავსის შემცირებით 2-ით. ბრძანების მნიშვნელობის კოდია MOV@_(R1),R0 ბრძანების შესრულების სქემა მოცემულია ნახ. 2.3 ზე

ოპერატიული მეხსიერება



ნახ.2.3

2.3. საბორტო კომპიუტერის ბრძანებების კოდებში დაპროგრამების განსაკუთრებულობები

საბორტო კომპიუტერები და სისტემები თავისი არსით არიან მმართველი კომპიუტერები და სისტემები. მათი დანიშნულებაა „საფრენი აპარატი-საბორტო დანადგარი-იარაღი“ სისტემის მართვის ალგორითმების რეალიზაცია დროის რეალურ ან ფორსირებულ მასშტაბში. ამოცანის გადაწყვეტა დროის რეალურ მასშტაბში ნიშნავს გამოთვლების შედეგების მიღებას საწყისი მონაცემების არსებით ცვლილებებამდე. ამისგან უფრო დაჩქარებული ტემპით, მაგრამ სიზუსტის მიხედვით შენარჩუნების მოთხოვნით ეგრეთწოდებულ დროის ფორსირებულ მასშტაბში, წყდება ექსტრაპოლაციის ამოცანები: სამიზნეს ფაზური კოორდინატების პროგნოზირება (საკუთრივ საფრენი აპარატის), სხვადასხვა საბორტო სისტემებისა და ქვესისტემების ფუნქციონირების პროცესის განვითარება.

ალგორითმების რეალიზაცია დროის რეალურ მასშტაბში და მითუმეტეს ფორსირებულ მასშტაბში გამოთვლით სისტემებს უყენებს მაღალ მოთხოვნებს სწრაფქმედების თვალსაზრისით, გადაწყვეტის რიცხვითი მეთოდების ეკონომიურობის თვალსაზრისით, ასევე გამოთვლითი პროცესის ორგანიზაციის სპეციალური პაკეტურ-პრიორიტეტული სისტემების დანერგვის თვალსაზრისითაც.

ჩამოყალიბებული მსჯელობის შესაბამისად საბორტო კომპიუტერის პროგრამული უზრუნველყოფა პრინციპულად განსხვავდება საერთო დანიშნულების კომპიუტერის პროგრამების შექმნის ტექნოლოგიისგან, რომლებიც თავის პროგრამებს რეალიზებას უკეთებენ დროის არა რეალურ მასშტაბში, არამედ ჩვეულებრივ მაღალი დონის ენაზე.

საბორტო კომპიუტერზე ამოცანების გადაწყვეტისათვის მომზადების პროცესი შესაძლებელია დაიყოს შემდეგ ძირითად ეტაპებად:

- დასმული ამოცანის ფორმალიზაცია (ამოცანის მათემატიკური ფორმულირება);
- გადაწყვეტის რიცხვითი მეთოდების არჩევა;
- მასშტაბირება და მანქანურ თანაფარდობების შედგენა;
- ამოცანის ამოხსნის ალგორითმის სტრუქტურული სქემის შედგენა;
- მეხსიერების განაწილება;

- პროგრამების შედგენა პირობითი და ჭეშმარიტი მისამართების შემთხვევაში;
- პროგრამის გამართვა და მისი ექსპლუატაცია.

ამოცანის მათემატიკური ფორმულირება წარმოადგენს რეალური ფიზიკური სიდიდეების, საწყისი მონაცემების და გამოთვლების შედეგების ურთიერთკავშირს. ფორმულური დამოკიდებულებების დამუშავების პროცესში ისმება საკითხი მისაღები განტოლებათა სისტემის განსაზღვრულობის შესახებ, წყდება თუ როგორი ინფორმაციული მხარდაჭერა იქნება საჭირო დამუშავებული ალგორითმისთვის, ფორმირდება მმართველი სისტემის სტრუქტურა.

ამოცანის გადაწყვეტის რიცხვითი მეთოდების არჩევა საბორტო მმართველი კომპიუტერებისთვის გვთავაზობს ძორძოხა კლასიკური გამოსახულებების (ინტეგრალუბის, დიფერენციალური განტოლებების და ა.შ.) შეცვლას ოპერაციების მიხედვით ეკონომიური, ზუსტი და სწრაფადმოქმედი ფორმულებით. ამასთან ერთად, აუცილებელია გამარტივდეს ამოცანა გონივრულ საზღვრამდე, დაცულ იქნეს სიზუსტის მოთხოვნა და ამოცანის მათემატიკური აღწერა.

საბორტო კომპიუტერი ოპერირებს ინფორმაციასთან, რომელიც ძირითადად წარმოდგენილია წესიერი წილადებით შემოსაზღვრული სათანრიგო ბადით. ასეთ შემთხვევაში გადასაწყვეტი ამოცანის ყველა ფიზიკური სიდიდე საბორტო კომპიუტერში წარმოდგება რაღაც მასშტაბებში. მასშტაბი წარმოადგენს შემდეგ დამოკიდებულებას:

$$M = \frac{\bar{x}}{x},$$

სადაც X -ცვლადის ფიზიკური მნიშვნელობაა; $\bar{X}$ -კი ცვლადის წარმოდგენაა საბორტო კომპიუტერში.

მასშტაბური თანაფარდობების საფუძველზე ფიზიკური სიდიდეები გარდაისახება მანქანურში. განვიხილოთ მასშტაბირების პროცესი საბორტო კომპიუტერისთვის ფიქსირებული მძიმის შემთხვევაში სამიზნემდე დაშორების (მანძილის) გაზომვის ამოცანის გადაწყვეტის მაგალითზე სიხშირული მეთოდით. უმარტივეს შემთხვევაში გამოთვლის ფორმულას აქვს ასეთი სახე:

$$D = \frac{c * F_p}{2 * a}, \quad (3.1)$$

სადაც C -რადიოტალის გავრცელების სიჩქარე; $F_p = f_{გამ} - f_{მიღ}$;

a -სიხშირის ცვლილების სიჩქარე.

$F_p = f_{გამ} - f_{მიღ}$ - სხვაობითი სიხშირე გამოსხივებულ და მიღებულ სიგნალებს შორის.

იმდენად რამდენადაც მანქანური სიდიდეების მაქსიმალური მნიშვნელობა არ უნდა აჭარბებდეს ერთს, ამიტომ ცვლადების მასშტაბები (რომლებიც შედიან (3.1) გამოსახულებაში შესაძლებელია განისაზღვროს შემდეგნაირად

$$m_f = 1/f_{max} , \quad m_a = 1/a_{max} . \quad (3.2)$$

რაც შეეხება (3.1) გამოსახულებაში გამოყენებული მუდმივების მასშტაბს, ის შესაძლებელია განისაზღვროს ასეთნაირად:

$$m_k = 1/K - 1_{უმც.თან} , \quad (3.3)$$

სადაც K - კონსტანტას მნიშვნელობაა; $1_{უმც.თან}$ - უმცროსი თანრიგის ერთიანი (წიშნადი ციფრი, რომელიც განსაზღვრავს გაზომვის სიზუსტეს ან კონსტანტის ჩაწერის სიზუსტეს).

თუ (3.2) ფორმულაში ჩავსვამთ ფიზიკურ ცვლადებს, რომლებიც გამოსახულია მასშტაბის საშუალებით და (3.1) გამოსახულებით მოცემულ მანქანური ცვლადების შესაბამისად, მივიღებთ

$$\bar{D} = \frac{m_{k1} - m_f * \bar{c}f_{გამ} - f_{მიღ}}{m_{k2} * m_a * \bar{2} * \bar{a}}$$

$$\bar{D} = K_D \frac{\bar{C} \cdot f_{გამ} - f_{მიღ}}{\bar{2} \cdot \bar{a}} . \quad (3.4)$$

ეს გამოსახულება თავის მხრივ წარმოადგენს სიხშირული მეთოდით სიშორის გაზომვის ამოცანის თანაფარდობას, რომლის შესაბამისად დგება პროგრამა. ყველა ცვლადები ამ გამოსახულებებში არ აჭარბებს ერთს. K_D - წარმოადგენს მასშტაბურ კოეფიციენტს.

არითმეტიკული ოპერაციების შესრულება უნდა მიმდინერეობდეს ისეთი თანმიმდევრობით, რომ გამოთვლის თითოეულ ეტაპზე არ უნდა წარმოიშვას გადავსება.

K_D -კოეფიციენტი შესაძლებელია მასშტაბის კორექტირების საფუძველზე შეიცვალოს ისე, რომ მისი მნიშვნელობა ტოლი იყოს $2^{\pm m}$ -ის. ამის შემდეგ გამოითვლება გამოსახულება

$$\bar{D} = \frac{\bar{C} \cdot \overline{f_{გამ} - f_{მიღ}}}{\bar{Z} \cdot \bar{a}},$$

რომელიც m თანრიგზე არითმეტიკული ძვრის ოპერაციის დახმარებით მრავლდება $2^{\pm m}$ კოეფიციენტზე. ასეთ შემთხვევაში გამოირიცხება K_D კოეფიციენტის შენახვა მეხსიერებაში.

საბორტო კომპიუტერისთვის ფიქსირებული მძიმით პროგრამების შედგენისას აუცილებელია არითმეტიკული ბრძანებების შესრულებისთვის გათვალისწინებულ იქნეს შემდეგი მოთხოვნები:

1. ოპერანდების შეკრების ან გამოკლების წინ მათი მასშტაბები უნდა იყოს გათანაბრებული;
2. გაყოფის შესასრულებლად აუცილებელია, რომ გასაყოფი აბსოლუტური სიდიდით ნაკლები იყოს გამყოფზე: განაყოფის მასშტაბის ხარისხის მაჩვენებელი ტოლია გასაყოფისა და გამყოფის ხარისხის მაჩვენებლების სხვაობისა.
3. გამრავლების ოპერაციის შესრულების შედეგს გააჩნია მასშტაბი, რომელიც ტოლია სიმრავლისა და მამრავლის მასშტაბების ნამრავლისა.

კომპიუტერის ამოცანის გადაწყვეტისათვის ამოცანის მომზადების შემდეგ ანალიზდება ამ ამოცანის ამოხსნის ლოგიკა და ამ ანალიზის შედეგად დგება ალგორითმის სტრუქტურული სქემა.

ალგორითმი - ეს არის მოქმედებათა განსაზღვრული თანმიმდევრობა, საწყის მონაცემებზე მათემატიკური ოპერაციების შესრულება, რომელთა განხორციელება იძლევა შედეგის მიღებას. ალგორითმი შეიძლება აღიწეროს:

- სიტყვიერად, მათემატიკური თანაფრდობების გამოყენებით;
- ბლოკ-სქემების დახმარებით ოპერატორულ ფორმაში;
- ალგორითმულ ენაზე.

საბორტო კომპიუტერებისათვის პროგრამების შედგენისას კიდევ ერთ აუცილებელ მოთხოვნას წარმოადგენს მეხსიერების რაციონალური განაწილება. უნივერსალური კომპიუტერებისგან განსხვავებით, საბორტო კომპიუტერში ბრძანებათა კოდები შეიძლება იმყოფებოდეს მხოლოდ გრძელვადიან მეხსიერებაში, ამიტომ ბრძანებათა მასივები და საწყისი მონაცემების, შუალედური და საბოლოო შედეგების მასივები შესაძლებელია იმყოფებოდეს მეხსიერების სხვადასხვა ბლოკებში. საბორტო კომპიუტერში A-15 გარდა აღნიშნულისა შესაძლებელია გამოვიყენოთ დამისამართების სხვადასხვა სახეები შემსრულებელი მისამართის ფორმირებისთვის. დამისამართების სხვადასხვა ნიშან-თვისებების გამოყენება საშუალებას იძლევა არსებითად შემცირდეს პროგრამაში გამოყენებული ბრძანებების რიცხვი და შესაბამისად მისი შესრულების დრო.

საბორტო კომპიუტერის ბრძანებების კოდში დაპროგრამებისას თავდაპირველად დაპროგრამებას აწარმოებენ პირობით (სიმბოლურ) მისამართებში, რადგან წინასწარ ცნობილი არაა თუ რამდენ უჯრედს დაიკავებს პროგრამა. ეს აუცილებელია კიდევ იმიტომ, რომ რთული პროგრამების დამუშავებისას ალგორითმის ნაწილები შეიძლება დაპროგრამირდეს რამდენიმე შემსრულებლის (პროგრამისტის) მიერ ცალ-ცალკე, ხოლო შემდეგ მოხდეს ამ ნაწილების ერთმანეთთან შეპირაპირება (გაერთიანება).

პირობითი მისამართი წარმოადგენს მასივის უჯრედის მისამართს, რომელიც ათვლილია რაღაც ჯერ კიდევ უცნობი საწყისი მისამართიდან, რომლის შემდეგ იწყება მასივი. ეს საწყისი მისამართი იწოდება საბაზისო მისამართად და აღინიშნება ასოთი, რადგან დაპროგრამების განსაზღვრულ ეტაპამდე მისი კონკრეტული მნიშვნელობა უცნობია. ჩვეულებრივ განიხილება შემდეგი მასივები: ბრძანებების მასივი, საწყისი მონაცემების და კონსტანტების მასივები, შუალედური შედეგების მასივები, საბოლოო შედეგების მასივები. თითოეულ მასივს მიენიჭება თავისი საბაზისო მისამართი. მაგალითად, K-ბრძანებების საბაზისო მისამართი, C-კონსტანტების საბაზისო მისამართი, P-საშუალო შედეგების საბაზისო მისამართი, R-საბოლოო შედეგების საბაზო მისამართი.

პირობითი მისამართები წარმოადგება საბაზო მისამართისა და კონსტანტის ჯამის სახით (რიცხვები და ასოები). მაგალითად, პირობითი მისამართი K+1 არის პირველი

ბრძანების მისამართი ბრძანებების მასივში, ხოლო პირობითი მისამართი C+5 მიუთითებს კონსტანტების მასივში მე-5 კონსტანტას.

საბაზისო მისამართებს და შესაბამისად პირობით მისამართებს დაპროგრამების ბოლოს, როცა უკვე ცნობილია თითოეული მასივის მოცულობა, ენიჭებათ რიცხვითი მნიშვნელობები.

პირობითი მისამართების გამოყენებისას საბორტო კომპიუტერის მეხსიერება გამოიყენება უფრო ეფექტურად, იმდენად რამდენადაც ამ შემთხვევაში მასივებს შორის ფაქტობრივად გამორიცხულია მეხსიერების ის უჯრედები, რომლებიც არ არიან ამოქმედებული პროგრამის მუშაობის დროს. რა თქმა უნდა, ამ დროს უნდა გვახსოვდეს მეხსიერების მოცულობის შეზღუდულობის ფაქტობრივად არსებობის შესაძლებლობის შესახებ და საჭიროა არ გადავაჭარბოთ მას.

პირობით მისამართებში დაპროგრამების დამთავრების შემდეგ ხორციელდება ჭეშმარიტ მისამართებში დაპროგრამებაზე გადასვლა. პირველ რიგში საბაზისო მისამართებისთვის მიიღება კონკრეტული მნიშვნელობები.

პროგრამის გამართვის ეტაპზე მოწმდება მოცემული ალგორითმის პროგრამის შესრულების სისწორე და პროგრამის ნაწილებს შორის ლოგიკური და დროითი ურთიერთქმედების სისწორე. ექსპლუატაციის ეტაპზე მოწმდება პროგრამების მუშაობისუნარიანობა და საექსპლუატაციო მახასიათებლები ფუნქციონირების რეალურ პირობებში. წარმოებს მუშაობა პროგრამებში აუცილებელი ცვლილებების შეტანის თვალსაზრისით, რათა მოხდეს აღმოჩენილი შეცდომების გასწორება, გაუმჯობესდეს საავიაციო რადიოელექტრონული კომპლექსში შემავალი მოწყობილობების მახასიათებლები და ფუნქციური შესაძლებლობები, მათ შორის ურთიერთკავშირისა და შემაღენლობის ცვლილება, რომელთა მართვა ხორციელდება საბორტო კომპიუტერის საშუალებით.

2.4. საბორტო კომპიუტერის ბრძანებების სისტემა

მანქანური ენა ან ოპერატორების ერთობლიობა, რომლებიც სრულდება საბორტო კომპიუტერის პროცესორის მიერ, შეადგენს საბორტო კომპიუტერის ბრძანებათა სისტემას. ბრძანებათა სისტემა მნიშვნელოვან გავლენას ახდენს საბორტო კომპიუტერის პროცესორის სტრუქტურისა და მუშაობის წესრიგზე, ასევე მთლიანობაში საბორტო

კომპიუტერის ტექნიკურ (მწარმოებლურობა) და ეკონომიკურ (ღირებულება) მახასიათებლებზე. საბორტო კომპიუტერების ბრძანებათა სისტემის არჩევა ხორციელდება შემდეგი ძირითადი მოთხოვნების გათვალისწინებით:

- ბრძანებათა სისტემის ფუნქციური სისრულით;
- საბორტო კომპიუტერის მაქსიმალური მწარმოებლურობის უზრუნველყოფით;
- საბორტო კომპიუტერის მეხსიერების მინიმიზაციით, რომელიც საჭიროა პროგრამებისა და მონაცემების დამახსოვრებისათვის.

ბრძანებათა სისტემის ფუნქციური სისრულის მოთხოვნა ნიშნავს მისი დახმარებით პროგრამების შექმნის (დაწერის) შესაძლებლობას, რომლებიც ახორციელებენ მართვის ნებისმიერი მოცემული ალგორითმის რეალიზაციას საბორტო მართვის თვალსაზრისით.

საბორტო კომპიუტერისადმი წაყენებული მოთხოვნები მაქსიმალური მწარმოებლურობის უზრუნველსაყოფად აუცილებელს ხდის, რომ მისი ბრძანებათა სისტემა ორიენტირებული იყოს ისეთი ოპერაციების შესრულებისთვის, რომლებიც ნიშანდობლივი და შესაბამისია კონკრეტული საგნობრივი სფეროსთვის.

ბრძანებათა სისტემის შემადგენლობის განსაკუთრებულობის შესახებ საკითხი განვიხილოთ A-15A საბორტო კომპიუტერის ბრძანებათა სისტემის მაგალითზე. A-15A საბორტო კომპიუტერის ბრძანებათა სისტემა შეიცავს 71 ბრძანებას. ბრძანების ფორმატში ოპერაციის კოდისთვის გამოყოფილია 6 თანრიგი, რომელიც საშუალებას იძლევა დაფორმირდეს ოპერაციის 64 სხვადასხვა კოდი. ყველა ბრძანება არ მოითხოვს ოპერანდის მისამართის მითითებას, ამიტომ ბრძანების სამისამართო ნაწილის თავისუფალი თანრიგები შესაძლებელია გამოყენებულ იქნეს ოპერაციის კოდის გაფართოებისთვის. ამჟამად ამოქმედებულია 59 ოპერაციის კოდი, ხოლო 5 ვარიანტი კი თავდარიგშია.

ყველა ბრძანება დაყოფილია 12 ჯგუფად:

- არითმეტიკული ოპერაციები;
- ლოგიკური ოპერაციები;
- ბიტური ოპერაციები;
- გადაგზავის ოპერაციები;
- ჩაწერის ოპერაციები;
- მართვის ოპერაციები;

- ინფორმაციის გაცვლის ოპერაციები გამოთვლითი მოწყობილობის რეგისტრებს შორის;
- შეტანა-გამოტანის და ნორმალიზაციის ოპერაციები;
- ფუნქციების გამოთვლის ოპერაციები;
- სიგნალების დაყენების ოპერაციები;
- ოპერაციები რეგისტრების შიგთავსებზე;
- სპეციალური ბრძანებები.

არიტმეტიკული ოპერაციების შემადგენლობაში ჩვეულებრივ შედის შეკრებისა და გამოკლების ბრძანებები, ორმაგი სიგრძის შეკრებისა და გამოკლების ბრძანებები, ასევე შენაცვლებით, რომლის დროსაც ხორციელდება შედეგის ჩაწერა ოპერატიული მეხსიერების იმ უჯრედში, საიდანაც ამოღებული იყო ერთ-ერთი ოპერანდი. ასევე მის შემადგენლობაში შედის გამრავლებისა და გაყოფის ბრძანებები. კერძოდ ორი სახის: ჩვეულებრივი და დამრგვალებითი.

გარდა ამისა არის მოდულების გამოკლების ბრძანებები და ბრძანება, რომელიც განსაზღვრავს მეხსიერების უჯრედის შიგთავსის ნიშანს.

ლოგიკური ოპერაციები უზრუნველყოფენ ბულის ალგებრის ძირითადი ოპერაციების შესრულებას: ლოგიკური გამრავლება, ლოგიკური შეკრება, შეკრება 2-ის მოდულით ორ მრავალთანრიგა სიტყვების შიგთავსზე.

ბიტური ოპერაციები გამოიყენება სიტყვის ცალკეული თანრიგებზე მოქმედებების შესასრულებლად, როცა სიტყვის მისამართი მითითებულია ბრძანების სამისამართო ნაწილში. ამასთან შესაძლებელია თანრიგების ნულში ან ერთიანში დაყენება. ასევე შესაძლებელია მოცემული სიტყვის კონკრეტული თანრიგის ლოგიკური გამრავლება და ლოგიკური შეკრება შედეგის ნიშან-თვისების მნიშვნელობით **ა**.

გადაგზავნისა და ჩაწერის ოპერაციების დანიშნულებაა მოცემული უჯრედების შიგთავსის გადაგზავნა პროცესორის ძირითად რეგისტრებში და რეგისტრების შიგთავსის ჩაწერა საბორტო კომპიუტერის მეხსიერებაში.

მართვის ბრძანებების დანიშნულებაა ოპერაციათა შესრულების თანმიმდევრობის შეცვლა და შეიცავს პირობითი და უპირობო გადასვლის ბრძანებებს.

გაცვლის ოპერაციას და ფუნქციას გამოთვლის ოპერაციებს გააჩნიათ ოპერაციის ერთი კოდი 16 და უზრუნველყოფენ პროცესორების ძირითად რეგისტრებს შორის

ინფორმაციის გადაგზავნას, ასევე ფუნქციების სინუსი, კოსინუსის გამოთვლას, კვადრატული ფესვის ამოღებას არგუმენტიდან, რომელიც მოთავსებულია ამჯამავის რეგისტრში.

შეტანა-გამოტანის ოპერაციები ემსახურება საბორტო კომიუტერისა და შემტან-გამომტან მოწყობილობებს შორის ინფორმაციის გაცვლას.

სიგნალების დაყენების ოპერაციები და რეგისტრების შიგთავსზე მოქმედი ოპერაციები გამოიყენება იმ რეგისტრების შიგთავსისა და სხვადასხვა ნიშან-თვისების მნიშვნელობის შესაცვლელად, რომლებიც ემსახურება წყვეტის რეჟიმს.

სპეციალურ ბრძანებებს მიეკუთვნება: უქმი ბრძანება (ბრძანება, რომელიც არ ასრულებს რაღაც მოქმედებას და უზრუნველყოფს გამოთვლების დაყოვნებას ერთი ბრძანების შესრულების დრომდე); ბრძანება, რომელიც მეხსიერების მოცემული უჯრედის შიგთავსს ზრდის ერთით უმცროს თანრიგში; მოდიფიკაციის ბრძანება, რომელიც ემსახურება მის შემდგომი ბრძანების ცვლილებას მისი კოდის შეჯამების გზით იმ უჯრედის კოდთან, რომლის მისამართი მითითებულია მოდიფიკაციის ბრძანებაში.

ლექცია 3

საბორტო კომპიუტერის პროცესორის მართვისა და სინქრონიზაციის მოწყობილობა. მმართველი სიგნალების ფორმირება

3.1. საბორტო კომპიუტერის მუშაობის მართვის იერარქიული სტრუქტურა

საბორტო კომპიუტერის პროცესორად იწოდება ისეთი მოწყობილობა, რომელიც ასრულებს პროგრამით მოცემული ინფორმაციის გარდაქმნას, ახორციელებს მთელი გამოთვლითი პროცესის მართვას და ურთიერთქმედებს გამოთვლითი სისტემის სხვა მოწყობილობებთან. ფუნქციების შესრულების მიხედვით პროცესორი წარმოადგენს ცენტრალურ მოწყობილობას ნებისმიერი კომპიუტერისთვის.

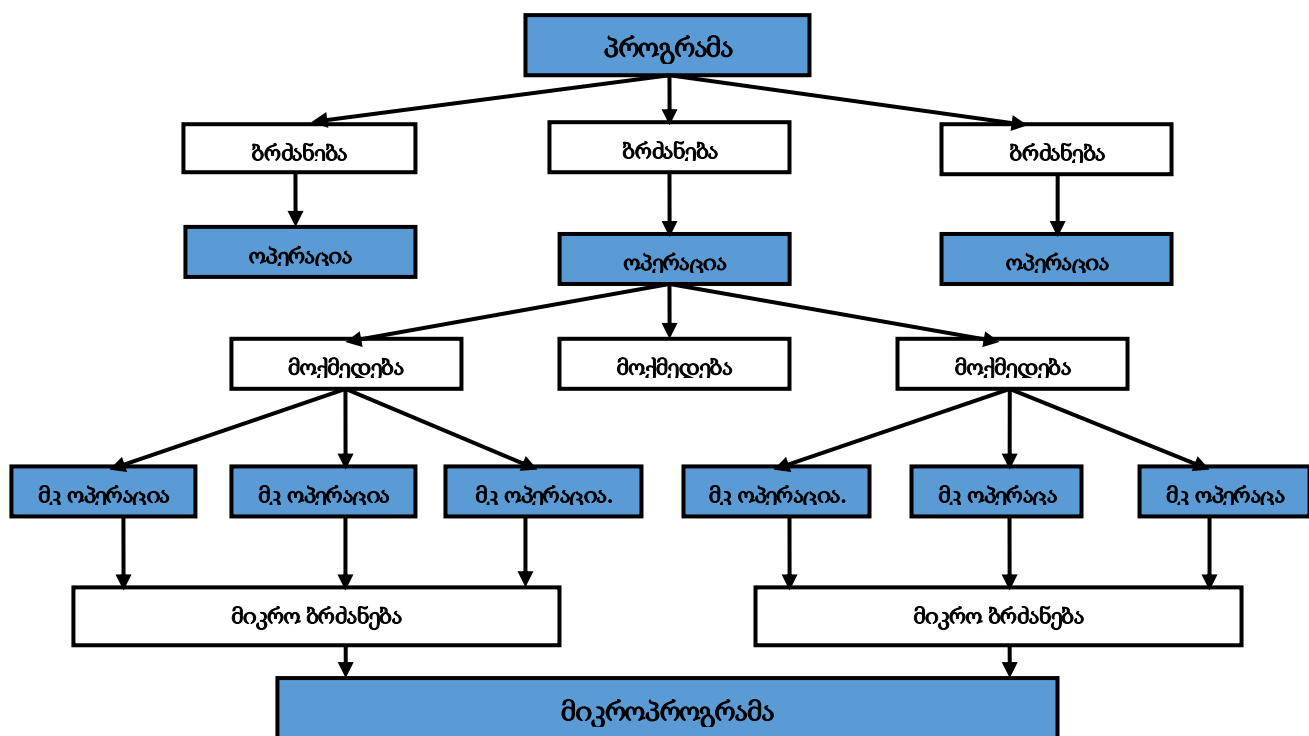
პროცესორი შეიცავს ოპერაციულ და მმართველ ნაწილს. პროცესორის ოპერაციული ნაწილი უზრუნველყოფს ოპერანდებზე მოქმედებას, ე.ი. ოპერაციის მოცემული კოდით ასრულებს ოპერაციებს. ოპერაციულ ნაწილს ჩვეულებრივ უწოდებენ არითმეტიკულ-ლოგიკურ მოწყობილობას. იმისთვის, რომ კოდებზე მოვახდინოთ ოპერაციის რეალიზაცია, საჭიროა მმართველი სიგნალები მიეწოდოს პროცესორის ოპერაციულ ნაწილს არითმეტიკული მოწყობილობის შესაბამის კვანძებში, დროის განსაზღვრულ მომენტებში. ასეთი მოწყობილობა შედის პროცესორის შემადგენლობაში, რომელიც მმართველი სიგნალების თანმიმდევრობით მიწოდების საშუალებით ახორციელებს ოპერაციული მოწყობილობის კვანძების მუშაობის იძულებით კოორდინაციას და მას მართვისა და სინქრონიზაციის მოწყობილობა ეწოდება.

საბორტო კომპიუტერის მუშაობის თანმიმდევრობა განისაზღვრება კონკრეტული ამოცანის ამოხსნის პროგრამით. პროგრამა თავის მხრივ წარმოადგენს ბრძანებების ერთობლიობის მოწესრიგებულ თანმიმდევრობას. ყოველი ბრძანება განსაზღვრავს კომპიუტერის მოქმედებას, რომელიმე ოპერაციის შესრულებისთვის. მაგალითად, შეკრების ოპერაცია სრულდება ერთი ბრძანებით, გამოკლების - მეორე ბრძანებით, გამრავლების - მესამით და ა.შ. ერთი ოპერაციის შესრულების დრო (ან ერთი ბრძანების მოქმედების დრო) იწოდება მანქანის მუშაობის ციკლად. როგროც წესი მანქანის მუშაობის ციკლი ემთხვევა მართვის მოწყობილობის მუშაობის ფუნქციონირების ციკლს.

ოპერაცია შეიძლება დაყოფილ იქნეს უფრო მცირე ეტაპებად, რომელთა დროს სრულდება განსაზღვრული ელემენტარული მოქმედებები: მაგალითად, ციკლის განმავლობაში ორი რიცხვის შეკრების ოპერაცია შესაძლებელია განვიხილოთ, როგორც შემდეგ მოქმედებათა ერთობლიობა: 1) მეხსიერებიდან შეკრების ბრძანების ამორჩევა და ოპერაციის კოდის გაშიფრვა; 2) ბრძანებაში მოცემული მისამართების შესაბამისად ოპერანდების ამორჩევა მეხსიერებიდან; 3) არითმეტიკულ-ლოგიკურ მოწყობილობაში ოპერანდების შეკრების ოპერაციის შესრულება. 4) ოპერატიულ მეხსიერებაში ან პროცესორის შიგა რეგისტრებში შედეგის შეტანა.

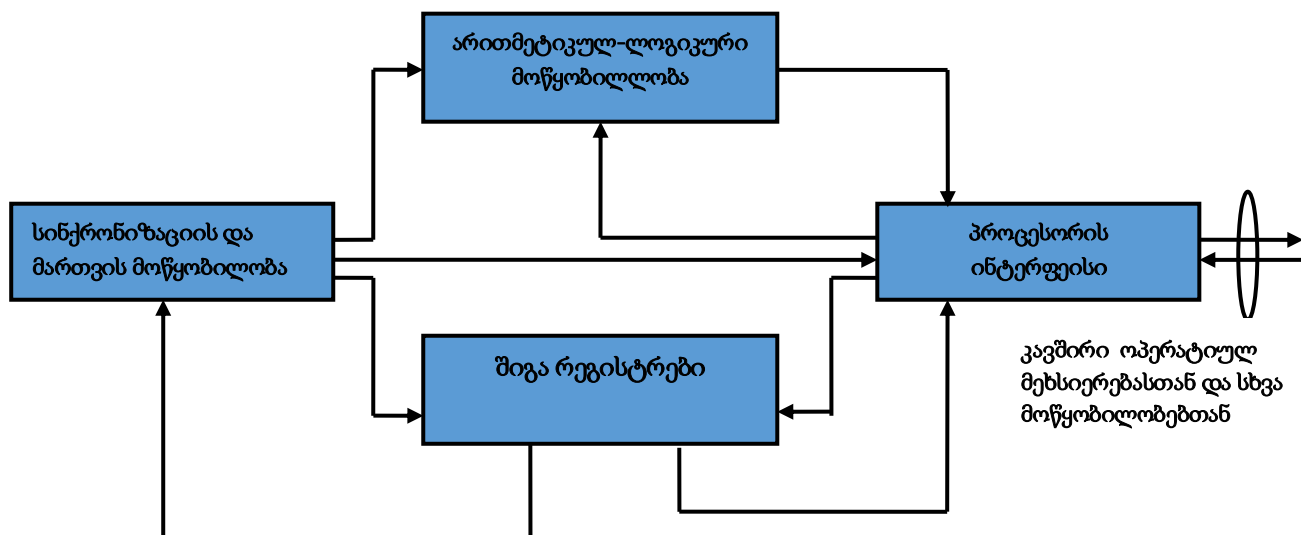
ყოველი მოქმედება წარმოადგენს მიკროოპერაციების ერთობლიობას. მიკროოპერაცია სრულდება ერთი მმართველი სიგნალის ზემოქმედებით. ერთი მმართველი სიგნალი ყოველთვის ასრულებს ლოგიკურ დამოკიდებულებაში ერთ ფუნქციას, თუმცა მას შეუძლია ზემოქმედება მოახდინოს სხვადასხვა ელექტრულ წრედებზე. ურთიერთთავსებადი მიკროოპერაციების ერთდროულად შესრულების დროს ფორმირდება რამდენიმე მმართველი სიგნალი. ამ სიგნალების ნაკრებს, რომლებიც ფორმირდებიან ერთდროულად და სრულდებიან ერთი ტაქტის განმავლობაში, იწოდება მიკრობრძანებად. მიკრობრძანებების მიმდევრობა, რომლებიც აუცილებელია ერთი მანქანური ოპერაციის რეალიზაციისთვის შეადგენს ამ ოპერაციის შესრულების მიკროპროგრამას და სრულდება ციკლის განმავლობაში. ამრიგად, მიკროპროგრამა განსაზღვრავს პროცესორის კვანძების მუშაობის თანმიმდევრობას, ასევე კომპიუტერის სხვა მოწყობილობების მუშაობასაც კონკრეტული ოპერაციის შესრულების დროს.

საბოროტო კომპიუტერის მუშაობის მართვის იერარქიული სტრუქტურა პროგრამის შესრულების დროს წარმოდგენილია ნახ.3.1-ზე.



ნახ. 3.1. საბოროტო კომპიუტერის მუშაობის მართვის იერარქიული სტრუქტურა
პროგრამის შესრულებისას

ზემოთ თქმულიდან გამომდინარე პროცესორის გამართივებული სტრუქტურული
სქემა შესაძლებელია წარმოდგენილ იქნეს ისე, როგორც ეს ნაჩვენებია ნახ. 3.2-ზე.



ნახ.3.2

როგორც ავღნიშნეთ, მართვისა და სინქრონიზაციის მოწყობილობა წარმოადგენს
პროცესორის მმართველ ნაწილს და უზრუნველყოფს ბრძანებების მიღებას, შენახვას და
გაშიფვრას შემდეგში მმართველი სიგნალების გამომუშავებით.

არითმეტიკულ-ლოგიკურ მოწყობილობა თავის მხრივ წარმოადგენს პროცესორის ოპერაციულ ნაწილს და ემსახურება ოპერანდების კოდებზე არითმეტიკული და ლოგიკური ოპერაციების შესრულებას. ამიტომ ალმ-ის შემადგენლობაში შედის ამჯამავე, რეგისტრები და ლოგიკური სქემები.

შიგა რეგისტრები ქმნიან პროცესორის შიგა მეხსიერებას და უზრუნველყოფენ დასამუშავებელი და მმართველი ინფორმაციის დროებით დამახსოვრებას. უმეტესი მიკროპერაციები ახორციელებს მონაცემებზე მოქმედებებს ალმ-ის რეგისტრებში ან მმართველ რეგისტრებში. მმართველი რეგისტრების ნაწილი წარმოადგენს პროგრამულად ხელმისაწვდომს, ე.ი. შეიძლება დამისამართდეს ბრძანებაში ოპერატიული მეხსიერების უჯრედების მსგავსად.

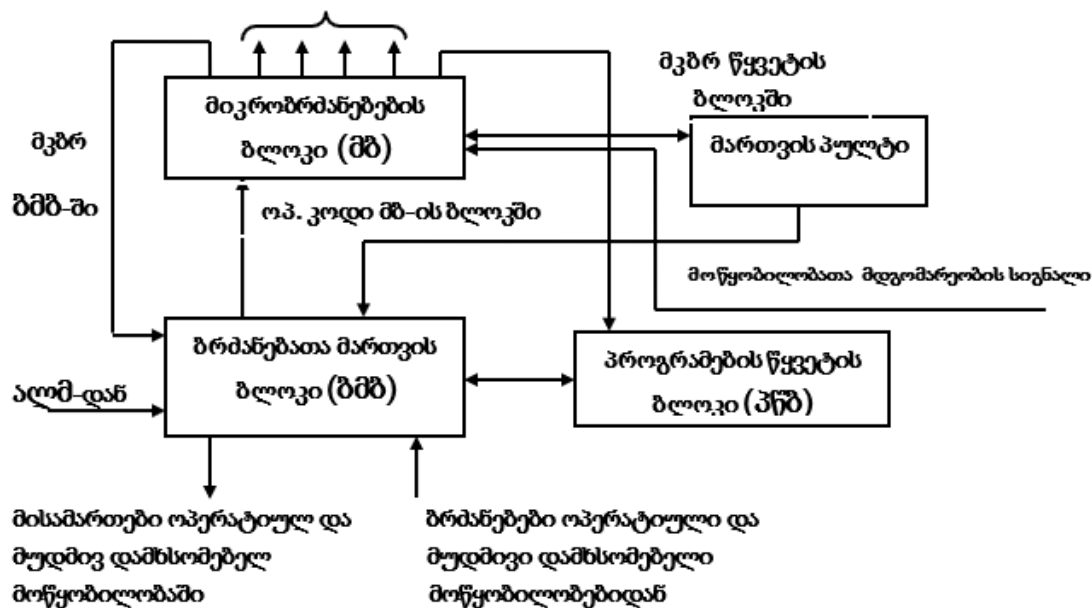
პროცესორის შიგა მეხსიერების არსებობა საშუალებას იძლევა არსებითად შემცირდეს ოპერაციათა შესრულების დრო.

პროცესორის ინტერფეისი უზრუნველყოფს ინფორმაციის გაცვლას საბორტო პროცესორსა და კომპიუტერის ოპერატიულ (მუდმივ) მეხსიერებას შორის. ასევე პროცესორსა და სხვა მოწყობილობებს შორის, რომლებიც შეიძლება მიერთებული იყოს პროცესორთან.

3.2. საბორტო კომპიუტერის მართვის მოწყობილობისა და სინქრონიზაციის განზოგადებული სტრუქტურული სქემა

სხვადასხვა ტიპის საბორტო კომპიუტერებში მართვის და სინქრონიზაციის მოწყობილობებს აქვთ სხვადასხვა ტიპის სტრუქტურა, რომელიც განპირობებულია ბრძანებათა სისტემებში განსხვავებით, ასევე მმართველი სიგნალების ფორმირებისა და მოწყობილობთა მუშაობის სინქრონიზაციის პრინციპებით. თუმცა ფუნქციები რომლებსაც მართვის მოწყობილობა ასრულებს უმეტესი საბორტო კომპიუტერებისთვის ერთნაირია, რაც საშუალებას იძლევა წარმოდგენილ იქნეს ის განზოგადებული სტრუქტურული სქემის სახით, რომელიც შეიცავს ბლოკებს ტიპური ფუნქციებით. ასეთი სქემა ნაჩვენებია ნახ. 3.3-ზე.

მიკრობრძანებები ა.ლ.მ.-ში, ოპერატიულ, მუდმივ დამსხმებელ და შეტანა-გამოტანის მოწყობილობაში



ნახ.3.3

მომდევნო ბრძანების ამორჩევის, ციკლის განმავლობაში მისი შენახვის, ბრძანების სამისამართო ნაწილის ოპერანდების შესაბამის მისამართებში გარდაქმნის, ასევე ოპერაციის კოდის გამოყოფის ფუნქციის შესასრულებლად არსებობს ცალკე ბლოკი, რომელიც იწოდება ბრძანებების მართვის ბლოკად.

შესასრულებელი ოპერაციების შესაბამისად (ოპერაციის კოდის მიხედვით) მმართველი სიგნალების ფორმირებისთვის გამოიყენება სპეციალური ბლოკი რომელსაც მიკრობრძანებების ბლოკი ეწოდება.

საბოროტო კომპიუტერის მრავალპროგრამული მუშაობის ორგანიზაციისთვის გამოიყენება პროგრამების წვევტის ბლოკი.

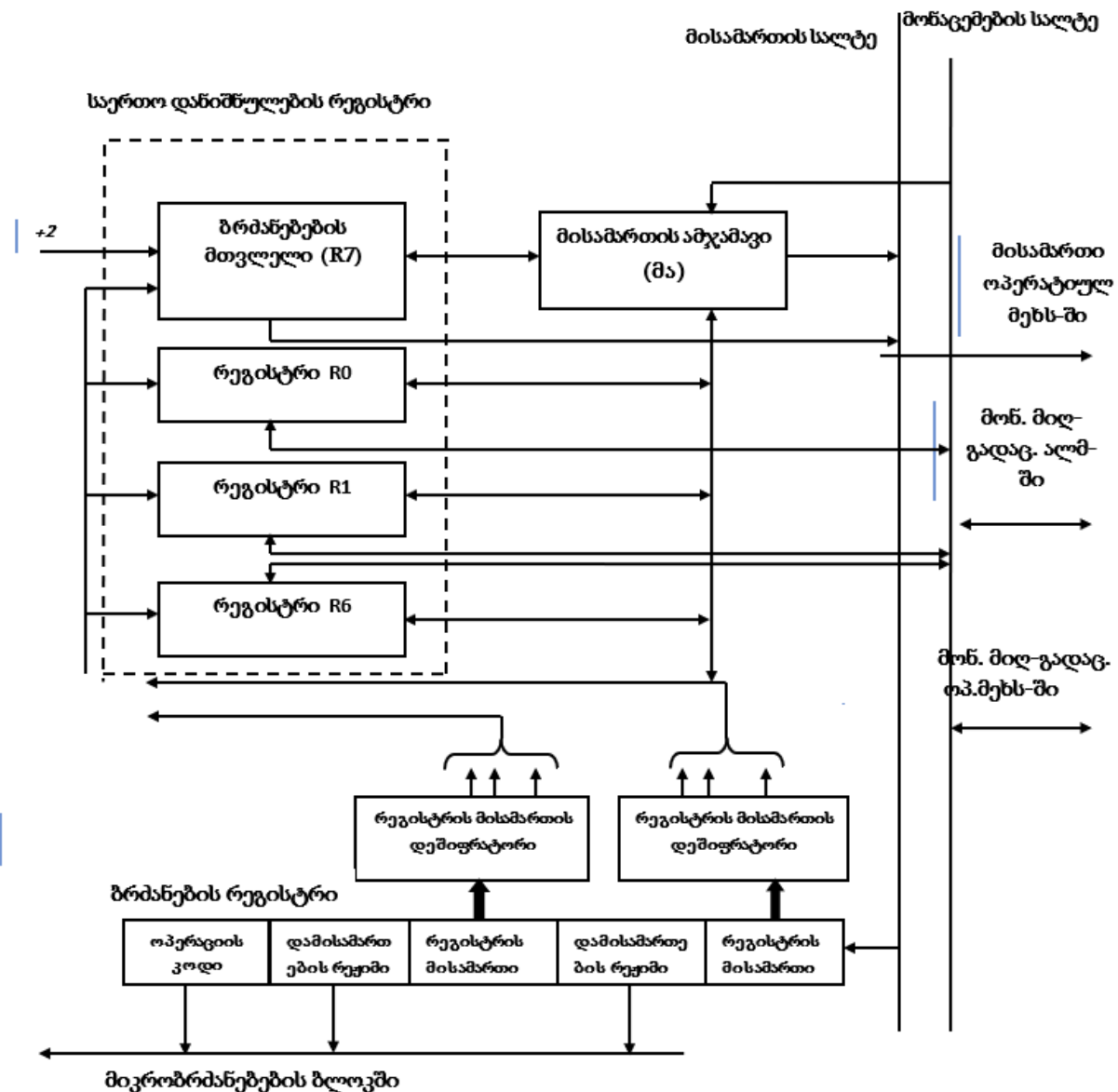
მართვის პულტი, როგორც მართვის და სინქრონიზაციის მოწყობილობის ნაწილი, აუცილებელია ოპერატორის კომპიუტერთან კავშირისთვის და საშუალებას იძლევა მიმდინარე პროცესებზე მოხდეს ზემოქმედება ხელით.

3.3. ბრძანებების მართვის ბლოკის აგების სქემა და შემადგენლობა

ბრძანებების მართვის ბლოკის მიერ ზემოთ აღწერილი ფუნქციების შესრულებისთვის საჭიროა შემდეგი კვანძები და სქემები:

- ბრძანებების მთვლეელი, რომელიც ახორციელებს მიმდინარე ბრძანების ამორჩევას მასში მითითებული ბრძანებების მიხედვით და უზრუნველყოფს მომდევნო ბრძანების მისამართის ფორმირებას;
- ბრძანებების რეგისტრი, რომელიც განკუთვნილია მიმდინარე ბრძანების კოდის შენახვისათვის მისი შესრულების ციკლის განმავლობაში და ბრძანების სამისამართო და ოპერაციული ნაწილის გამოყოფისათვის;
- მრავალმიზნობრივი დანიშნულების რეგისტრების ჯგუფი, რომელიც ბრძანების ფორმატისაგან დამოკიდებულებით გამოიყენება ოპერაციის შესრულების შედეგის შენახვისათვის, ბრძანების მთვლელის სახით, ოპერანდების და სხვადასხვა მუდმივების (ინდექსების) შენახვისათვის;
- მისამართის ამჯამავი, რომელიც განკუთვნილია ბრძანების სამისამართო ნაწილის მოდიფიკაციისათვის და შემსრულებელი მისამართის ფორმირებისათვის;
- ლოგიკური (ვენტილური) სქემები, რომლებიც უზრუნველყოფენ ბლოკების კვანძებს შორის ინფორმაციის გადაცემის წრედების გადართვას.

ბრძანებების მართვის ბლოკის სქემა, რომელიც ასრულებს ყველა ზემოთ ჩამოთვლილ ფუნქციას წარმოდგენილია ნახ. 3.4-ზე.



ნახ.3.4

ბრძანებების მართვის ბლოკის შემადგენლობა უზრუნველყოფს ორმისამართიანი ბრძანებების შესრულებას, საერთო დანიშნულების რეგისტრებიდან (სდრ) დამისამართებით, რომლებთაგან ერთ-ერთი რეგისტრი გამოიყენება ბრძანებათა მთვლელების სახით (R7).

ვთქვათ მოცემულია ორმისამართიანი ბრძანების კოდი: 06 61 00 (პირველი სიტყვა). ოპერაციის კოდი - 06 აღნიშნავს შეკრების ოპერაციის შესრულების აუცილებლობას. ამ ოპერაციის შესრულებისთვის საჭიროა პირველი მისამართით (SS) მოძებნილი ოპერანდი შეიკრიბოს მეორე ოპერანდთან, რომელიც იმყოფება მეორე

მისამართზე (DD - მიმღები), ხოლო ჯამი მოვათავსოთ მეორე ოპერანდის მისამართზე. სამისამართო ნაწილში კოდი 61 - მიუთითებს იმაზე, რომ პირველ მისამართზე მოცემულია ინდექსური დამისამართება (რეჟიმის მაჩვენებელი 6), ბაზა იმყოფება R1 რეგისტრში, ხოლო წანაცვლება ბრძანების მეორე სიტყვაში. შემსრულებელი მისამართი ფორმირდება როგორც ბაზისა და წანაცვლების ჯამი. კოდი - 00 განსაზღვრავს რეგისტრულ დამისამართებას (რეჟიმის მაჩვენებელი - 0). მეორე ოპერანდისა და ჯამის ადგილსამყოფელს წარმოადგენს R0 რეგისტრი.

ბრძანებების მართვის ბლოკის მუშაობა მოცემული ბრძანებების შესრულების დროს ხორციელდება მიკრობრძანებების თანმიმდევრობითი შესრულების სახით. ბრძანების მიმდინარე მისამართი რომელიც იმყოფება ბრძანებების მთვლელში (ბმთ), მისამართის სისტემური სალტით (მს) იგზავნება ოპერატიულ მეხსიერებაში. მეხსიერების უჯრედიდან მითითებული მისამართით ამოირჩევა შესასრულებელი ბრძანება და მოთავსდება ბრძანების რეგისტრში. ბრძანების მთვლელის მნიშვნელობა იზრდება 2-ით და წარმოადგენს ბრძანების მომდევნო სიტყვის მისამართს. ოპერაციის კოდი და პირველი და მეორე ოპერანდების დამისამართების რეჟიმების კოდი მიეწოდება მიკრობრძანებების ბლოკს მმართველი სიგნალების ფორმირებისათვის, ხოლო რეგისტრის მისამართი - მისამართის დეშიფრატორზე R1 რეგისტრის ამორჩევისათვის, რომელშიც იმყოფება შემსრულებელი მისამართის ბაზა. ბრძანებათა მთვლელის მისამართის მიხედვით ოპერატიული მეხსიერებიდან ამოირჩევა ბრძანების მეორე სიტყვა - წანაცვლება. ბრძანებათა მთვლელის მნიშვნელობა კვლავ ავტომატურად იზრდება 2-ით. ბაზა R1 რეგისტრიდან და წანაცვლება ოპერატიული მეხსიერებიდან ჯამდება მისამართის ამჯამაჟში. ფორმირებული შემსრულებელი მისამართის მიხედვით ამოირჩევა პირველი ოპერანდი და თავსდება პროცესორის არითმეტიკულ-ლოგიკური ბლოკის ერთ-ერთ ბუფერულ რეგისტრში (სქემაზე ნაჩვენები არ არის). მეორე ოპერანდი, რომელიც იმყოფება R0 რეგისტრში, მონაცემების სისტემური სალტის საშუალებით, ასევე მიეწოდება ალმ - ში, რომელიც ამჯამაჟში იკრიბება პირველ ოპერანდთან. მიღებულ ჯამი თავსდება R0 რეგისტრში.

დასკვნის სახით საჭიროა აღინიშნოს, რომ კოდების გადაცემა კვანძებს შორის ხორციელდება ლოგიკურ ელემენტების - ვენტაილების გავლით (სქემაზე ნაჩვენები არაა),

რომელთა მართვა ხორციელდება მიკრობრძანებების მმართველი სიგნალების დახმარებით.

3.4. მიკრობრძანების ბლოკების აგება მმართველი სიგნალების სქემური ფორმირებით

მიკრობრძანების ბლოკი მმართველი სიგნალების სქემური ფორმირებით შესაძლებელია ფუნქციურად დავყოთ ორ ნაწილად. პირველი ნაწილი უზრუნველყოფს ციკლის მოცემას და მის დაყოფას ტაქტებად, ხოლო მეორე ნაწილი უშუალოდ აფორმირებს მმართველ სიგნალებს. ამიტომ პირველი ნაწილი თავის მხრივ წარმოადგენს ტაქტების გენერატორს, რომელიც შედგება იმპულსების გენერატორისგან, მთვლელებისგან და ტაქტების დეშიფრატორებისგან.

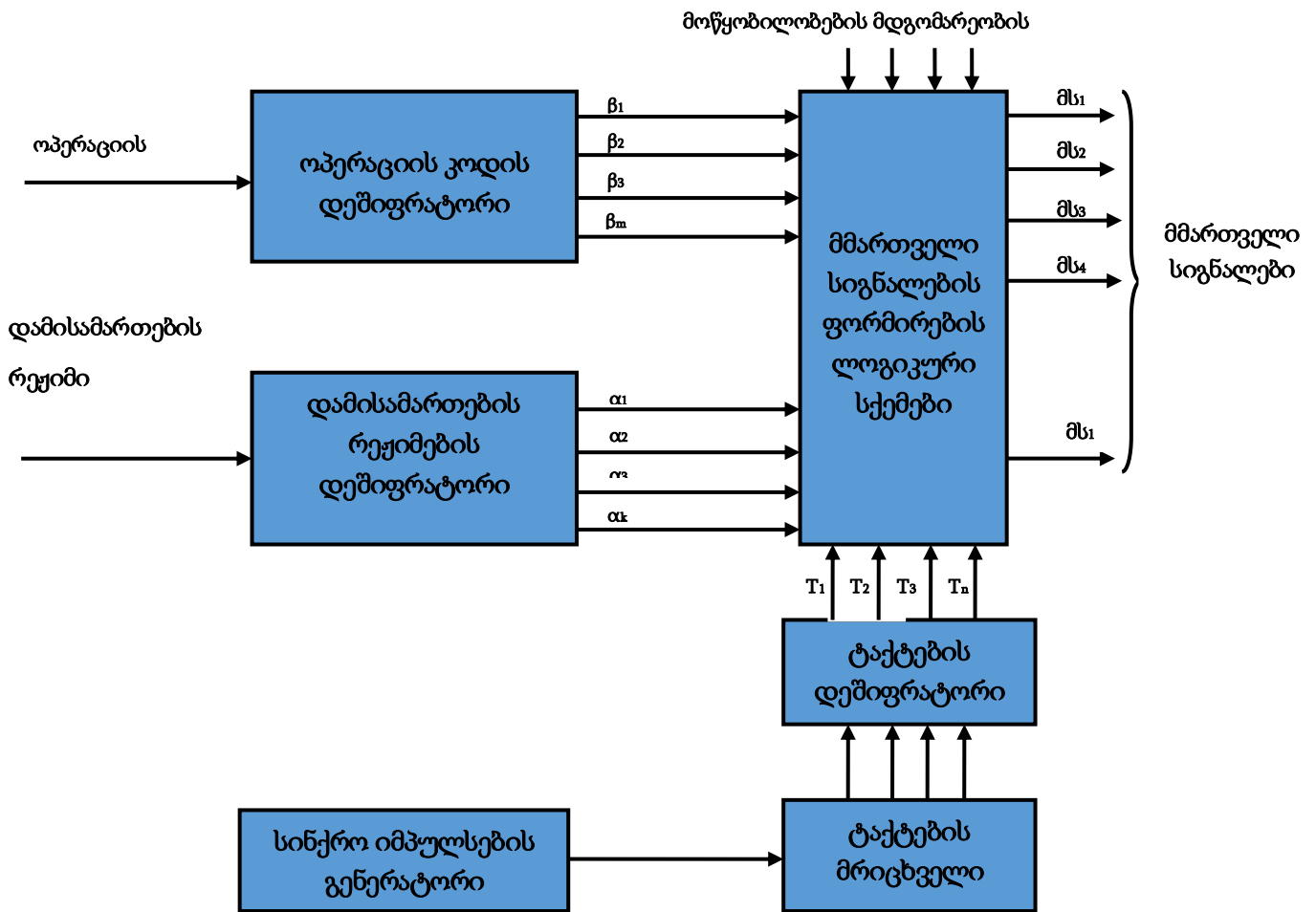
მიკრობრძანების ბლოკის მეორე ნაწილი შედგება ოპერაციის კოდის დეშიფრატორისგან, დამისამართების რეჟიმის დეშიფრატორისგან და მმართველი სიგნალების ფორმირების სქემისგან, რომელიც თავის მხრივ წარმოადგენს, ლოგიკური სქემის ნაკრებს და ფუნქციური სიგნალების მაფორმირებლებს. მიკრობრძანებათა ბლოკის რაოდენობრივი შემადგენლობა დიდადაა დამოკიდებული ოპერაციათა რიცხვისგან, რომელსაც ასრულებს კომპიუტერი, იმდენად რამდენადაც, თითოეული ოპერაციისათვის აუცილებელია მმართველი სიგნალების თავისი კონკრეტული ნაკრები და სიგნალების ფორმირების შესაბამისი სქემების ნაკრები.

მართვისა და სინქრონიზაციის მოწყობილობის მიკრობრძანებათა ბლოკის აგების ერთ-ერთი ვარიანტი მმართველი სიგნალების სქემური ფორმირებით მუდმივი ციკლის დროს მოცემულია ნახ.3.5-ზე.

სინქრო იმპულსების გენერატორიდან ტაქტების მთვლელზე მიეწოდება სინქრო იმპულსები. მთვლელი იცვლის თავის მდგომარეობას თითოეულ ტაქტში. მთვლელის მდგომარეობების შესაბამისი ორობითი კოდები დეშიფრაციის შემდეგ (ტაქტების დეშიფრატორი) გარდაიქმნება $T_1, T_2, \dots, T_n$ სიგნალების თანმიმდევრობაში, რომლებიც დაძრული არიან ერთმანეთის მიმართ და იძლევიან დროით ინტერვალებს - ტაქტებს.

ზემოთ ჩამოთვლილი სიგნალები, ასევე მოწყობილობის მდგომარეობის სიგნალები {მაგალითად არითმეტიკულ - ლოგიკური მოწყობილობის) მიეწოდება ლოგიკურ

სქემებზე მმართველი სიგნალების ფორმირებისთვის. დაწვრილებით განვიხილოთ მმართველი სიგნალების წარმოქმნის პრინციპი, რომლებიც შეადგენენ ბრძანებებს.



ნახ.3.5

ოპერაციათა კოდის და დამისამართების რეჟიმის დეშიფრატორები შესაბამისად აფორმირებენ სიგნალებს $\{\beta\}$ და $\{\alpha\}$ თავიანთ გამოსასვლელებზე განსაზღვრული ბრძანების შესრულების დროს. ზემოთ ჩამოთვლილი სიგნალები, ასევე მოწყობილობის მდგომარეობის სიგნალები (მაგალითად არითმეტიკული-ლოგიკური მოწყობილობის) მიეწოდება ლოგიკურ სქემებზე მმართველი სიგნალების ფორმირებისათვის. ვთქვათ ლოგიკურ განტოლებათა სისტემას რომლებიც აღწერს მიკრობრძანებათა ბლოკის მუშაობას მმართველი სიგნალების გამომუშავებისას გააჩნია შემდეგი სახე:

$$ms_1 = \beta_1 T_1 \vee \beta_1 T_3;$$

$$ms_7 = \beta_1 T_1 \vee \beta_2 T_3;$$

$$ms_2 = \beta_1 T_2 \vee \beta_2 T_3;$$

$$ms_8 = \beta_2 T_1 \vee \beta_1 T_3;$$

$$ms_3 = \beta_1 T_1 \vee \beta_1 T_2 \vee \beta_2 T_3 ;$$

$$ms_9 = \beta_1 T_1 \vee \beta_1 T_2.$$

$$ms_4 = \beta_1 T_1;$$

$$ms_5 = \beta_2 T_2;$$

$$ms_6 = \beta_2 T_2;$$

მაგალითად, განტოლება, ms_3 - თვის უჩვენებს, რომ ეს სიგნალი უნდა დაფორმირდეს პირველი ბრძანებისათვის (β_1) T_2 ტაქტში, ხოლო მეორე ბრძანებისთვის (β_2) - პირველ T_1 და მესამე T_3 ტაქტებში. მოყვანილი განტოლების მიხედვით შესაძლებელია აიგოს ფუნქციური სქემა და დროითი დიაგრამა, რომელიც ნაჩვენებია ნახაზზე 3.6. დროითი დიაგრამიდან ჩანს, რომ პირველი ბრძანების შესრულებისათვის გამოიყენება მიკრობრძანებები:

$$მკბრ_1 : ms_1, ms_4, ms_7, ms_9;$$

$$მკბრ_2 : ms_2, ms_3, ms_9;$$

$$მკბრ_3 : ms_1, ms_8.$$

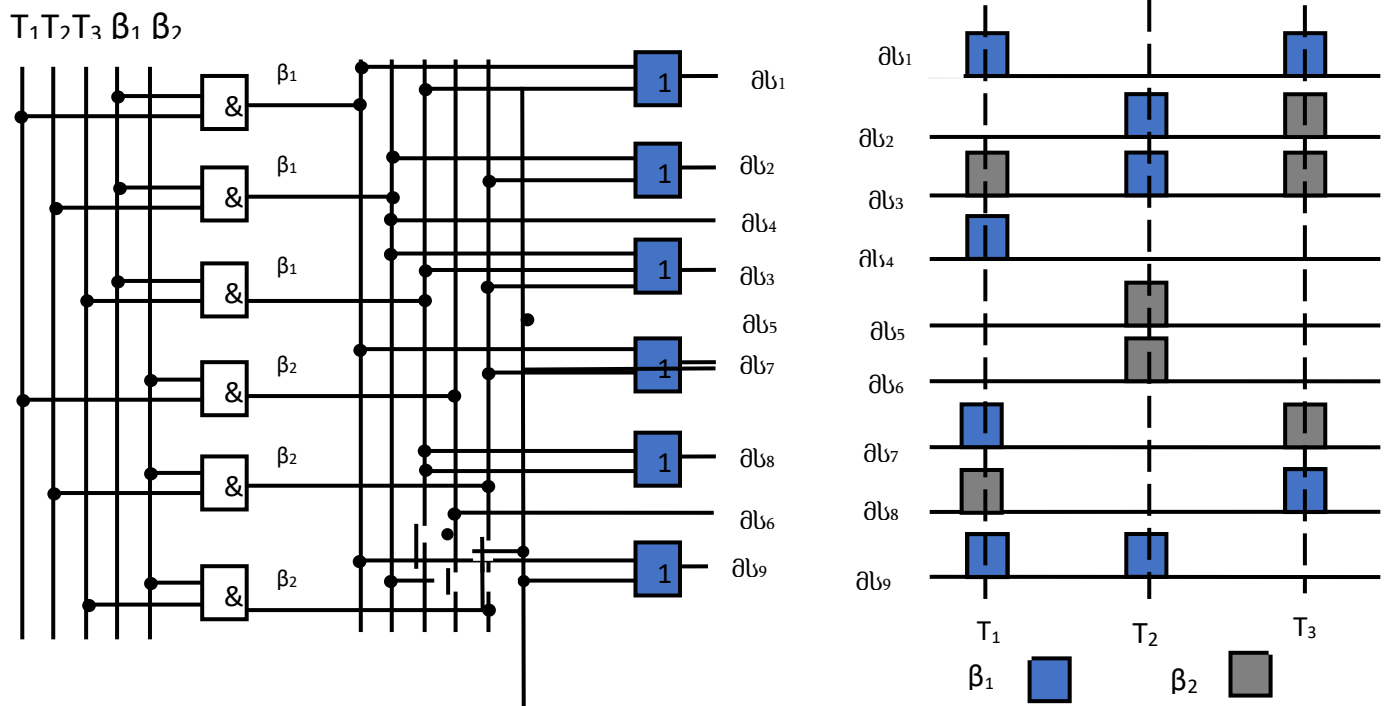
მეორე ბრძანების შესასრულებლად გამოიყენებიან მიკრობრძანებები:

$$მკბრ_1 : ms_3, ms_8;$$

$$მკბრ_2 : ms_5, ms_6;$$

$$მკბრ_3 : ms_2, ms_3, ms_7.$$

მიკრობრძანებათა ბლოკის მოყვანილი სქემის არსებით ნაკლოვანებას წარმოადგენს ის, რომ თითოეულ ციკლში ხდება ტაქტების ერთნაირი რიცხვის ფორმირება, როგორც მოკლე ისე შედარებით უფრო გრძელი ოპერაციებისთვის. ეს იწვევს დროის არამწარმოებლურ დანახარჯებს და ამცირებს პროცესორის სწრაფქმედებას. ციკლში ტაქტების რაოდენობის ცვლილება შესაძლებელია იმპულსების განაწილების დახმარებით, რომლებიც აგებულია ძვრის რეგისტრებზე „მორბენალი 1 ით“. ცვლადი ციკლი ფორმირდება რეგისტრის თანრიგული გამოსასვლელების კომუტაციის გზით.



ნახ.3.6

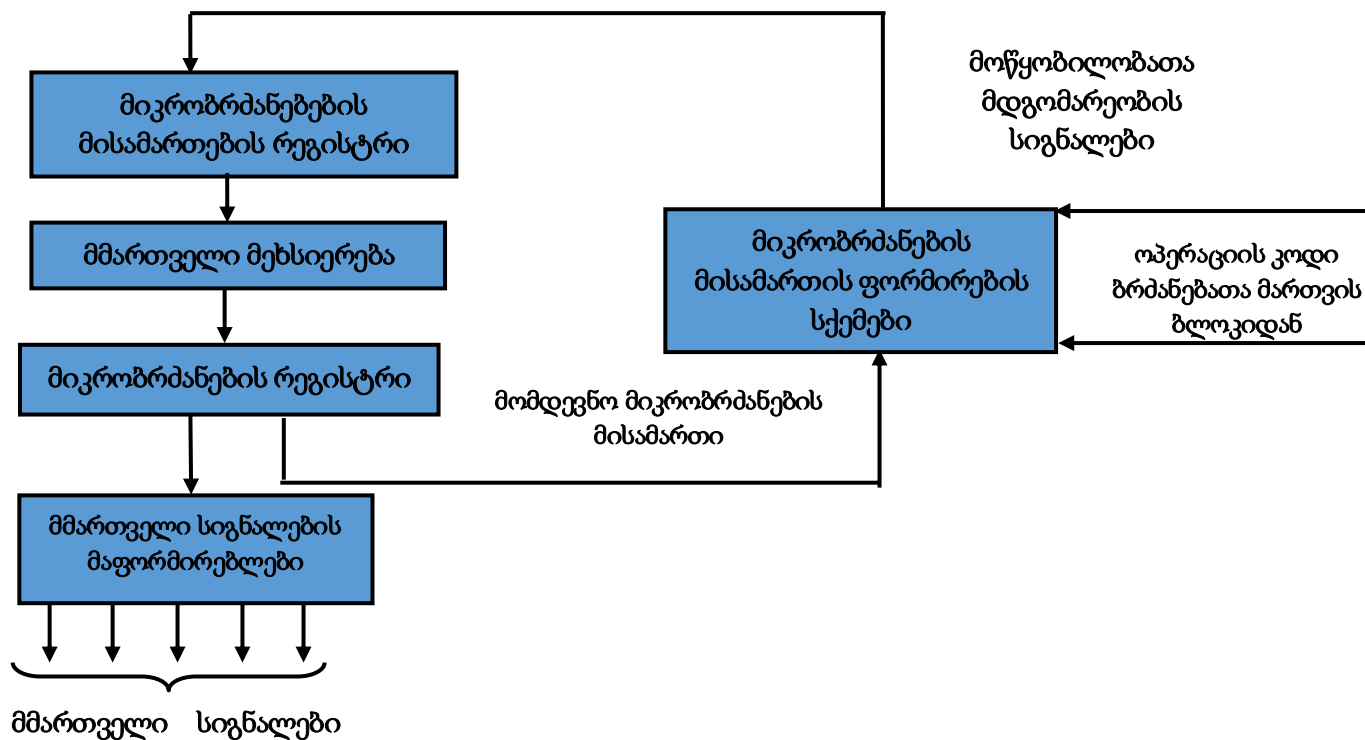
3.5. მიკრობრძანებების ბლოკების აგება მმართველი სიგნალების

პროგრამული ფორმირებით

ამჟამად მმართველი სიგნალების პროგრამული (მიკროპროგრამული) ფორმირება წარმოადგენს ყველაზე უფრო გავრცელებულ მეთოდს და ის გამოიყენება როგორც საბორტო მიკროპროცესორულ სისტემებში, ისე მიწისზედა კომპიუტერებში.

მმართველი სიგნალები ფორმირდებიან მიკრობრძანებების თანმიმდევრობითი ამოკითხვისა და გაშიფვრის საფუძველზე, რომლებიც (მიკრობრძანებები) თავის მხრივ შენახული არიან მიკროპროგრამების სპეციალურ მეხსიერებაში ან მმართველ მეხსიერებაში. მიკრობრძანებების კოდები მმართველი მეხსიერების უჯრედებში შენახულია ისევე, როგორც ბრძანებების კოდები კომპიუტერის ოპერატიულ ან მუდმივ მეხსიერებაში. ამოკითხულ მიკრობრძანებათა თანმიმდევრობა თავის მხრივ წარმოადგენს კომპიუტერის ოპერაციის მართვის მიკროპროგრამას. მიკროპროგრამები შესაძლებელია განლაგდნენ მმართველი მეხსიერების განსაზღვრული არის თანმიმდევრობით უჯრედებში ან დაფორმირდნენ მიკრობრძანებებისგან, რომლებიც განლაგებული იქნებიან მეხსიერების ნებისმიერ ნაწილში. უკანასკნელ შემთხვევაში მიიღწევა

მეხსიერების ეკონომია ერთი და იმავე მიკრობრძანებების გამოყენების ხარჯზე სხვადასხვა მიკროპროგრამებში. მიკროპროგრამული ტიპის მიკრობრძანებათა ბლოკის სტრუქტურული სქემის ნაწილი მუდმივი მეხსიერების მართვის შემთხვევაში მოცემულია ნახ.3.7-ზე. ნახაზზე ნაჩვენებია არაა ტაქტების გენერატორი და მიკრობრძანებათა ამოკითხვის მართვის სქემები.



ნახ.3.7

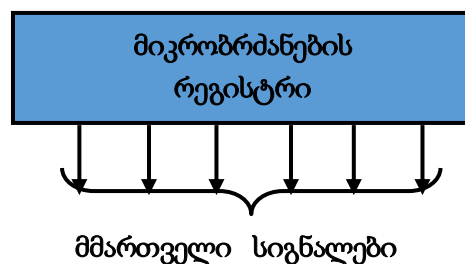
მიკრობრძანებების კოდები, რომლებიც შენახულია მეხსიერებაში ზოგადად შედგება ოპერაციული და სამისამართო ნაწილებისგან. კოდის ოპერაციული ნაწილი გამოიყენება თითოეულ ტაქტში მმართველი სიგნალების ფორმირებისათვის. სამისამართო ნაწილის დანიშნულებაა მომდევნო მიკრობრძანების მისამართის მიღება, იმ შემთხვევაში როცა მიკრობრძანებები მეხსიერებაში განლაგებულია ნებისმიერად და მოითხოვება მისამართის იძულებითი ფორმირება. მისამართის ბუნებრივი ფორმირების შემთხვევაში მომდევნო მიკრობრძანების მისამართი ფორმირდება მიკრობრძანებათა მთვლელის დახმარებით მისი შიგთავსის ავტომატურად გაზრდით 1-ით ან 2-ით ან სხვა რიცხვით იმისგან დამოკიდებულებით, თუ ერთდროულად რამდენი უჯრედისაგან ამოკითხება ინფორმაცია.

იმისათვის, რომ განხორციელდეს გადასვლები მიკრობრძანებებთან, რომლებიც არ ფორმირდებიან აღწერილი წესის მიხედვით, გამოიყენება სპეციალურ მისამართიანი

მიკრობრძანებები, რომლებიც განლაგებულია მიკროპროგრამის განსაზღვრულ ადგილში ოპერაციულ მიკრობრძანებათა შორის. ისინი განსხვავდებიან სპეციალური ნიშან-თვისებებით (ერთი დამატებითი თანრიგით). მომდევნო მიკრობრძანების მისამართის ფორმირების ასეთი ხერხის გამოყენება იწვევს მიკროპროგრამის სიგრძის გაზრდას, თუმცა ამავედროულად იწვევს თვითონ მიკრობრძანების სიგრძის შემოკლებას, რაც საბოლოო ჯამში განაპირობებს არსებით მოგებას აპარატურული დანახარჯების თვალსაზრისით.

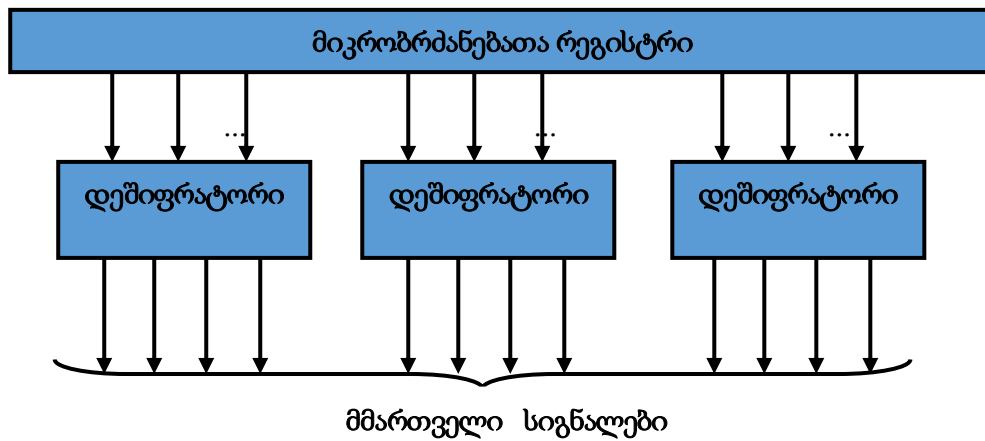
მმართველის სიგნალების ფორმირებისათვის გამოიყენება მიკრობრძანების ოპერაციული ნაწილი. მმართველი წრედების დიდი რაოდენობის შემთხვევაში, შესაბამისად მმართველი სიგნალების დიდი რაოდენობის დროსაც მოითხოვება მიკრობრძანების ოპერაციული ნაწილის დიდი თანრიგობა და შესაბამისად მმართველი მეხსიერების უჯრედების შესაბამისი თანრიგობა. ამიტომ სხვადასხვა კომპიუტერებში მმართველი წრედების რიცხვისგან დამოკიდებულებით გამოიყენება მმართველი სიგნალების ფორმირების სხვადასხვა ხერხები.

მმართველი სიგნალების მცირე რაოდენობის შემთხვევაში (100-150) მიკრობრძანების ოპერაციული ნაწილის თითოეულ თანრიგს შეესატყვისება თავისი მმართველი სიგნალი, რაც საშუალებას იძლევა თითოეულ ტაქტში დაფორმირდეს სიგნალების ნებისმიერი რიცხვი (ნახ.3.8).



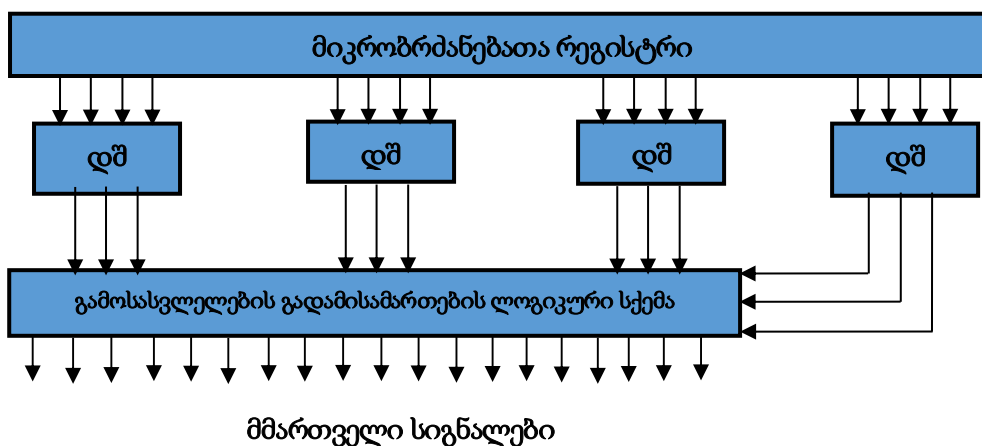
ნახ.3.8

თუ ოპერაციულ ნაწილს განვიხილავთ, როგორც ველების სიმრავლეს რომლებიც შედგება რამდენიმე თანრიგისაგან და მოვახდენთ მიკრობრძანებათა ველების დეშიფრირებას, მაშინ შესაძლებელია არსებითად გაიზარდოს მმართველი სიგნალების რაოდენობა მიკრობრძანებათა თანრიგობის უცვლელობის დროს (ნახ.3.9).



ნახ.3.9

განხილულ მაგალითებში მიკრობრძანების ყველა ველი ასრულებს მკაცრად განსაზღვრულ ფუნქციას და გამოიყენება მმართველი სიგნალების მოცემული ერთობლიობის ფორმირებისათვის. ამ ერთობლიობის გაფართოების მიღწევა შესაძლებელია გამოსასვლელი წრედების გადამისამართების გზით და ამით შესაბამისად მიიღწევა გამოსასვლელი სიგნალების რაოდენობის გაზრდა. ამისათვის შემოიტანება გადამისამართების დამატებითი ველი და ამ ველის დეშიფრაცია. მმართველი სიგნალების რაოდენობის გაფართოება მიიღწევა გადამისამართების ლოგიკურ სქემაში, რომელიც ახორციელებს მიკრობრძანებათა რეგისტრის გამოსასვლელებისა და ველების დეშიფრატორების გამოსასვლელების გადართვას (ნახ.3.10).



ნახ.3.10

მიკრობრძანება სრულდება ერთი ტაქტის განმავლობაში. მიკროოპერაციები კი შეიძლება შესრულდნენ ან ერთდროულად, ან ტაქტის სხვადასხვა ფაზებში (მიკროტაქტები), მაშინ თუ წამოიჭრება აუცილებლობა იმისა რომ მოხდეს მმართველი სიგნალების განაწილება კომპიუტერის კვანძების მართვის მდგრადობის ამაღლებისათვის.

თანამედროვე საბორტო კომპიუტერებში და საბორტო მიკროპროცესორულ სისტემებში მართვის მიკროპროგრამული ხერხი წარმოადგენს ყველაზე უფრო გავრცელებულ და უპირატეს ხერხს. მიკროპროგრამირების სიმარტივე და მოქნილობა და მუდმივი დამხმარებელი მოწყობილობების არსებობა ინტეგრალური შესრულებით და მაღალი სწრაფქმედებით განაპირობებს იმას, რომ ამაღლდეს შესასრულებელი ოპერაციების მართვის ეფექტურობა.

ლექცია 4

საბორტო კომპიუტერების არითმეტიკულ-ლოგიკური მოწყობილობები

4.1. ზოგადი ცნობები არითმეტიკულ-ლოგიკური მოწყობილობების შესახებ

პროცესორი თავის თავში მოიცავს ოპერაციულ ნაწილს და მართვის ნაწილს. ოპერაციული ნაწილი უზრუნველყოფს ოპერანდებზე მოქმედებას ე.ი. ასრულებს ოპერაციებს, რომელიც ოპერაციის კოდითაა მოცემული. ოპერაციულ ნაწილს ჩვეულებრივ უწოდებენ არითმეტიკულ-ლოგიკურ მოწყობილობას (ალმ).

არითმეტიკულ-ლოგიკური მოწყობილობის ზოგადი სტრუქტურული სქემა წარმოდგენილია ნახ.4.1-ზე.

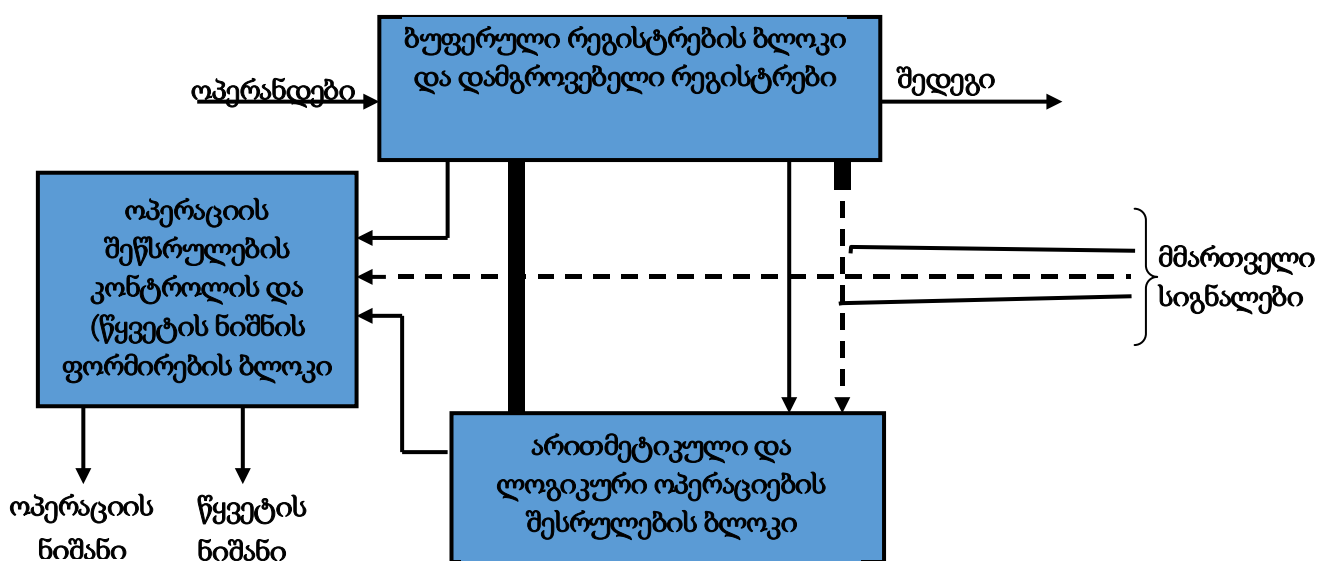
ალმ საბორტო კომპიუტერში განკუთვნილია ოპერანდების კოდებზე არითმეტიკული და ლოგიკური ოპერაციების შესრულებისათვის. ასეთ ოპერაციებს მიეკუთვნება:

1. ფიქსირებული მძიმით წარმოდგენილი რიცხვებზე ორობითი არითმეტიკის ოპერაციები (შეკრება, გამოკლება, გამრავლება, გაყოფა);
2. მცოცავი მძიმით წარმოდგენილ რიცხვებზე არითმეტიკული ოპერაციები;
3. ორობით-ათობითი კოდში წარმოდგენილ რიცხვებზე არითმეტიკული ოპერაციები;
4. რიცხვების კოდებზე ლოგიკური ოპერაციები (ინვერსია, ლოგიკური შეკრება, ლოგიკური გამრავლება, შეკრება ორის მოდულით, რიცხვითი კოდების შედარება და სხვ.);
5. სპეციალური ოპერაციები (კოდების დაძვრა მარცხნივ ან მარჯვნივ, რიცხვების კოდის 1-ით გაზრდა ან შემცირება, მანტისის ნორმალიზაცია და სხვა).

საბორტო გამოთვლით სისტემებში ჩამოთვლილი ოპერაციები შეიძლება სრულდებოდეს სრული მოცულობით ან შემცირებული ვარიანტით. მაგალითად შესაძლებელია არ გვექონდეს ალმ-ში ოპერაციები მცოცავი მძიმით, მიკროპროცესორულ საბორტო სისტემებში ალმ-ის გამრავლებისა და გაყოფის ოპერაციების ბრძანებები. აღნიშნული ოპერაციების შესრულება ხორციელდება სპეციალური ქვეპროგრამებით.

იმის მიხედვით პარალელურ თუ მიმდევრობით კოდში ხდება ინფორმაციის დამუშავება, ალმ-ის აგება ხდება პარალელური ან მიმდევრობითი მოქმედებით. პარალელური მოქმედების ალმ უზრუნველყოფს შედარებით მაღალ სიჩქარეს ოპერაციის შესრულების დროს და ამიტომ პოულობენ უპირატესობას გამოყენების თვალსაზრისით.

ვინაიდან ინფორმაციის წარმოდგენის ხერხი გავლენას ახდენს ალმ-ის სტრუქტურაზე, ამიტომ განასხვავებენ ალმ-ს ფიქსირებული მდომით, მცოცავი მდომით რიცხვებისათვის და ორობით-ათობით კოდში წარმოდგენილი რიცხვებისათვის.



ნახ. 4.1

ალმ-ები შეიძლება იყოს რეალიზებული ბლოკურ პრინციპზე ე.ი. როცა ერთ ალმ-ში თავმოყრილია სხვადასხვა ბლოკები ფიქსირებული მდომით, მცოცავი მდომით და ორობით-ათობით კოდში წარმოდგენილი რიცხვებისათვის. ამის გარდა ალმ-ის შემადგენლობაში შეიძლება იყოს დამოუკიდებელი ბლოკები გამრავლების, გაყოფის, ან ლოგიკური ოპერაციებისათვის. ასეთი ხერხი იძლევა საშუალებას, ერთის მხრივ უზრუნველყოფილ იქნეს ბლოკების პარალელური მუშაობა, მეორეს მხრივ, თითოეულ ბლოკში უნიფიცირებული სქემის ხარჯზე მღწეულ იქნეს უფრო მაღალი სწრაფქმედება. ორივე შემთხვევაში საგრძნობლად მაღლდება ინფორმაციის დამუშავების სისწრაფე.

ბლოკური ტიპის ალმ-ის გარდა გამოიყენება მრავალფუნქციური მოქმედების ალმ-ები. ასეთ მოწყობილობებში ინფორმაციის წარმოდგენის ყველა ფორმისათვის და

შესასრულებელი ოპერაციებისთვის გამოიყენება ერთი და იგივე კვანძები, რომლებიც ლოგიკური სქემების კომპუტაციის გზით მიკრობრძანებების აუცილებელი შერწყმის დახმარებით ახდენენ მუშაობის მოთხოვნილი რეჟიმის განხორციელებას.

შესასრულებელი ოპერაციების ნაკრების და ინფორმაციის წარმოდგენის ფორმის მიხედვით, ასევე შესაძლო სხვა მიზეზებით ბლოკების შემადგენლობა და აგების ხერხები შეიძლება იყოს სხვადასხვა.

4.2. არითმეტიკულ-ლოგიკური მოწყობილობა ფიქსირებული მძიმით წარმოდგენილი რიცხვების შეკრებისა და გამოკლებისათვის

შეკრების ან გამოკლების ოპერაციის შესრულების ალგორითმის თანახმად უარყოფითი რიცხვები წარმოდგენილ უნდა იქნეს შებრუნებულ ან დამატებით კოდში, ხოლო დადებითი რიცხვები პირდაპირ კოდში. შეკრება ხორციელდება თითოეულ თანრიგზე დაწყებული უმცირესი თანრიგიდან ნიშნის თანრიგის ჩათვლით. შებრუნებულ კოდში შეკრების დროს გადატანის ერთიანი ნიშნის თანრიგიდან მიემატება მიღებულ შედეგს უმცროს თანრიგში საბოლოო ჯამის წარმოქმნისათვის.

დამატებით კოდში რიცხვების შეკრების დროს ეს ერთიანი იკარგება. აქედან გამომდინარე შეკრების ოპერაციის დაჩქარების მიზნით მიზანშეწონილია გამოვიყენოთ დამატებითი კოდი, რომლებიც ინახება ოპერატიულ მეხსიერებაში. რიცხვების შეკრების ან გამოკლების დროს ავტომატურად განისაზღვრება მიღებული შედეგის ნიშანთვისება: შედეგი ნულის ტოლია, შედეგი უარყოფითია, შედეგი დადებითია, თანრიგთა ბადის გადავსება (ცხრილი. 4.1).

ცხრილი 4.1

N	გადატანები		ოპერაციის შედეგი	შედეგის ნიშან- თვისების კოდი
	ნიშნის თანრიგიდან	უფროსი თანრიგიდან		
1	0	0	$\Sigma > 0$	10
2	0	1	გადავსება	11
3	1	0		
4	1	1	$\Sigma < 0$	01
5	--	--	$\Sigma = 0$	00

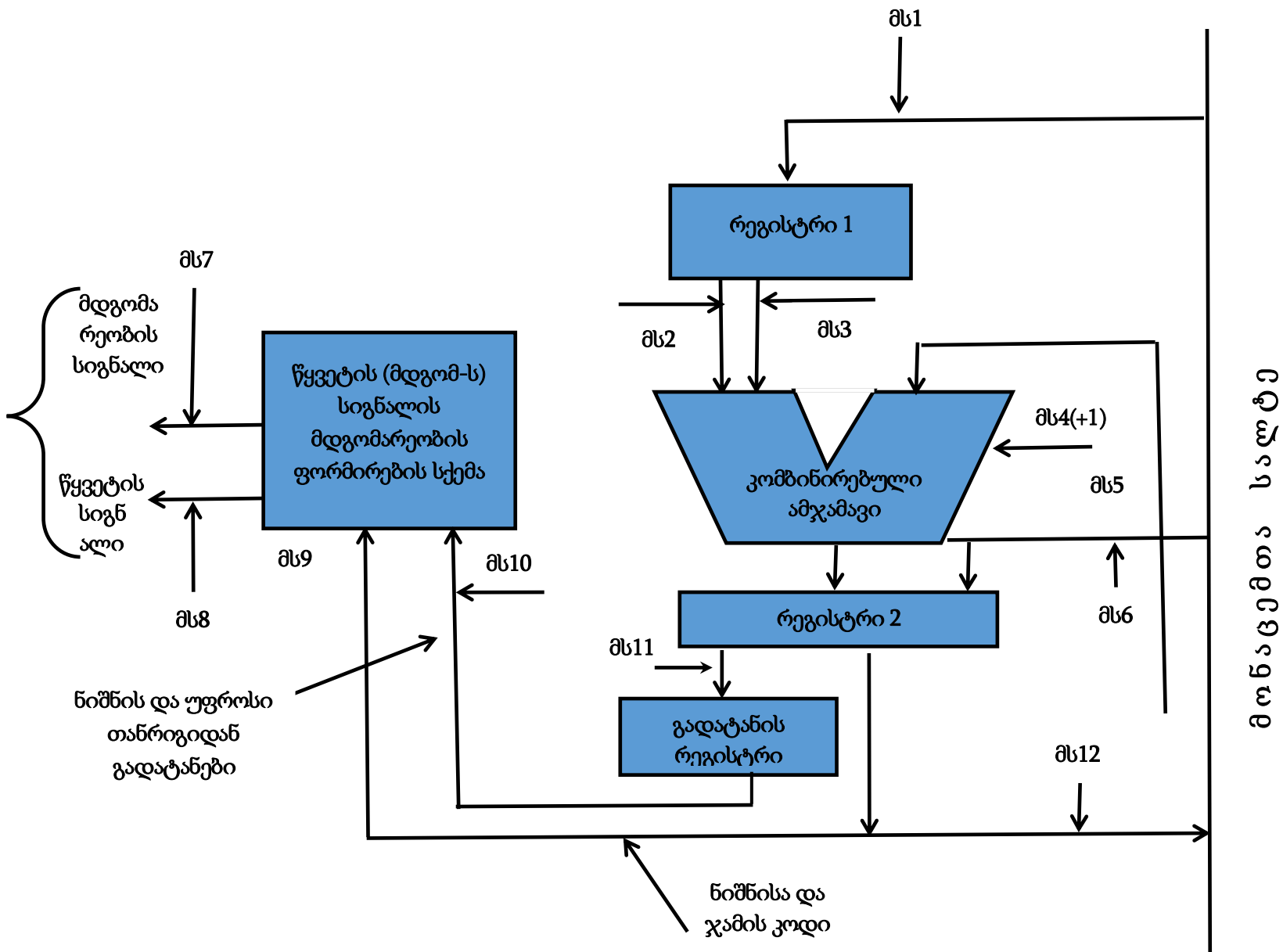
როგორც ცხრილიდან ჩანს შეკრების შედეგის ინდიკატორად გამოიყენება ნიშნის და უფროსი თანრიგიდან წარმოქმნილი გადატანის თანრიგები, რომლებიც დამახსოვრდებიან და მონაწილეობას იღებენ შედეგის ნიშანთვისების ფორმირებაში. არითმეტიკულ-ლოგიკური მოწყობილობის სტრუქტურული სქემა ფიქსირებული მძიმით წარმოდგენილი n თანრიგა რიცხვების შეკრებისა და გამოკლებისათვის, ასევე უარყოფითი რიცხვების დამატებით კოდში წარმოდგენის შემთხვევაში ნაჩვენებია ნახ.4.2-ზე.

სქემაზე ნაჩვენები მმართველი სიგნალები მს1-მს12, უზრუნველყოფენ კოდების გადაცემის მიკროოპერაციების შესრულებას სქემის ერთი კვანძიდან მეორეზე. გარდა ამისა იგულისხმება, რომ ახალი ინფორმაციის ჩაწერა რეგისტრებში ხორციელდება ერთდროულად წინა ინფორმაციის წაშლასთან ერთად.

კოდების გამოკლების ოპერაცია შეიძლება დაყვანილ იქნეს შეკრებაზე მაკლების ნიშნის შეცვლით, რომელიც გამოდის ტოლობიდან

$$R = A - B = A + (-B).$$

თუ მაკლები წარმოადგენს დადებით რიცხვს და მოცემულია პირდაპირ კოდში, მაშინ ის გარდაიქმნება რიცხვად დამატებით კოდში. კოდის თანრიგები ინვერტირდება (შებრუნებული კოდი), რაც ტოლფასია რეგისტრის ინვერსიული გამოსავლელებიდან



კოდის წაკითხვისა და „1“ მიემატა ჯამის უმცირეს თანრიგს. თუ მაკლები უარყოფითი რიცხვია და წარმოდგენილია დამატებით კოდში, მაშინ მათი გარდაქმნა პირდაპირ კოდში ხორციელდება ანალოგიური გზით.

მმართველი სიგნალების (მიკრობრძანებების) გამომუშავების თანმიმდევრობა ოპერაციებისათვის „შეკრება“ და „გამოკლება“ მოცემულია ცხრილში 4.2.

ცხრილი 4.2.

	N	შესასრულებელი მოქმედება	მართვის სიგნალები												მიკრობრძანება	
	1	პირველი ოპერანდის მიღება	1	2	3	4	5	6	7	8	9	10	11	12	მ ი კ რ ო ბ რ ა მ ა 1	მ ი კ რ ო ბ რ ა მ ა 2
შეკრება	2	მეორე ოპერანდის მიღება	1					1								
	3	შეკრება		1			1						1			
	4	შედეგის ანალიზი							1		1	1				
	5	თუ შნთ = 11, მაშინ წყვეტა								1						
	6	თუ შნთ ≠ 11, მაშინ კოდის გაცემა											1			
გამოკლება	1	I ოპერანდის მიღება						1							მ ი კ რ ო ბ რ ა მ ა 2	მ ი კ რ ო ბ რ ა მ ა 2
	2	II ოპერანდის მიღება	1													
	3	შეკრება			1	1	1						1			
	4	შედეგის ანალიზი:							1		1	1				
	5	თუ შნთ = 11, მაშინ წყვეტა.								1						
	6	თუ შნთ ≠ 11, მაშინ კოდის გაცემა.											1			

განვიხილოთ ოპერაციის „გამოკლება“ შესრულების ალგორითმი. ოპერაცია იწყება პირველი ოპერანდის მიღებით რეგისტრ 2-ში (მმართველი სიგნალით მს6) მონაცემთა სალტიდან (მონ.ს). რეგისტრი 2 ერთდროულად წარმოადგენს ამჯამავის რეგისტრს და რეგისტრ-აკუმულატორს. შემდეგ ტაქტში (მს1) რეგისტრ1-ში შეიტანება მეორე ოპერანდი-მაკლები. მს-3 სიგნალით რეგისტრ 1-ის ინვერსიული გამოსასვლელიდან წაიკითხება მაკლების ინვერტირებული კოდი, რომელიც მიეწოდება კომბინაციური ამჯამავის პირველ შესასვლელს. ერთდროულად მიკრობრძანება 3-ში (მბ3) ფორმირდება მს4, მს5 და მს11 სიგნალები რეგისტრ 2-დან პირველი ოპერანდის მისაწოდებლად ამჯამავის მეორე შესასვლელზე (მს-5), გადატანის დამახსოვრებისათვის გადატანის რეგისტრში (მს11) და შედეგის კორექციის რეგისტრში (მს4=+1). შედეგი (სხვაობა) ინახება რეგისტრში 2. შედეგის ანალიზისათვის მდგომარეობის და წყვეტის სიგნალების მაფორმირებელ ლოგიკურ სქემაზე ნიშნის და უფროსი თანრიგიდან მიეწოდება გადატანები (მს-9), სხვაობის ნიშნის და ციფრული თანრიგები. მდგომარეობის სიგნალების ფორმირება ხორციელდება ცხრილ 4.1-ის შესაბამისად. თანრიგთა ბადის გადავსების დროს (მიღებული სხვაობა ერთზე მეტია) გამომუშავდება წყვეტის სიგნალი (მს-7) მიმდინარე პროგრამის შესრულების შეწყვეტისათვის. მდგომარეობის სიგნალები შეინახება სპეციალურ რეგისტრში (მდგომარეობის რეგისტრში) და გამოიყენება მართვის მოწყობილობაში მართვის სიგნალების მიმდევრობის შეცვლისათვის და პირობითი გადასვლის მოცემისათვის შესასრულებელ პროგრამაში (პირობითი გადასვლის ბრძანებებით).

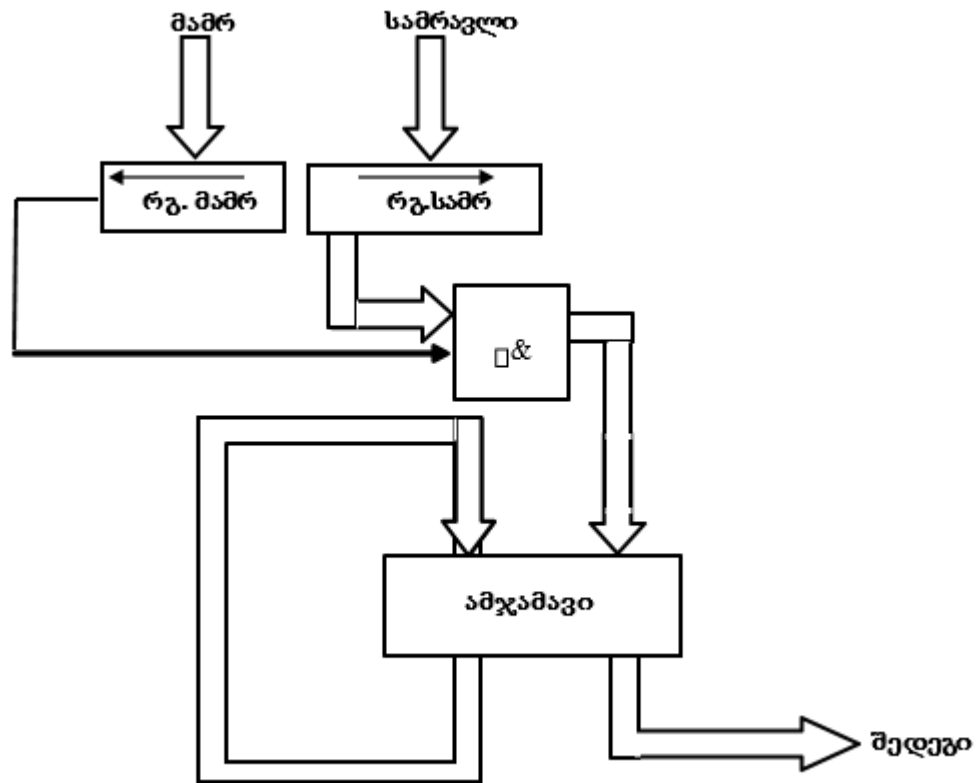
4.3. არითმეტიკულ-ლოგიკურ მოწყობილობა ფიქსირებული მძიმით წარმოდგენილი რიცხვების გამრავლებისათვის

არითმეტიკულ-ლოგიკური მოწყობილობის სტრუქტურული სქემა, რომელიც ასრულებს ფიქსირებული მძიმით წარმოდგენილ რიცხვებზე გამრავლების ოპერაციას ძირითადად განისაზღვრება ოპერაციის ალგორითმით. საბოლოო კომპიუტერში შეიძლება გამოყენებულ იქნეს გამრავლების ალგორითმები, რომელთაგანაც გამრავლება იწყება მამრავლის უმცროსი ან უფროსი თანრიგებიდან, დადებითი რიცხვების

თანამამრავლების აბსოლუტური მნიშვნელობებისათვის და დამატებით კოდში წარმოდგენილი რიცხვებისათვის. განვიხილოთ ეს მეთოდები:

პირველი მეთოდი

ამ მეთოდის არსი ნაჩვენებია ნახ. 4.3-ზე.



ნახ.4.3

გამრავლება იწყება მამრავლის უფროსი თანრიგების ანალიზით და მარცხნივ დაძვრის გზით უფროსი თანრიგების მხარეს. გამრავლების ყოველ ბიჯზე სამრავლი დაიძვრება მარჯვნივ. თუ მამრავლის მორიგი თანრიგია “1”, მაშინ დაძვრული სამრავლი გადაიცემა ამჯამავში და შეიკრიბება მის წინა შიგთავსთან. თუ მორიგი თანრიგია არის “0”, მაშინ დაძვრული სამრავლი არ მიეწოდება ამჯამავს. შედეგად ფორმირდება რიცხვების ნამრავლი. ვთქვათ X-სამრავლია, Y-მამრავლი, მაშინ $X=0, X_1, X_2, \dots, X_n$, $Y=0, Y_1, Y_2, \dots, Y_n$. $Y=0, Y_1, Y_2, \dots, Y_n = Y_1 2^{-1} + Y_2 2^{-2} + \dots + Y_n 2^{-n}$. აქედან $Z = X \cdot Y = X \cdot (Y_1 2^{-1} + Y_2 2^{-2} + \dots + Y_n 2^{-n}) = (X \cdot 2^{-1}) Y_1 + (X \cdot 2^{-2}) Y_2 + \dots + (X \cdot 2^{-n}) Y_n$.

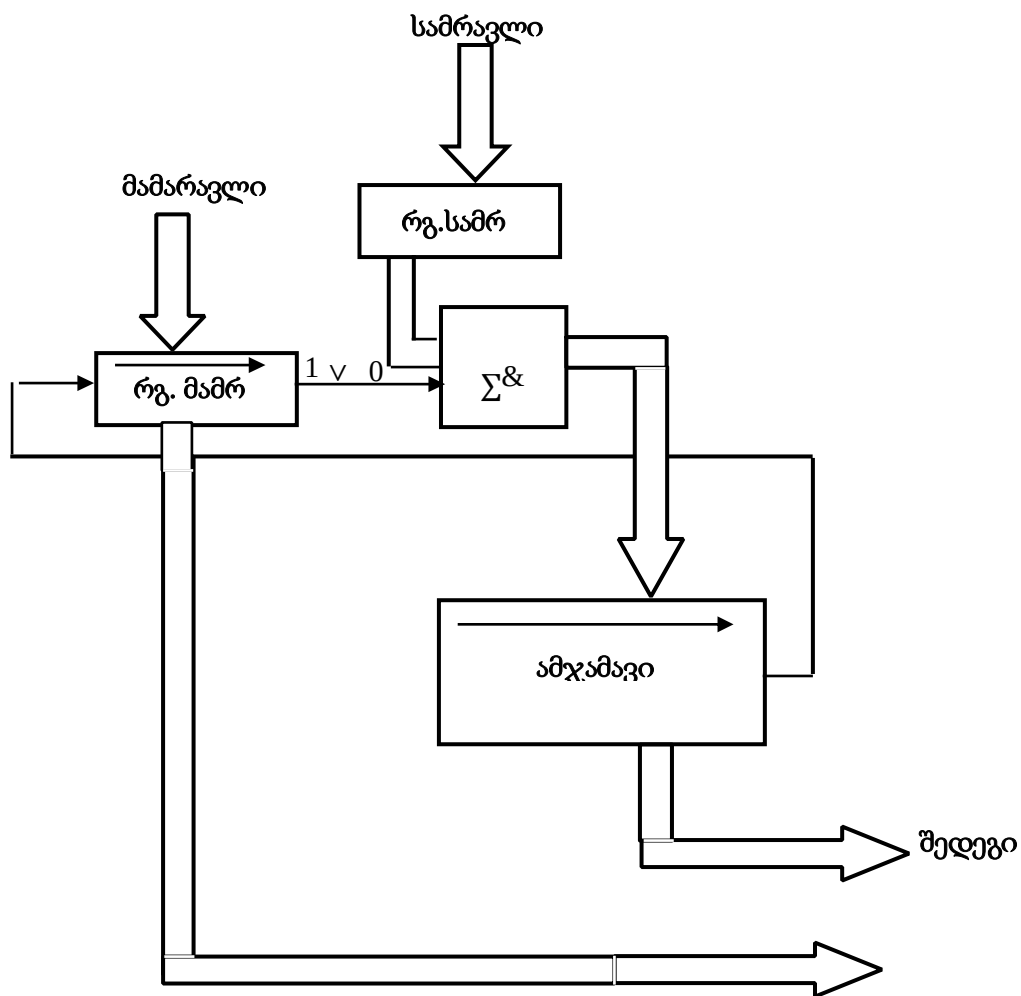
გამოსახულება $(X \cdot 2^{-j}) Y_j$ აღნიშნავს X სამრავლის დაძვრას j-თანრიგებით მარჯვნივ და მის გამრავლებას მამრავლის j-ურ თანრიგზე. ვინაიდან სამრავლის დაძვრის დროს

შეიძლება დაიკარგოს დაძრული თანრიგები, ამიტომ აუცილებელია გათვალისწინებულ იქნეს სამრავლის რეგისტრში დამატებითი Π -თანრიგები. ქვემოთ მოყვანილია გამრავლების მაგალითი, როცა მამრ = 0,1101 და სამრ=0,1101.

$$\begin{array}{rcl}
 + \begin{array}{r} 0,00000000 \\ 0,01101000 \\ \hline \end{array} & \xrightarrow[\text{მმ}]{\Sigma_0} & 1 \\
 + \begin{array}{r} 0,01101000 \\ 0,00110100 \\ \hline \end{array} & \xrightarrow[\text{მმ}]{\Sigma_1} & 2 \\
 \begin{array}{r} 0,10011100 \\ (0,00011010) \\ \hline \end{array} & \xrightarrow[\text{მმ}]{\Sigma_2} & 3 \\
 + \begin{array}{r} 000000000 \\ 0,10011100 \\ \hline \end{array} & \xrightarrow[\text{მმ}]{\Sigma_3} & 4 \\
 + \begin{array}{r} 0,00001101 \\ 0,10101001 \\ \hline \end{array} & \xrightarrow{\Sigma_4} & \text{ნამრავლი}
 \end{array}$$

მეორე მეთოდი

გამრავლება იწყება მამრავლის უმცროსი თანრიგების ანალიზით მათი მარჯვნივ დაძვრით უმცროსი თანრიგების მხარეს. გამრავლების ყოველ ბიჯზე კერძო ნამრავლი დაიძვრება მარჯვნივ, რომელიც მიიღება წინა ჯამისა და კერძო ნამრავლის შეკრებით (სამრავლის მამრავლის მორიგ თანრიგზე). ნამრავლი ფორმირდება კერძო ჯამების ძვრით Π -ჯერ. ნამრავლის უმცროსი თანრიგები შეიძლება შენახულ იქნეს მამრავლის რეგისტრში, რომლის თანრიგები თავისუფლდებიან მარჯვნივ ძვრის შემდეგ. მე-2 მეთოდის მიხედვით გამრავლების შესრულების სქემა ნაჩვენებია ნახ.4.4-ზე.



ნახ. 4.4.

ვთქვათ X -არის სამრავლი, Y -მამრავლი. მაშინ

$$X=0, X_1, X_2 \dots X_n, \quad Y=0, Y_1, Y_2 \dots Y_n,$$

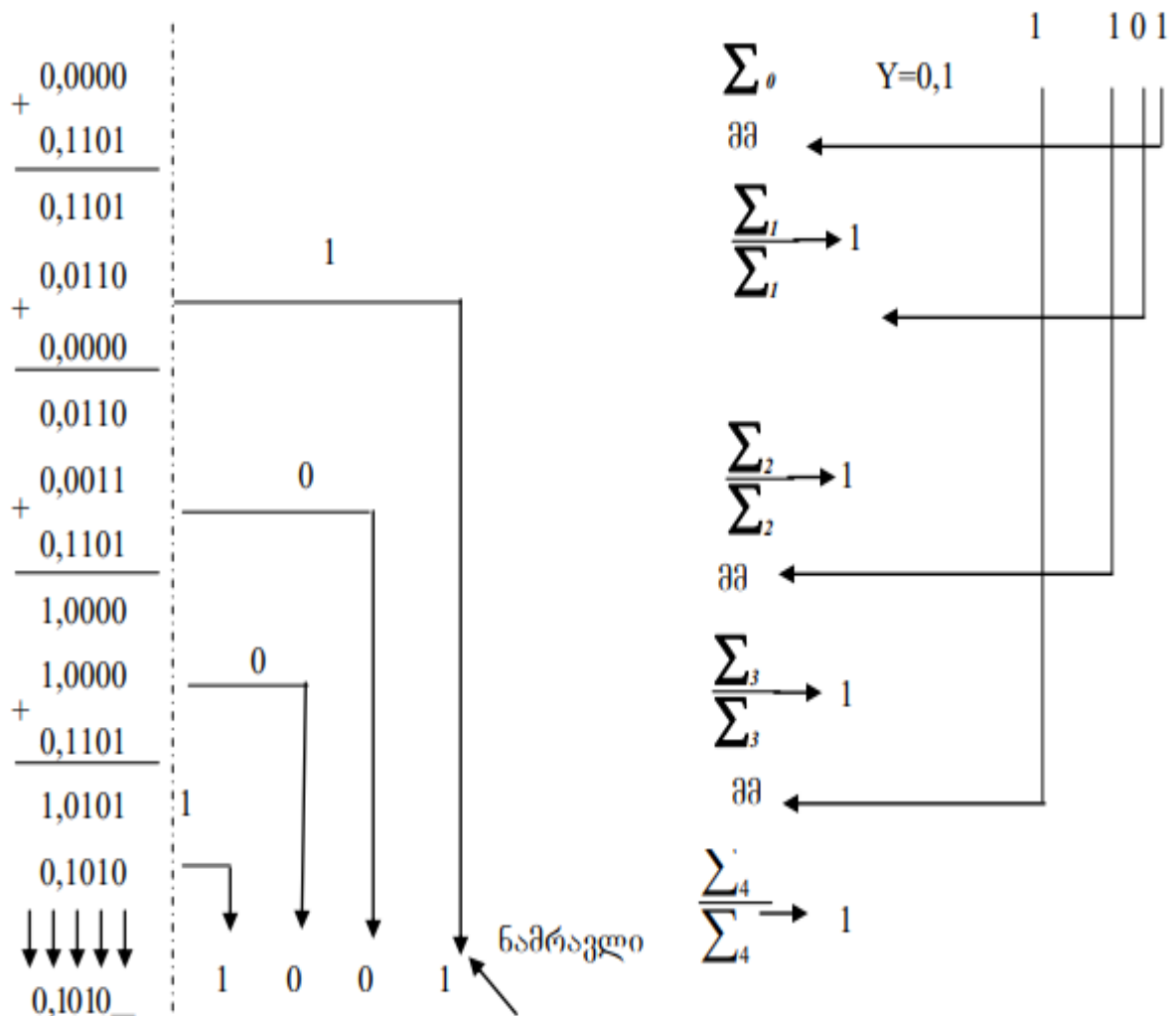
$$Y = 0, Y_1, Y_2 \dots Y_n = Y_1 2^{-1} + Y_2 2^{-2} + Y_3 2^{-3} + Y_{n-2} 2^{-(n-2)} + Y_{n-1} 2^{-(n-1)} + Y_n 2^{-n} =$$

$$(\dots((Y_n 2^{-1} + Y_{n-1}) 2^{-1} + Y_{n-2}) 2^{-1} + \dots + Y_2) 2^{-1} + Y_1) 2^{-1}.$$

აქედან

$$Z = X \cdot Y = (\dots((Y_n X) 2^{-1} + Y_{n-1} X) 2^{-1} + Y_1 X) 2^{-1}.$$

მეორე მეთოდით შესრულებული გამრავლების ოპერაციის მაგალითის ილუსტრაცია ასეთია



4.4 არითმეტიკულ-ლოგიკური მოწყობილობა მცოცავი მძიმით წარმოდგენილი რიცხვების შეკრების და გამოკლების ოპერაციების შესასრულებლად

მცოცავი მძიმით რიცხვების წარმოდგენა ხდება მანტისისა და რიგის სახით:

$X = \pm P_X; \pm m_x$, სადაც $X = 2^{\pm P_X} (\pm m_x)$. ამიტომ მცოცავი მძიმით წარმოდგენილ რიცხვებზე

არითმეტიკული ოპერაციების შესრულების დროს აუცილებელია ვიმოქმედოთ,

როგორც მანტისებზე, ასევე რიგებზეც. განვიხილოთ შეკრებისა და გამოკლების

ალგორითმი და არითმეტიკულ-ლოგიკური მოწყობილობის სტრუქტურული სქემა
მცოცავი მძიმით წარმოდგენილი რიცხვებისთვის.

ვთქვათ $X > Y$, მაშინ

$$Z = X \mp Y = 2^{PX} (\pm mx) \pm 2^{PY} (\pm my) = 2^{PX} \left[(\pm mx) \pm \frac{\pm my}{2^{(PX-PY)}} \right]$$

ამ გამოსახულებიდან გამომდინარეობს, რომ $P_Z = P_X$, ხოლო

$$m_z = \left[(\pm m_x) \pm \frac{\pm my}{2^{(PX-PY)}} \right],$$

ჩამოვყალიბოთ ალგორითმი მცოცავი მძიმით წარმოდგენილი რიცხვების
შეკრების და გამოკლების ოპერაციების შესრულების პირობებში.

1. რიცხვის გათანაბრება და მანტისების შეკრება (გამოკლება).

რიცხვის გათანაბრება მდგომარეობს დაბალი რიგის მქონე რიცხვისათვის მეტი
რიცხვის რიგის მინიჭებაში. ჯამის (სხვაობის) რიგი ტოლი იქნება უდიდესი რიცხვის
რიგის. რიგის გაზრდის შემთხვევაში საჭიროა შემცირდეს მანტისა იმდენად რამდენითაც
ერთი რიცხვის რიგი მეტია მეორე რიცხვის რიგზე, ე.ი.

$$\Delta P = P_X - P_Y.$$

თუ $\Delta P = 0$, მაშინ შედეგს მიენიჭება რიგი $P_Z = P_X = P_Y$ და ხორციელდება
მოცემული მანტისების შეკრება (გამოკლება), ისე როგორც ფიქსირებული მძიმით
წარმოდგენილი რიცხვების დროს

$$Z = X \mp Y = 2^{\pm PX} [(\pm mx) \pm (m_y)]$$

თუ $\Delta P > 0$, მაშინ $P_X > P_Y$. Y რიცხვის მანტისას იძვრება მარჯვნივ (შემცირების
მხარეს) ΔP სიდიდით. რიგი $P_Z > P_X$, ხოლო მანტისა $m_z = (\pm m_x) + (\pm m'_y)$, სადაც

$$\pm m'_y = m_y \Delta p.$$

თუ $\Delta P < 0$, მაშინ $P_X < P_Y$. X -ის მანტისა დაიძვრება მარჯვნივ (შემცირების
მხარეს)

ΔP სიდიდით. რიგი $p_z = p$, ხოლო მანტისა

$$m_z = (\pm m'_x) \pm (\pm m'_y), \text{ სადაც } \pm m'_x = \overrightarrow{m_x p}.$$

რიგი $P_z = P$, ხოლო მანტისა

$$m_z = (\pm m'_x) \pm (\pm m'_y), \text{ სადაც } \pm m'_x = \overrightarrow{m_x p}.$$

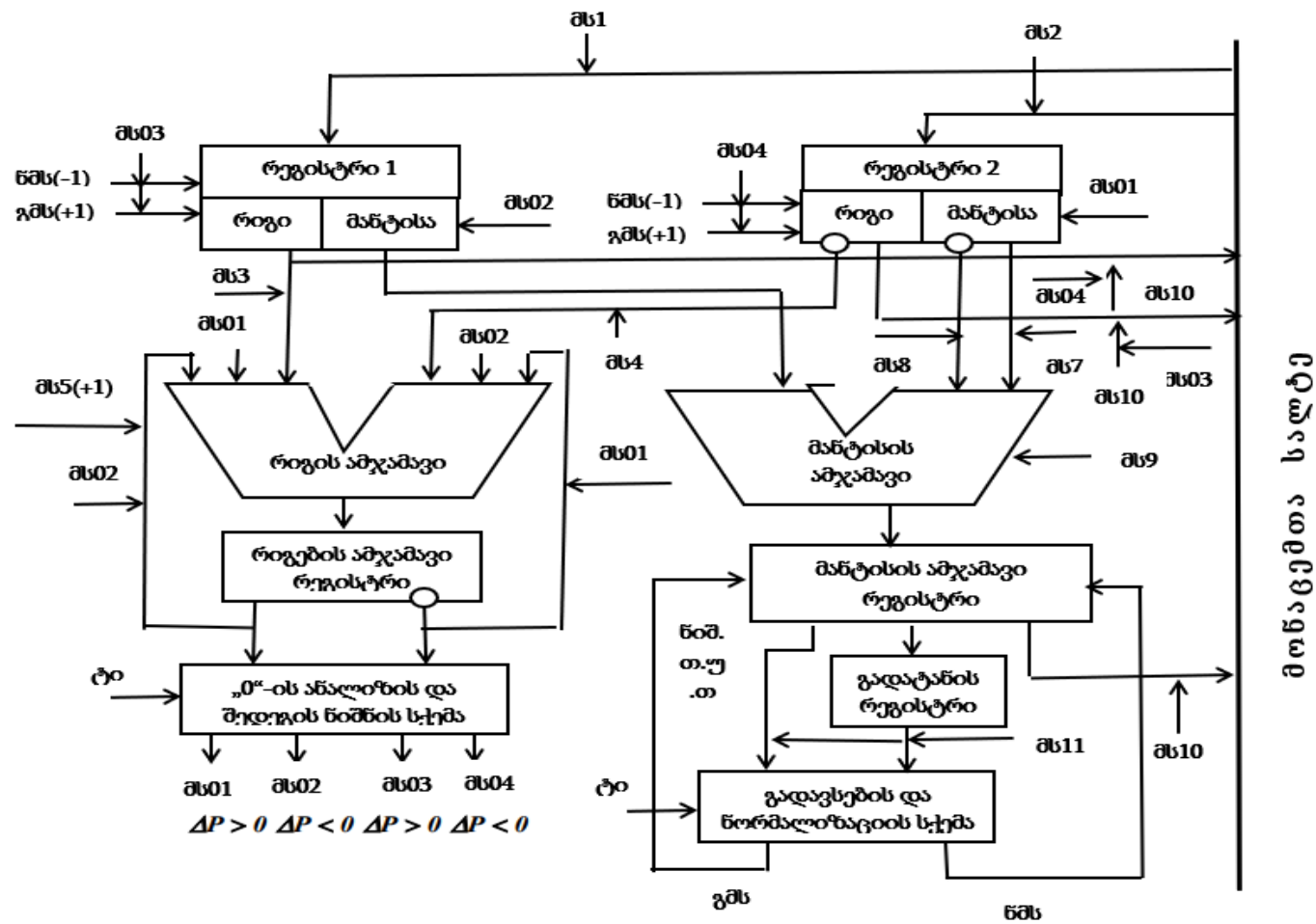
2. შედეგის ნორმალიზაცია. ნორმალიზაცია ხორციელდება იმ შემთხვევაში, როდესაც შედეგის მანტისის უფროს თანრიგებში ჩაწერილია 0 (პირდაპირი კოდი) ან 1 (დამატებითი კოდი) ე.ი. არ სრულდება პირობა

$$1 > |m_z| > 0,5.$$

ნორმალიზაციის დროს ხორციელდება მანტისის დამკრა მარცხნივ მისი გაზრდის მხარეს, მანამ სანამ პირდაპირ კოდში მოცემულ მანტისის უფროს თანრიგში არ გაჩნდება 1. იმავდროულად შედეგის შენარჩუნებისათვის საჭიროა შევამციროთ მისი რიგი იმდენი რიცხვით, რომელიც ტოლი შესრულებული ძვრების რიცხვისა.

ნახ.4.5-ზე მოცემულია არითმეტიკულ-ლოგიკური მოწყობილობის სტრუქტურული სქემა მცოცავი მძიმით წარმოდგენილი რიცხვების შეკრებისა და გამოკლების შესასრულებლად.

სქემაზე მოცემულია რეგისტრები 1 და 2 (რგ 1 და რგ 2) მონაცემთა სალტებიდან მონაცემების მისაღებად, რიგების და მანტისების ამჯამავები (რამჯ და მამჯ) პირველი ოპერანდის რიგიდან მეორე ოპერანდის რიგის გამოკლებისთვის და მანტისების შეკრება გამოკლებისათვის, რიგების ამჯამავის რეგისტრები და მანტისის ამჯამავის რეგისტრები (რამჯ რგ და მამჯ რგ) შედეგების შენახვისათვის, მასზე ნაჩვენებია ასევე “0”-ის ანალიზის, შედეგის ნიშნის და ნორმალიზაციის გადავსების სქემები.



ნახ.4.5

3. “0”-ის და შედეგის ნიშნის ანალიზის სქემა

ეს სქემა უზრუნველყოფს შიგა მმართველი სიგნალების მს01, მს02, მს03, და მს04-ის გამომუშავებას. სიგნალებს მს01 (იმპულსური) და მს03 ადგილი აქვთ იმ შემთხვევაში, როცა რიგების სხვაობა $\Delta P > 0$. მს02 (იმპულსური) და მს04 (პოტენციალური) გენერირდებიან, როცა $\Delta P < 0$. როცა რიგების სხვაობის დადებითი მნიშვნელობა გვაქვს, მაშინ შედეგის რიგს მიაკუთვნებენ პირველი ოპერანდის რიგს, მეორე ოპერანდის რიგი მიიღება შედეგის რიგის მნიშვნელობით ერთდროულად რიცხვის მანტისის დაძვრით მის შემცირების მხარეს. რიგების სხვაობის უარყოფითი მნიშვნელობის დროს, შედეგის რიგს მიენიჭება მეორე ოპერანდის რიგი, პირველი ოპერანდის რიგი მიიღება შედეგის რიგის ტოლად ერთდროულად პირველი რიცხვის მანტისის დაძვრით მისი შემცირების მხარეს. რიგების გათანაბრების შემდეგ ხორციელდება მანტისების შეკრება (გამოკლება). იმპულსური სიგნალები მს01 და მს02 გამოიყენება რიგების ამჯამავის რეგისტრის განულებისათვის (რამჟ რგ). როგორც კი რიგების ამჯამავის რეგისტრის შიგთავსი გახდება 0-ის ტოლი, შეწყდება მს01 და მს02 იმპულსური სიგნალების გამომუშავება და მანტისების ძვრა.

გადავსების და ნორმალიზაციის სქემა. გამოიმუშავებს ორ შიგა სიგნალს - გადავსების მართვის სიგნალი (გმს) და ნორმალიზაციის მართვის სიგნალი (ნმს), რომლებთაგან თითოეული გამოიყენება ნორმალიზაციის დარღვევის აღმოფხვრისათვის. გადავსების მართვის სიგნალის დახმარებით შედეგის მანტისა დაიძვრება მარჯვნივ ერთი თანრიგით, თანრიგთა ბადის გადავსების დროს ($m_z \geq 1$), რითაც აღმოიფხვრება ნორმალიზაციის დარღვევა მარცხნივ. ამასთან ერთით იზრდება შედეგის რიგი. ნორმალიზაციის მართვის სიგნალი აღმოფხვრის მარცხნივ ნორმალიზაციის დარღვევას, როცა შედეგის მანტისა $m_z < 0,5$.

ოპერანდების შეკრებისა და გამოკლების ოპერაციის ალგორითმი მდგომარეობს შემდეგში. მართვის სიგნალი მს1 და მს2 უზრუნველყოფენ ოპერანდების შეტანას რგ1 და რგ2 რეგისტრებში. რიგის ამჯამავის შესასვლელელებზე ერთდროულად მიეწოდება პირველი ოპერანდა (მს3) და მე-2 ოპერანდის რიგის ინვერსიული მნიშვნელობა (მს4), რომელიც აუცილებელია გამოკლების დროს. იგულისხმება, რომ რიგები და მანტისები

მოცემულია დამატებით კოდში. სიგნალი მს5 ახორციელებს რიგების სხვაობის კორექციას (უმცროს თანრიგზე 1-ის დამატებით) და მის წარმოდგენას დამატებით კოდში უარყოფითი შედეგის დროს ან პირდაპირ კოდში დადებითი შედეგის მიღების შემთხვევაში (ნახ. 4.2.).

როცა $\Delta P > 0$ შედეგის ინვერტირებული მნიშვნელობა შეიტანება რიგის ამჯამავში (რამჯ) (მს01) და აიჯამება ამჯამავის მეორე შესასვლელზე მიწოდებული მს01 სიგნალით. აჯამვის პროცესი განხორციელდება ΔP -ჯერ, მანამ სანამ კოდის მნიშვნელობა რიგის ამჯამავ რეგისტრის ინვერსიულ გამოსასვლელებზე მნიშვნელობით არ გახდება ნულის ტოლი. ერთდროულად 2-ე ოპერანდის მანტისა დაიდგრება ΔP თანრიგებით მარჯვნივ.

როცა $\Delta P < 0$, მაშინ შედეგი დამატებით კოდში მიეწოდება რამჯრგ ამჯამავის შესასვლელს (მს02) და აიჯამება სიგნალით მს02, რომელიც მიეწოდება ამჯამავის მეორე შესასვლელს. აჯამვის პროცესი გაგრძელდება მანამ, სანამ კოდების მნიშვნელობა რამჯრგ-ის პირდაპირ გამოსასვლელებზე არ გახდება ნულის ტოლი. ერთდროულად პირველი ოპერანდის მანტისა დაიდგრება Δp თანრიგებზე მარჯვნივ. ორივე შემთხვევაში მს01 და მს02 სიგნალების გამომუშავება შეწყდება. პოტენციალური სიგნალები მს03 და მს04 გამომუშავდება ΔP ნიშნის მიხედვით, რომლებიც უზრუნველყოფენ სქემაში შემდგომ მოქმედებებს (მიკროოპერაციებს).

მანტისების შეკრება (გამოკლება) სრულდება ფიქსირებული მძიმით რიცხვების შეკრების (გამოკლების) ჩვენს მიერ უკვე განხილული წესის მიხედვით. ამასთან მანტისების შეკრების დროს გამოიყენება მმართველი სიგნალები მს6, მს7, მს9 და მს11, ხოლო მანტისების გამოკლების დროს კი მმართველი სიგნალები მს6, მს8, მს9 და მს11. გადავსებისა და ნორმალიზაციის სქემა ანალიზს უკეთებს გადატანების რეგისტრის თანრიგებს. თუ უფროსი და ნიშნის თანრიგებიდან გადატანები არ დაემთხვევა მნიშვნელობით, მაშინ გამომუშავდება გადავსების მართვის სიგნალი (გმს), რომელიც უზრუნველყოფს შედეგის მანტისის დაძვრას მარჯვნივ და რიგის გაზრდას 1-ით მს03 ან მს04 სიგნალებისგან დამოკიდებულებით.

თუ ნიშნის და უფროსი თანრიგები ტოლია 0, ხორციელდება დადებითი შედეგის ნორმალიზაცია მანტისის დამკვრით მარცხნივ მანამ, სანამ უფროს თანრიგში არ გაჩნდება 1. თუ ნიშნის და უფროსი თანრიგები ტოლია 1, ხორციელდება უარყოფითი შედეგის ნორმალიზაცია მანტისის დამკვრით მარცხნივ, მანამ სანამ უფროს თანრიგში არ გაჩნდება 0. მოცემულ შემთხვევაში შედეგის რიგი მცირდება ძვრის რაოდენობის მიხედვით მს03 ან მს04 სიგანალების არსებობისას.

შედეგის მანტისის გაცემა ხორციელდება მანტისის ამჯამავი რეგისტრიდან მს10 მართვის სიგნალით ხოლო შედეგის რიგის გაცემა რგ1 ან რგ2 რეგისტრებიდან ხდება იმისდამიხედვით ადგილი ექნება მს03 თუ მს04 შიგა მმართველი სიგნალების არსებობას. მცოცავი მძიმით რიცხვების შეკრების (გამოკლების) ალგორითმის რეალიზაციისათვის საჭირო მმართველი სიგნალების თანმიმდევრობა აუცილებელი მოქმედებების შესასრულებლად მოცემულია ცხრილში 4.4.

ცხრილი 4.4

	მოქმედების შესრულება	შიგა სიგნალები					მართვის სიგნალები											მიკრო ბრძანებ ები	
		შმსი1	შმსი2	შმსი3	შმსი4	წმ ს	გმ ს	1	2	3	4	5	6	7	8	9	10	11	
1	პირველი ოპერანდის მნიშვნელო ბის შეტანა რეგისტრში 1							1											მბრძ1
2	მეორე ოპერანდის მნიშვნელო ბის შეტანა რეგისტრში 2							1											მბრძ2
3	რიგების შედარება $\Delta p=P_1-P_2$									1	1	1							მბრძ3
4	რიგების სხვაობის ნიშნის ანალიზი, მს01 და მს03	1		1															

	ფორმირება ($\Delta P > 0$)																		
5	პირველი ოპერანდის მანტისის დაძვრა ΔP სიდიდით	1		1															
6	ამჯამავის რგ -ის შემცირება 0-მდე ძვრის რიცხვის დათვლისა თვის	1		1															
7	მანტისების შეკრება და შედგენის ანალიზი			1								1	1		1		1		მბრძ4
8	მანტისების გამოკლება და შედეგის ანალიზი			1								1		1	1		1		მბრძ5

9	გადავსების კორექცია, მანტისის დაძვრა და რიგის გაზრდა			1			1												
10	შედეგის ნორმალიზაცია, მანტისის დაძვრა და რიგის შემცირება			1		1													
11	მანტისის და დიდი რიგის გაცემა მონაცემთა საღებზე			1												1			მბრძ6
12	რიგების სხვაობის ნიშნის ანალიზი, მს 02 , მს 04 ფორმირება ($\Delta P < 0$)		1		1														
13	მეორე ოპერანდის მანტისის		1		1														

	დაბერა ΔP სიდიდით																	
1 4	ამჯამავის რეგისტრის შემცირება 0- მდე ბერის რაოდენობის დათვლისათვის		1		1													
1 5	7,8,9,10 და 11 პუნქტების შესრულება როცა მს03=0 და მს04=1				1													მბრძ 4 მბრძ 5 მბრძ 6

4. 5 საბორტო მიკროპროცესორულ სისტემებში არითმეტიკულ-ლოგიკური მოწყობილობები

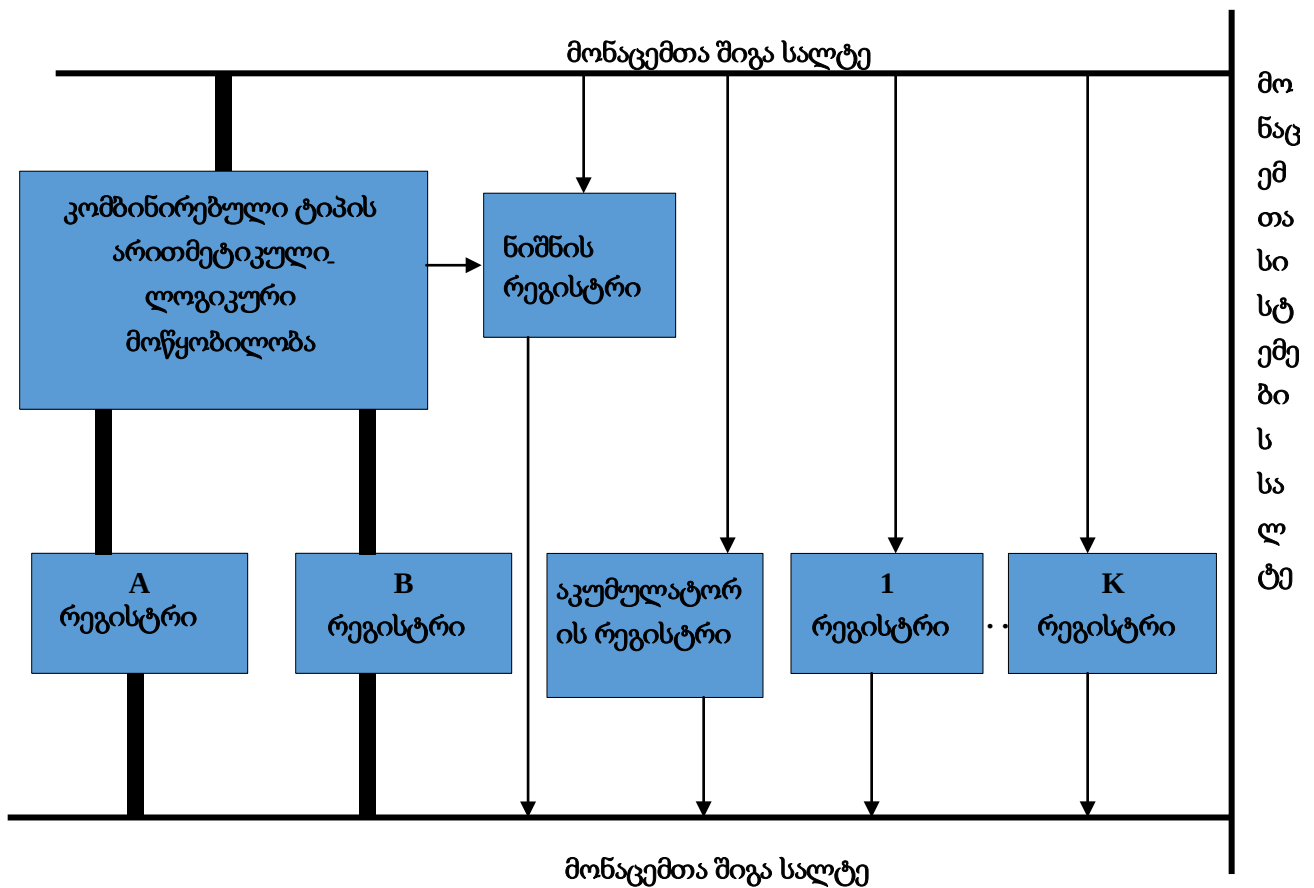
მიკროპროცესორულ სისტემებში ალმ-ის აგების თავისებურება მდგომარეობს იმაში, რომ ოპერანდების შენახვისათვის და გარდაქმნისათვის გამოიყენება შიგა მეხსიერება, რომელიც შედგება რეგისტრების ჯგუფისაგან. ისინი ერთმანეთთან დაკავშირებული არიან შიგა მაგისტრალური სალტით ინფორმაციის გადაგზავნისთვის. მაგისტრალურ სალტზე შეიძლება მიერთებული იყოს ერთი ან რამდენიმე ამჯამავი კოდების შეკრებისათვის (გამოკლებისათვის). ასეთი სტრუქტურა წარმოადგენს კომბინირებული ტიპის ალმ-ს, რომელიც ასრულებს არითმეტიკულ და ლოგიკურ ოპერაციებს ოპერანდებზე, რომლებიც შეიძლება იმყოფებოდნენ შიგა მეხსიერებას ნებისმიერ რეგისტრში.

რეგისტრები გამოიყენება, როგორც ინფორმაციის წყაროებად ასევე მიმღებებად. მოცემულ შემთხვევაში რეგისტრები აუცილებელია ოპერანდების დროებითი დამახსოვრებისათვის. ალმ-ის ასეთი აგების მაგალითი მოცემულია ნახ.4.6-ზე.

განვიხილოთ შეკრების ოპერაციის შესრულების ვარიანტები იმ პირობით, რომ ოპერანდები შესაძლებელია იმყოფებოდნენ ან ოპერატიულ მეხსიერებაში ან რეგისტრებში. ოპერანდების შეკრების ბრძანებების შესრულებისას, როცა ისინი იმყოფებიან რეგისტრებში, ოპერანდები მონაცემების შიგა სალტის საშუალებით გადაიგზავნება არითმეტიკულ-ლოგიკური მოწყობილობის მიმღებ რეგისტრებში -

რეგისტრი A და რეგისტრი B. შედეგი რომელიც ჩაწერილია რეგისტრ-აკუმულატორში შესრულებადი ბრძანების დამოკიდებულების მიხედვითან შეინახება ამ რეგისტრში, ან შეიტანება ერთ-ერთ

დამისამართებად რეგისტრში ან ოპერატიული მეხსიერების უჯრედში.



ნახ.4.6

სამისამართო სალტეები, რომლების საშუალებით გადაიცემა უჯრედის მისამართი ოპერატიულ მეხსიერებაში სქემაზე არ არის ნაჩვენები.

ოპერანდების შეკრების ბრძანებების შესრულების დროს, რომლებიც იმყოფებიან ოპერატიულ მეხსიერებაში, ოპერანდები უშუალოდ შეიტანება ალმ-ის მიმღებ რეგისტრებში, ხოლო შედეგი დაფიქსირდება ან ერთ-ერთ რეგისტრში, ან ოპერატიულ მეხსიერებაში.

ლექცია 5

პროგრამების წყვეტის სისტემა საბორტო კომპიუტერში

5.1 წყვეტის სისტემების ფუნქციები და დანიშნულება

მიმდინარე პროგრამის შესრულების პროცესში საბორტო კომპიუტერმა განსაზღვრულად უნდა მოახდინოს რეაგირება სხვადასხვა სიტუაციებზე, რომლებიც წარმოიქმნება ამოცანის გადაწყვეტის დროს. ასეთი სიტუაციები განპირობებულია საბორტო კომპიუტერის მოწყობილობების, მართვის ობიექტების მდგომარეობების ცვლილებით ან შედეგების მიღებისას, როცა საჭიროა ოპერატორის ჩარევა ან დამატებითი პროგრამების ჩართვა. ასეთ სიტუაციებს შეიძლება მიეკუთვნოს:

1. საბორტო კომპიუტერის მოწყობილობების მუშაობაში შეცდომების აღმოჩენა;
2. არითმეტიკული და ლოგიკური ოპერაციების შესრულების დროს შეცდომების აღმოჩენა;
3. მეხსიერებიდან დასამუშავებელი დამატებითი ინფორმაციის ამოკითხვის აუცილებლობის წარმოქმნა;
4. შესაბამისი გადამწოდიდან მიმდინარე ინფორმაციის შეტანის აუცილებლობა და ა.შ.

ასეთი სიტუაციების წარმოქმნის დროს გაიცემა შეტყობინების სიგნალები, რომელთაც წყვეტის სიგნალები ან წყვეტის მოთხოვნები ეწოდება.

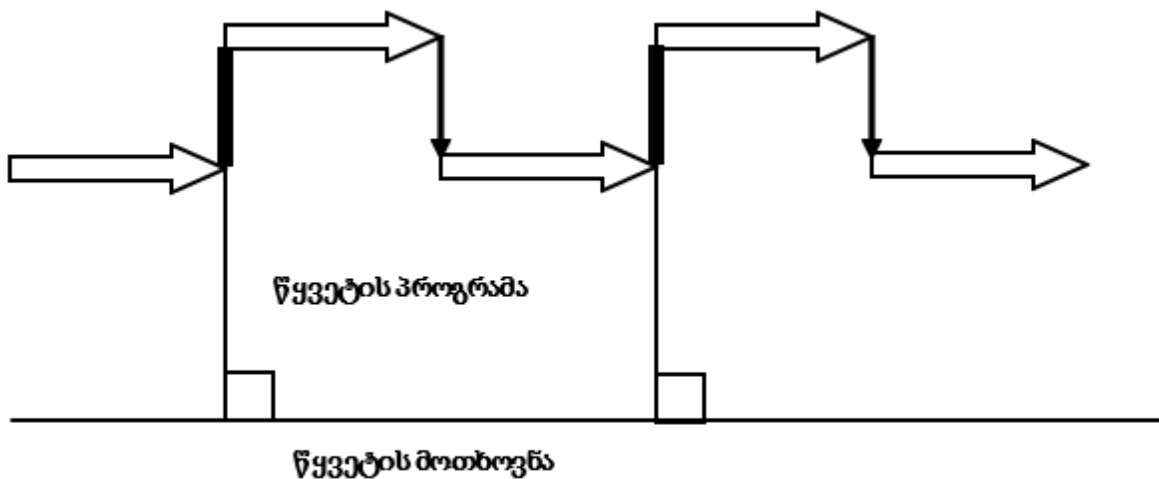
საბორტო კომპიუტერის რეაქცია წყვეტის მოთხოვნებზე ასეთია:

1. მიმდინარე პროგრამის შესრულების შეწყვეტა და გადასვლა სპეციალური პროგრამის შესრულებაზე - შემწყვეტ პროგრამაზე;
2. შეწყვეტილ პროგრამაზე დაბრუნება.

პროგრამების წყვეტის პროცესის ილუსტრაცია წარმოდგენილია ნახ.5.1-ზე.

პროგრამების წყვეტის სისტემა წარმოადგენს აპარატულ-პროგრამულ სისტემას, რომელიც უზრუნველყოფს საბორტო კომპიუტერის რეაქციას მიმდინარე გამოთვლითი პროცესის დროს რაღაც გარე და შიგა მოვლენების ერთობლიობისადმი და თხოულობს ახალ პროგრამაში ბრძანებების სხვა თანმიმდევრობაზე გადასვლას, მის შესრულებას და შემდეგ შეწყვეტილ პროგრამაზე დაბრუნებას.

შემწყვეტი პროგრამები



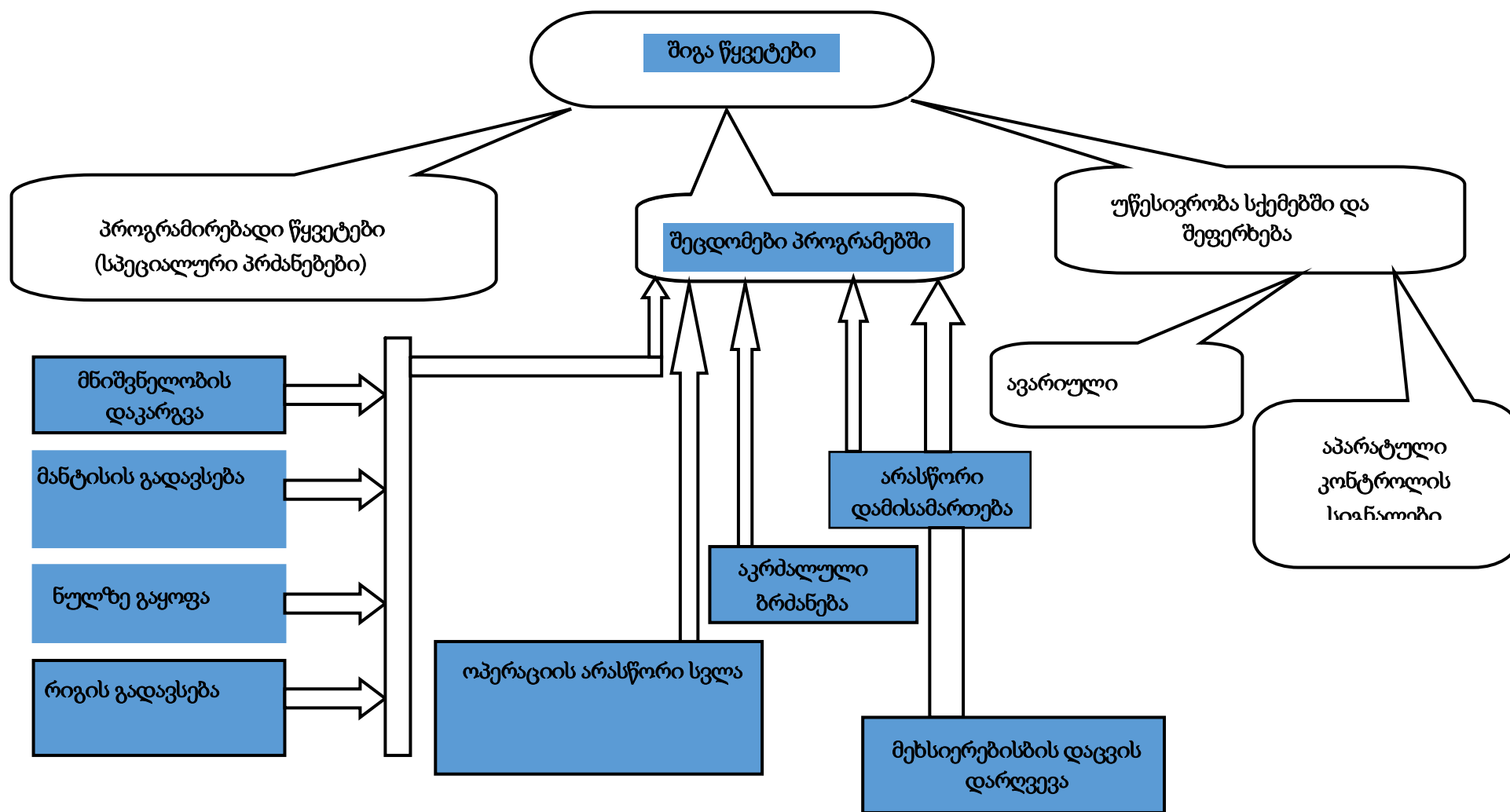
ნახ.5.1.

წყვეტის გამომწვევი მიზეზები უმრავლეს შემთხვევაში განისაზღვრება გამოთვლითი სისტემის დანიშნულებითა და სტრუქტურით. ზოგადად, ყველა სიგნალი, რომელიც იწვევს მიმდინარე პროგრამის შესრულების შეწყვეტას, შეიძლება განზოგადდეს და დაიყოს ოთხ ჯგუფად:

1. სიგნალები მოვლენების შესახებ, რომლებიც წარმოიქმნება თვითონ საბორტო კომპიუტერში (როგორც აპარატურაში, ისე პროგრამებში), მათ შორის სპეციალური ბრძანებებით გამოწვეული სიგნალები;
2. სიგნალები დროითი გადამწოდებიდან;
3. სიგნალები დაკავშირებული ინფორმაციის შეტანა-გამოტანასთან;
4. მიმართვის სიგნალები, რომლებიც მიწოდებულია საბორტო კომპიუტერში მასთან მიერთებული გარე ობიექტებიდან.

სიგნალები, რომლებიც შეადგენენ პირველ ჯგუფს, მიღებულია იწოდებოდეს შიგა სიგნალებად, იმდენად რამდენადაც მათი წარმოშობა განპირობებულია კომპიუტერის შიგნით წარმოშობილი განსაკუთრებული პირობებით: აპარატურაში ან პროგრამებში.

ნახ 5.2-ზე მოცემულია მიზეზთა ჩამონათვალი, რომლებიც წარმოიშობა საბორტო კომპიუტერში და იწვევს პროგრამების წყვეტას. წყვეტის სიგნალების ფორმირება ხორციელდება ჩამოთვლილი მოვლენების დაფიქსირების შემდეგ.



ნახ 5.2. სიგნალები, რომლებიც იწვევენ მიმდინარე პროგრამის შესრულების წყვეტას საბორტო კომპიუტერში

წყვეტები რომლებიც დაკავშირებულია შესრულებადი პროგრამების შეცდომებთან მიღებულია იწოდებოდეს პროგრამულ წყვეტებად. პროგრამული წყვეტები წარმოიშობა შემდეგი მოვლენების არსებობის დროს:

1. ბრძანებებში ოპერაციის კოდი არასწორია;
2. არასწორია ოპერანდების დამისამართება;
3. ადგილი აქვს ნულზე გაყოფის მცდელობას;
4. მცურავ მძიმეიანი რიცხვებისათვის რიგის ან მანტისის წარმოდგენისას ხდება თანრიგთა ბადის გადავსება;
5. თუ არითმეტიკული ოპერაციების შესრულების დროს რიცხვის მანტისა ღებულობს ნულოვან მნიშვნელობას ხდება შედეგის მნიშვნელობის დაკარგვა.

პროგრამულ შეცდომებს მიეკუთვნება შემთხვევა, როცა დარღვეულია პროგრამისთვის განკუთვნილი მეხსიერების დადგენილი საზღვრები და გამოყენებით პროგრამებში იმ ბრძანებების გამოყენება, რომლებიც ნებადართულია, მხოლოდ ოპერაციულ სისტემებში.

მთელი რიგი პროგრამების შესრულებისათვის ფართოდ გამოიყენება სპეციალური ბრძანებები, რომლებიც იწვევენ წყვეტას განსაზღვრული პირობების (პირობითი გადასვლის ბრძანებები) შესრულებისას, ან საჭიროების შემთხვევაში ქვეპროგრამებზე დროებით გადასვლას, მაგალითად, გამრავლების ოპერაციის შესრულების დროს, როდესაც არ არსებობს გამრავლების შესაბამისი ბრძანება .

შიგა წყვეტების წარმოშობა იწვევს მიმდინარე პროგრამის შესრულების შეწყვეტას და პროცესორის გადართვას მოცემული სახის წყვეტის პროგრამის დამუშავებაზე. ასე მაგალითად, საბორტო კომპიუტერის აპარატული კონტროლის სქემიდან წყვეტის სიგნალის წარმოქმნისას წყვეტის სისტემას მოქმედებაში მოჰყავს შეცდომების ანალიზისა და უწყისივრობების აღმოფხვრის პროგრამა. საბორტო კომპიუტერში უწყისივრობის გამო წყვეტის სიგნალების წარმოშობისას, ხორციელდება გადასვლა მომსახურე და დიაგნოსტიკურად პროგრამებზე, რითაც ამოწმებენ შესაბამის მოწყობილობებს და წრედებს უწყისივრობის ადგილის აღმოსაჩენად და მათ გასასწორებლად. ასე მაგალითად, ძაბვის ავტომატური რეგულირების სისტემის ჩართვით შეიძლება დადგინდეს კვების ძაბვის ნომინალური მნიშვნელობა. თუ პროგრამული გზით ან აპარატული საშუალებებით შეუძლებელია უწყისივრობის აღმოფხვრა, მაშინ ფორმირდება და გაიცემა შეტყობინება ოპერატორისთვის (სრული ინფორმაცია უწყისივრობის არსებობისა და

ხასიათის შესახებ). ასეთ შემთხვევაში კომპიუტერის მუშაობა პროგრამის შესრულების მიხედვით ჩერდება ან ხორციელდება ავარიულ რეჟიმში.

დროითი წყვეტები შეადგენს წყვეტის მეორე ჯგუფს და აერთიანებს სიგნალებს, რომლებიც მიეწოდება დროითი გადამწოდებიდან. ასეთი ტიპის წყვეტა დამახასიათებელია გამოთვლითი სისტემისათვის, რომელიც მუშაობს დროის რეალურ მასშტაბში, რომელთა რიცხვს მიეკუთვნება ასევე საბორტო გამოთვლითი სისტემები.

დროის გადამწოდები შეიძლება იყოს პერიოდული სიგნალების ავტონომიური გენერატორები (ტაიმერები), რომლებიც იძლევიან მოცემული სიხშირის და ხანგრძლივობის მართვის სიგნალებს, აბსოლუტური და ფარდობითი დროის მთვლელები, რეალიზებული აპარატული ან პროგრამული საშუალებებით.

წყვეტები, რომლებიც დაკავშირებულია ინფორმაციის შეტანა-გამოტანასთან აღიძვრება მაშინ, როცა ხორციელდება პროგრამის ბრძანების რეალიზაცია ინფორმაციის შეტანის და გამოტანის დროს.

საბორტო გამომთვლით სისტემებში ასეთ მოწყობილობებად გვევლინება ინფორმაციის გარდამქმნელები, რომლებიც წარმოადგენენ შუალედურ რგოლს პროცესორსა და საბორტო ინფორმაციის წყაროს, ან ინფორმაციის მიმღებებს შორის. აღნიშნულს მიეკუთვნება ასევე დაკავების, მზადყოფნის ან შეტანა-გამოტანის მოწყობილობის გაუმართაობის (შეფერხების წარმოქმნის) სიგნალები, ასევე პროგრამის მუშაობაში ოპერატორის ჩარევის სიგნალები. გამომდინარე აქედან, ოპერატორის პულტი და ინფორმაციის ასახვის მოწყობილობა (დისპლეი) წარმოადგენენ თავისებურ შემტან-გამომტან მოწყობილობას.

გარე მოწყობილობებიდან წყვეტის სიგნალების წარმოშობა დაკავშირებულია გარე საბორტო ობიექტების მართვასთან ან დაქვემდებარებულ პროცესორების მუშაობასთან, რომლებიც გაერთიანებული არიან მრავალპროცესორიან საბორტო გამოთვლით სისტემებში.

5.2. წყვეტის სისტემის მახასიათებლები

სხვადასხვა წყვეტის სისტემების შეფასებისა და შედარებისათვის გამოიყენება შემდეგი მახასიათებლები:

1. პროგრამის წყვეტის სიგნალების საერთო რაოდენობა;
2. პროგრამის გადართვისათვის საჭირო დროის დანახარჯები (წყვეტის დანახარჯები);
3. რეაქციის დრო;
4. წყვეტის სიღრმე;
5. პროგრამის წყვეტის დასაშვები მომენტები;
6. წყვეტების მოთხოვნის პრიორიტეტულობა და წყვეტის პროგრამაზე გადასვლის ორგანიზაცია.

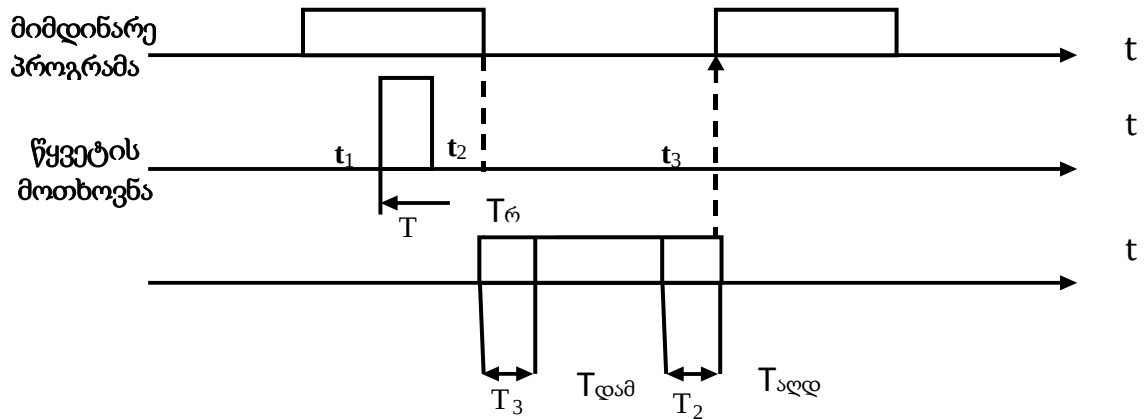
განვიხილოთ ჩამოთვლილი მახასიათებლებიდან თითოეული ცალ-ცალკე.

წყვეტის სიგნალების საერთო რაოდენობა განისაზღვრება შემადგენლობით, ასევე საბორტო მოწყობილობების შესასვლელების და გამოსასვლელების რაოდენობით, რომელთა შემადგენლობაში შედის გამოთვლითი სისტემა. წყვეტის წყაროების რაოდენობასთან დამოკიდებულებით აიგება წყვეტის სისტემა - მისი აპარატული და პროგრამული ნაწილი. სიგნალების დიდი რაოდენობის შემთხვევაში ხდება მათი დაყოფა ერთი ტიპის ჯგუფებად (კლასებად) და წარმოქმნიან წყვეტის ერთ დონეს. იმ შემთხვევაში თუ ჯგუფში ერთდროულად წარმოიშობა რამდენიმე მიმართვა, როგორც წესი, ფორმირდება წყვეტის ერთი სიგნალი და ის ინიცირებას უკეთებს ერთსა და იმავე წყვეტის პროგრამას.

წყვეტის თითოეულ სიგნალს ან სიგნალთა ჯგუფს, რომელიც მიიღება ტიპური წყაროებიდან, წყვეტის სისტემაში უნდა შეესაბამებოდეს მეხსიერების თავისი ელემენტი (მოვლენის ინდიკატორი), რომელიც აფიქსირებს სიგნალს. მეხსიერების ასეთი ელემენტების ერთობლიობას უწოდებენ წყვეტების მიმართვების რეგისტრს (წმრ.). წყვეტის მიმართვების რეგისტრი ღებულობს სიგნალებს წყვეტის წყაროებიდან და ინახავს მანამ, სანამ ისინი მთლიანად არ იქნება დამუშავებული. რეგისტრის თითოეული თანრიგი შეიძლება იყოს ან “ნულის” ან “ერთიანის” მდგომარეობაში, ამ მდგომარეობის შეცვლა ხდება დროის შემთხვევით მომენტში. წყვეტების მიმართვების რეგისტრში დაფიქსირებულ სიგნალების ერთობლიობას ეწოდება წყვეტის კოდი.

შესაძლებელია ერთდროულად წარმოიშვას წყვეტის რამდენიმე პირობა, მაგრამ დროის ყოველ მომენტში ერთპროცესორიან გამოთვლით სისტემაში ხდება წყვეტის მხოლოდ ერთი პირობის მომსახურება.

წყვეტის სისტემების დროითი მახასიათებლები ნაჩვენებია ნახ.5.3-ზე.



ნახ.5.3. წყვეტის სისტემების დროითი მახასიათებლები

რეაქციის დრო T_r ეს არის დრო, წყვეტის მიმართვის გამოჩენასა და შემწყვეტი პროგრამის შესრულების დაწყებას შორის. წყვეტის კონკრეტულ სისტემაში რეაქციის დრო დამოკიდებულია მომსახურების რიგში მდგომი მიმართვების რაოდენობაზე და წყვეტის მიზეზების მნიშვნელობის ხარისხზე. ამიტომ ნაკლები პრიორიტეტის მქონე მოთხოვნებზე რეაქციის დრო ყოველთვის მეტია, ვიდრე უფრო მაღალი პრიორიტეტების მოთხოვნაზე, ვინაიდან ამ უკანასკნელების მომსახურება ხდება პირველ რიგში.

დანახარჯები წყვეტაზე დროის მიხედვით ტოლია პროგრამის დამახსოვრებასა და მისი მდგომარეობის აღდგენაზე დახარჯული დროების ჯამისა

$$T = T_{\text{დამ}} + T_{\text{აღდ.}}$$

t_1 , t_2 და t_3 შესაბამისად განსაზღვრავენ მიმართვის წარმოქმნის, მწყვეტარა პროგრამის დასაწყისის და შემწყვეტილი პროგრამის მდგომარეობის დამახსოვრების მომენტებს, ასევე მიმდინარე პროგრამის შესრულების აღდგენას.

წყვეტის სიღრმე განისაზღვრება პროგრამების რაოდენობით, რომლებსაც შეუძლიათ ერთმანეთის წყვეტა. თუ პროცესორში მიმდინარე პროგრამის შემწყვეტა

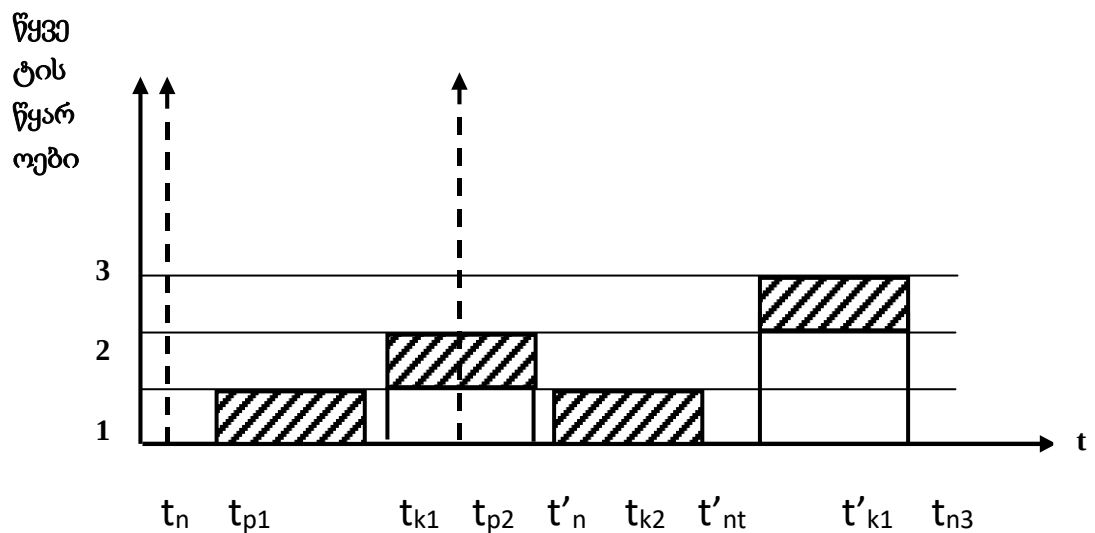
ხდება მხოლოდ ერთი პროგრამით, რომელიც სრულდება ბოლომდე, მაშინ ასეთი წყვეტის სისტემა მოიცავს წყვეტის სიღრმეს, რომელიც ტოლია ერთის, თუ მწყვეტარა პროგრამა, შესაძლებელია თავის მხრივ შეწყვეტილ იქნეს სხვა პროგრამით, მაშინ სისტემის წყვეტის სიღრმე ტოლია ორის და ა.შ. სისტემებში წყვეტის სიღრმის დიდი მნიშვნელობების შემთხვევაში მიიღწევა უფრო სწრაფი რეაქცია მიმართვებზე და მოითხოვენ დაუყოვნებლივ დამუშავებას.

ძირითადად პროგრამების წყვეტა სრულდება მიმდინარე ბრძანების შესრულების შემდეგ, გამომდინარე აქედან, რეაქციის დრო განისაზღვრება ბრძანების შესრულების ხანგრძლივობით. დროის რეალურ მასშტაბში მუშაობის დროს და პროცესორის არა საკმარისი სწრაფქმედების შემთხვევაში შესაძლებელია წყვეტა მოხდეს ბრძანების ნებისმიერ ტაქტზე, რაც ზრდის დასამახსოვრებელი და აღსადგენი ინფორმაციის რაოდენობას.

როგორც ზემოთ იყო აღნიშნული, წყვეტის მიმართვის პრიორიტეტი გულისხმობს მისი საჭიროებისა და სწრაფად შესრულების მოთხოვნის მაღალ ხარისხს სხვებთან შედარებით. უფრო პრიორიტეტული წყვეტის პროგრამები სრულდებიან პირველ რიგში.

პრიორიტეტი შეიძლება იყოს ორი სახის: აბსოლუტური ან ფარდობითი. იმასთან დაკავშირებით, თუ რომელი პრიორიტეტი გამოიყენება სისტემაში, ორგანიზაცია უკეთდება შესაბამის პროცედურას.

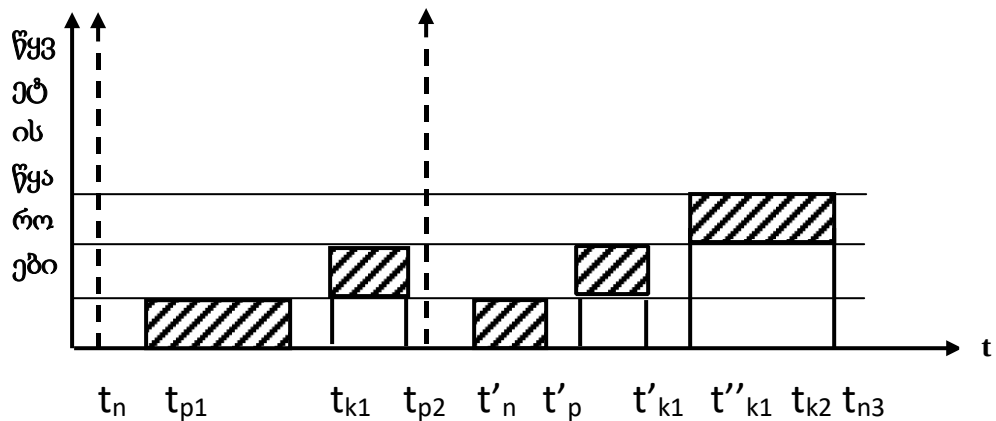
ფარდობითი პრიორიტეტით მომსახურების პროცედურა მდგომარეობს იმაში, რომ წყვეტის მოთხოვნათა სიმრავლიდან ამოირჩევა ყველაზე უფრო პრიორიტეტული მიმართვა (მოთხოვნა) და ხდება მისი მომსახურება. თუ ამ წყვეტის მომსახურების დროს მოდის მოთხოვნა უფრო მაღალი პრიორიტეტით, მაშინ ის დგება რიგში მომსახურებაზე და მისი დამუშავება მოხდება მხოლოდ მაშინ, როცა დამთავრდება წინამდებარე წყვეტის პროგრამის მომსახურება (ნახ. 5.4).



ნახ. 5.4. სისტემაში ერთდროულად სხვადასხვა პრიორიტეტით მოსული სამი მოთხოვნა დროის მომენტში

ნახ.5.4-ზე წარმოდგენილია სამი მოთხოვნა სხვადასხვა პრიორიტეტებით, რმლებიც სისტემას მიეწოდებიან ერთდროულად დროის მომენტში t_n , ამასთან $1 > 2 > 3$. პირველად ხორციელდება პირველი ($t_{p1} - t_{k1}$) მოთხოვნის მომსახურება, შემდეგ მეორე ($t_{p2} - t_{k2}$) მოთხოვნის მომსახურება. თუ მე-2-ის მომსახურების მომენტში კვლავ მოვიდა 1-ის მიმართვა მაღალი პრიორიტეტით, მაშინ მისი მომსახურება შეჩერდება მანამ, სანამ არ დამთავრდება მე -2-ის მომსახურება. ბოლო რიგით მოხდება მე-3 მიმართვის მომსახურება.

აბსოლუტური პრიორიტეტებით მომსახურების პროცედურა მდგომარეობს იმაში, რომ ყოველ დროის მომენტში ყოველთვის დამუშავდება ყველაზე პრიორიტეტული მოთხოვნა იმისდა მიუხედავად, თუ რამდენი მოთხოვნა მოვიდა რეგისტრში. შედარებით პრიორიტეტული მოთხოვნის გაჩენის დროს, ხდება მიმდინარე წყვეტის მომსახურების შეჩერება და გადასვლა მაღალი პრიორიტეტის მეორე მოთხოვნის მომსახურებაზე (ნახ.5.5).

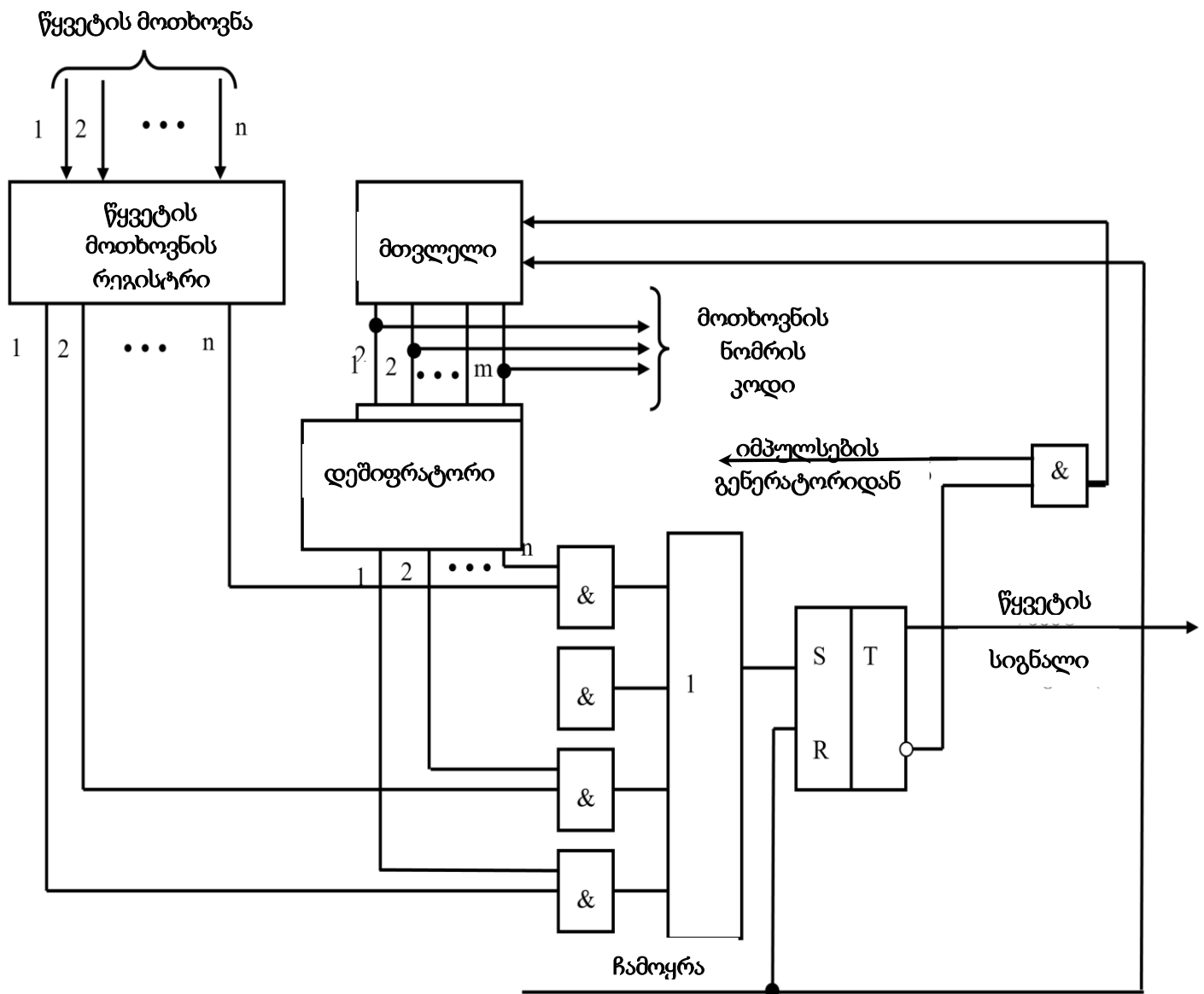


ნახ. 5.5. მიმართვების მომსახურების პროცედურის დროითი დიაგრამა
 აბსოლუტური პრიორიტეტების დროს

როგორც პირველ შემთხვევაში, აქაც პირველ რიგში ხდება უფრო პრიორიტეტული პირველი მოთხოვნის მომსახურება, შემდეგ კი მიმდევრობით მეორე მოთხოვნის მომსახურება. მეორე მოთხოვნის მომსახურების დროს კვლავ თუ წარმოიქმნება პირველი მოთხოვნა, მაშინ მოხდება წყვეტა მეორე მოთხოვნის შესრულებაზე და გადასვლა პირველი წყვეტის მომსახურებაზე. წყვეტის სისტემებში აბსოლუტური ან ფარდობითი პრიორიტეტებით მომსახურება ხისტადაა დაფიქსირებული და განისაზღვრება სისტემის შესასვლელელებზე მიერთების რიგით. დროის თითოეულ მომენტში განსაკუთრებული პრიორიტეტული მოთხოვნის განსაზღვრა ხორციელდება პრიორიტეტის ამორჩევის სპეციალური სქემით.

პრიორიტეტის ამორჩევის სქემა შეიძლება აგებულ იქნეს ნებისმიერი ამორჩევის პრინციპით ერთი ტაქტის განმავლობაში ყველაზე პრიორიტეტული მოთხოვნის უზრუნველყოფით, ან რამდენიმე ტაქტის განმავლობაში ციკლური (მიმდევრობითი) გამოკითხვით.

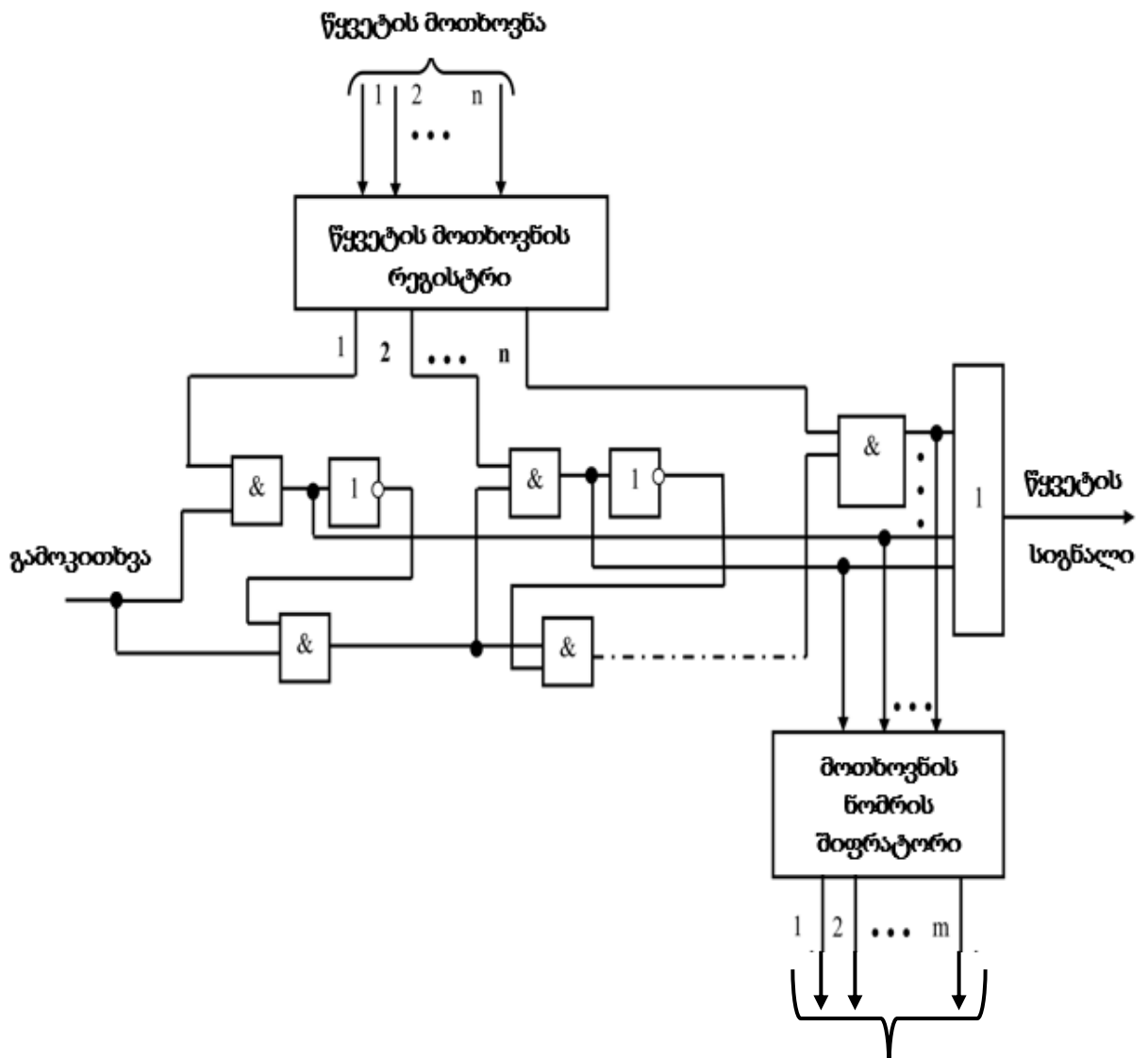
ნახ. 5.6-ზე წარმოდგენილია წყვეტის წყაროების გამოკითხვის ციკლური სქემა. წყვეტის მოთხოვნის რეგისტრების თანრიგების გამოკითხვა ხორციელდება მიმდევრობით იმპულსების გამანაწილებლის დახმარებით, რომელიც აგებულია ორობითი მთვლელის და დემიფრატორის გამოყენებით.



ნახ. 5.6. წვეტის წყაროების გამოკითხვის ციკლური სქემა.

გარე გენერატორიდან მიწოდებული იმპულსები გარდაიქმნება მიმდევრობით სიგნალებად დეშიფრატორების გამოსასვლელებზე ვენტილური სქემების მართვისათვის (ლოგიკური ელემენტები “და”). ვენტილის მეორე შესასვლელები დაკავშირებულია წვეტის მოთხოვნის რეგისტრის შესაბამის თანრიგებთან, ამასთან რეგისტრის პირველი თანრიგი, რომლისგანაც იწყება მიმდევრობითი გამოკითხვა, დაკავშირებულია ყველაზე პრიორიტეტულ წვეტის წყაროსთან. მომდევნო თანრიგების პრიორიტეტულობა მცირდება. თუ რეგისტრის შესაბამისი თანრიგი იმყოფება ერთიანის მდგომარეობაში, მაშინ წვეტის ტრიგერი ასევე დაყენდება “1”-ში და გამომუშავდება წვეტის საერთო სიგნალი. წვეტების მოთხოვნების რეგისტრის დანარჩენი თანრიგების გამოკითხვა

წყდება, ხოლო რეგისტრის შესაბამისი თანრიგი, გამოკითხვის მთვლელი და წვეტის ტრიგერი დადგება “0”-ში.



ნახ. 5.7. პრიორიტეტის ამორჩევის სქემა

ნახ.5.7-ზე ნაჩვენებია პრიორიტეტის ამორჩევის სქემის ვარიანტი, რომელიც აგებულია ნებისმიერი (თავისუფალი) ამორჩევის პრინციპის მიხედვით.

სქემის მუშაობა იწყება პირველ ვენტილზე (ელემენტი და) სიგნალ “გამოკითხვა” მიწოდების მომენტიდან. თუ წვეტების მოთხოვნის რეგისტრის პირველი თანრიგი იმყოფება “0”-ში, მაშინ სიგნალი “გამოკითხვა” მიეწოდება მეორე ვენტილს. დავუშვათ აქვს ადგილი მოთხოვნას მე - 2 პრიორიტეტის მიხედვით, ე.ი. მოთხოვნების რეგისტრის მეორე თანრიგი დაყენებულია “1”-ის მდგომარეობაში. ასეთ შემთხვევაში სიგნალის “გამოკითხვა” შემდგომი გავრცელება იბლოკება და ფორმირდება პროგრამების წვეტის

სიგნალი და პრიორიტეტული მოთხოვნის ნომრის კოდი, რომელიც გამოიყენება პროცესორში შემწყვეტი პროგრამის საწყისი მისამართის ფორმირებისათვის.

5.3. წყვეტის ხერხები

5.3.1. ვექტორული წყვეტა

პრიორიტეტული მოთხოვნის განსაზღვრის ზემოთ განხილული ხერხები სპეციალური სქემის გამოყენებით ხასიათდება იმ ნაკლოვანებით, რომ მწყვეტარა პროგრამაზე გადასვლისათვის მათ ჭირდებათ შედარებით დიდი დრო, ე.ი. T რეაქციის დრო დიდია.

თანამედროვე მიკროპროცესორულ სისტემებში ამ ნაკლოვანების გამოსასწორებლად გამოყენება ჰპოვა ე.წ. ვექტორულმა წყვეტამ. მეთოდის არსი დაფუძნებულია ვექტორული წყვეტის გამოყენებაზე და მდგომარეობს იმაში, რომ წყვეტის წყარო წყვეტის მოთხოვნასთან ერთად პროცესორის სისტემურ სალტებზე აყენებს თავისი წყვეტის ვექტორის მისამართის კოდს მეხსიერებაში.

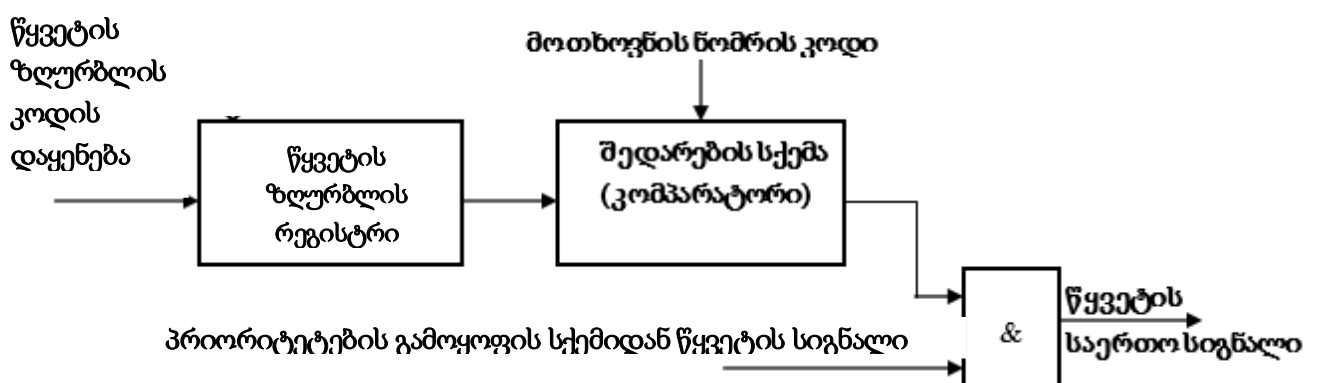
ვექტორული წყვეტისათვის ნიშანდობლივია ისეთი ცნებები, როგორიცაა “წყვეტის ვექტორი” და “მდგომარეობის ვექტორი”. მდგომარეობის ვექტორი და წყვეტის ვექტორი იდენტურია ინფორმაციული შინაარსით. ისინი თავის თავში შეიცავენ ინფორმაციას ბრძანების მთვლელის მდგომარეობის შესახებ (პროგრამის მორიგი ბრძანების მისამართს), შესასრულებელი ოპერაციის ნიშან-თვისებების რეგისტრს, წყვეტის ზღურბლს და სხვა ინფორმაციას, რომელიც საჭიროა ამა თუ იმ პროგრამის შესრულების გაგრძელებისათვის. სხვადასხვა გამოთვლით სისტემებში ვექტორების შინაარსი შეიძლება განსხვავდებოდეს ერთმანეთისაგან. მდგომარეობის ვექტორი ფორმირდება მიმდინარე პროგრამის შესრულების პროცესში. წყვეტების შემთხვევაში ხორციელდება ერთი ვექტორის შეცვლა მეორეთი, რაც უზრუნველყოფს წყვეტის პროგრამის შესრულებას.

5.3.2. წყვეტა პროგრამულად მართვადი პრიორიტეტით

საბორტო სისტემების მართვის პროცესში წარმოიქმნება სიტუაციები, როცა აუცილებელია გადაფასდეს და შეიცვალოს პროგრამების ადრე დაყენებული პრიორიტეტები გამოთვლით სისტემაში. ამ მიზნით რეალიზდება წყვეტის სისტემები მართვადი პრიორიტეტებით. მართვადი პრიორიტეტების არსი მდგომარეობს იმაში, რომ გამოთვლით სისტემის ფუნქციონირების დროს გამოთვლითი პოცესის მიმდინარეობისას სპეციალური ბრძანებებით შეიცვლება პროგრამების პრიორიტეტები და დაყენდება ახალი პრიორიტეტული თანაფარდობა. ეს კი პროგრამის წყვეტის სისტემებს ანიჭებს მოქნილობას და დინამიურობას.

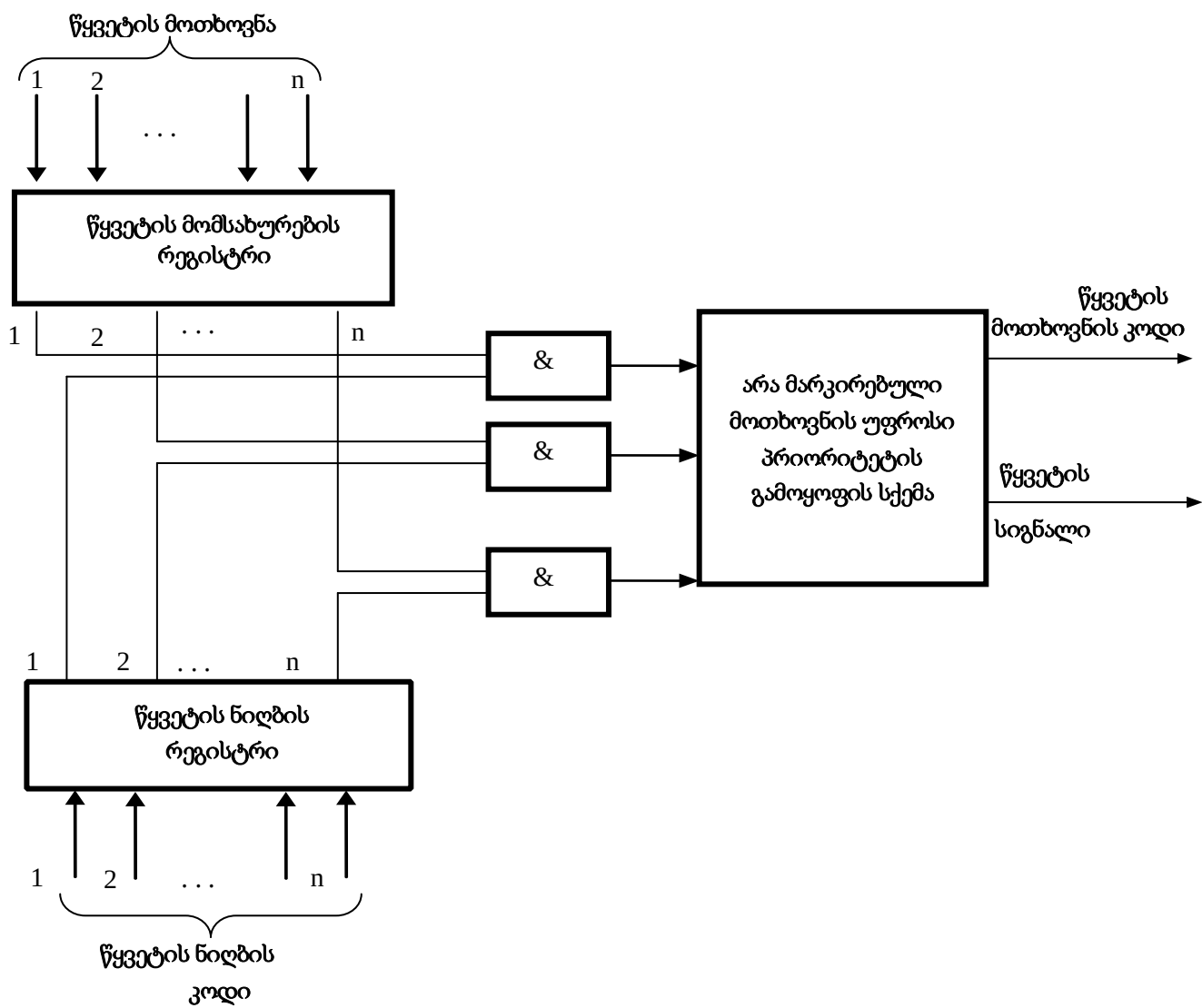
ყველაზე ფართოდ გამოიყენება მწყვეტარა პროგრამების პროგრამულად მართვადი პრიორიტეტის ორი ხერხი - წყვეტების ზღურბლის და წყვეტის ნილბის გამოყენებით.

წყვეტების ზღურბლის გამოყენების შემთხვევაში წყვეტის წყაროს თითოეულ ჯგუფს მიენიჭება სპეციალური კოდი (წყვეტის მოთხოვნის ნომერი), რომელიც გამოიყენება პროცესორში წყვეტის ზღურბლის რეგისტრში დაყენებულ კოდთან შედარებისათვის. მოთხოვნის ნომრის კოდს და წყვეტის ზღურბლს განსაზღვრავენ პროგრამების პრიორიტეტების დონით. თუ შემოსული მოთხოვნის ნომრის კოდი მეტია პროცესორში დაყენებულ წყვეტის ზღურბლზე, მაშინ გამომუშავდება წყვეტის საერთო სიგნალი და იწყება წყვეტის პროცედურა (ნახ. 5.8).



ნახ.5.8.

წყვეტის ნილაბი წარმოადგენს ორობით კოდს, რომლის თანრიგები შეესაბამება წყვეტის მოთხოვნების რეგისტრის თანრიგებს. პროგრამული გზით სპეციალური ბრძანებების დახმარებით ნილაბი ჩაიწერება ნილბის რეგისტრში (ნახ.5.9). თუ ნილბის შესაბამის თანრიგში შეტანილია “1”, მაშინ წყვეტის მოთხოვნა მოცემული ჯგუფიდან ნებადართულია. “0”-ის ჩაწერის დროს წყვეტა აკრძალულია (შენილბულია). მიმდინარე პროგრამის შესრულების პროცესში წყვეტის ნილბის ცვლილებით, შესაძლებელია შემოსული მოთხოვნების პრიორიტეტების ვარირება. თითოეული მწყვეტარა პროგრამა აყენებს თავის წყვეტის ნილაბს, ამასთან ნილბის ფორმირების დროს “1”-ანი ყენდება იმ თანრიგებში, რომელებიც შეესაბამებიან შედარებით მაღალი პრიორიტეტის მქონე მოთხოვნებს. შენილბული მოთხოვნები ან იგნორირდება, ან ხდება მათი დამახსოვრება და ბოლო რიგით მომსახურება.

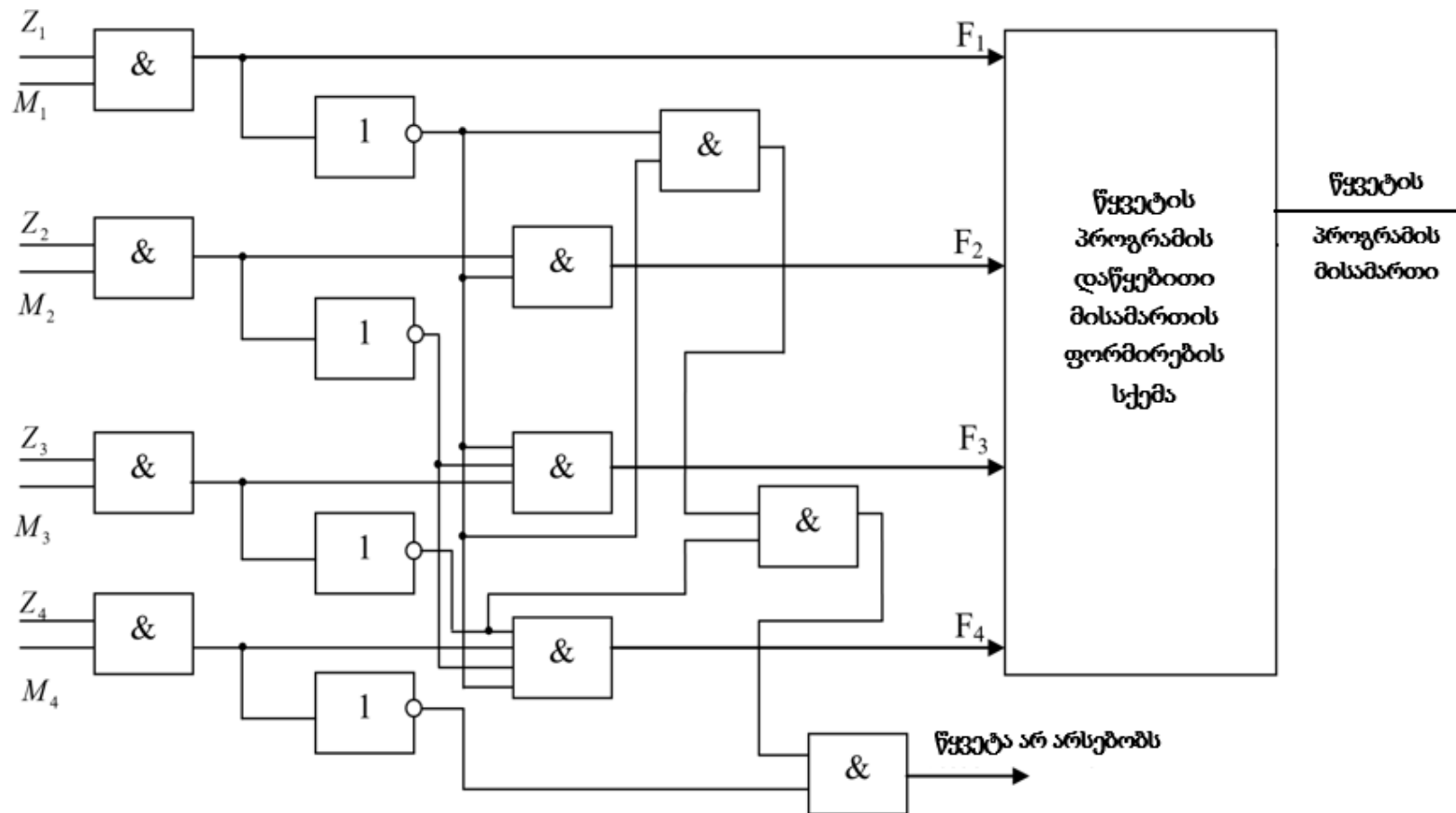


ნახ.5.9

უფროსი პრიორიტეტის არაშენილბული მოთხოვნის გამოყოფის სქემა აიგება, როგორც ლოგიკური სქემა (ნახ.5.10), რომელიც რეალიზაციას უკეთებს განტოლებათა შემდეგ სისტემას:

$$\begin{aligned} F_1 &= Z_1 \cdot M_1 \\ F_2 &= Z_2 \cdot M_2 (\overline{Z_1 \cdot M_1}) \\ F_3 &= Z_3 \cdot M_3 (\overline{Z_2 \cdot M_2}) (\overline{Z_1 \cdot M_1}) \\ &\vdots \\ F_n &= Z_n \cdot M_n (\overline{Z_3 \cdot M_3}) (\overline{Z_2 \cdot M_2}) (\overline{Z_1 \cdot M_1}) \end{aligned} \quad (5.1)$$

(5.1) განტოლებათა სისტემაში Z_i, M_i - შესაბამისად წარმოადგენენ მოთხოვნის რეგისტრის და ნილბის რეგისტრის თანრიგების მნიშვნელობებს.



5.4. საბორტო კომპიუტერების მუშაობის რეჟიმები გარე მოწყობილობებთან მუშაობის დროს და წყვეტების ინფორმაციული გაცვლის ორგანიზაციის განსაკუთრებულობები

იმასთან დამოკიდებულების მიხედვით თუ ინფორმაციის გაცვლისას საბორტო კომპიუტერსა და გარე მოწყობილობებს შორის რომელი წარმოადგენს ინიციატორს და როგორ ხორციელდება ინფორმაციის გადაცემა, განასხვავებენ ინფორმაციული გაცვლის სამ ძირითად ხერხს საბორტო კომპიუტერსა და გარე მოწყობილობებს შორის:

- პროგრამული გაცვლა;
- გაცვლა წყვეტების მიხედვით;
- გაცვლა პირდაპირი დაშვების არხით.

პროგრამული გაცვლის ინიციატორს წარმოადგენს საბორტო კომპიუტერის პროცესორი. გაცვლის მიზეზს წარმოადგენს მონაცემების შეტანის ან გამოტანის სპეციალური ბრძანება, რომელიც პროცესორის მიერ ამოკითხულია მეხსიერებიდან. ისე როგორც ნებისმიერ სხვა ბრძანებაში გაცვლის ბრძანებაში მითითებულია ინფორმაციის წყაროსა და მიმღების მისამართი. ბრძანების ფორმატის სიგრძის შემცირებისთვის ერთ-ერთი მისამართებიდან შესაძლებელია იყოს საგულისხმო მისამართი.

წყვეტების მიხედვით გაცვლა ხორციელდება გარე მოწყობილობის ინიციატივით. ინფორმაციული გაცვლისათვის მზადყოფნის შემთხვევაში გარე მოწყობილობა აფორმირებს მიმართვას მიმდინარე (შესრულებაში მყოფი) პროგრამის შეწყვეტაზე. პროცესორი ანალიზს უკეთებს წყვეტის მიმართვის პრიორიტეტის დონეს, და მიმართვის უფრო მნიშვნელოვანი პრიორიტეტის შემთხვევაში, იძლევა ნებართვას შესრულებაში მყოფი მიმდინარე პროგრამის წყვეტაზე. მიმდინარე პროგრამა ჩერდება და ყველა მისი ატრიბუტი ინახება სტეკში. ჩაიტვირთება ინფორმაციული გაცვლის პროგრამა იმ მოწყობილობით, რომელმაც დააფორმირა მიმართვა. გაცვლის დამთავრების შემდეგ ხდება შეწყვეტილი პროგრამის აღდგენა. გაცვლა ხორციელდება ჩვეულებრივ მცირე ზომის მონაცემებით.

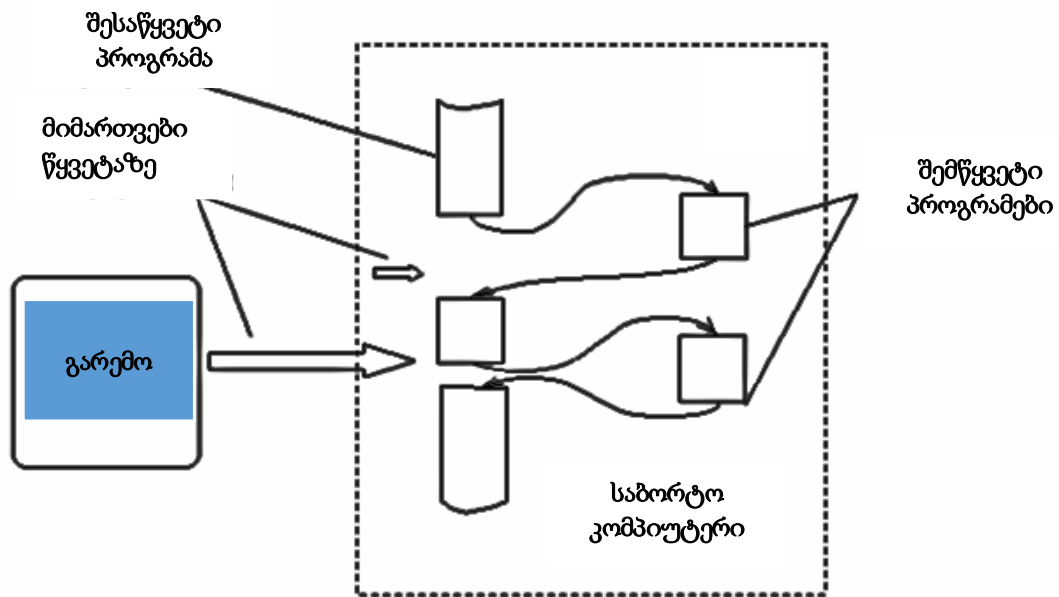
პირდაპირი დაშვების არხის საშუალებით ხორციელდება დიდი მოცულობის მონაცემების გაცვლა. ასეთ შემთხვევაში ინიციატორს წარმოადგენს გარე მოწყობილობა. ის აფორმირებს მიმართვას მონაცემების სალტის მიხედვით, და მიიღებს რა ნებართვას

პროცესორისგან პირდაპირ ჩაწერს ან ამოიკითხავს მონაცემებს ოპერატიულ მეხსიერებაში.

წყვეტის სისტემა არა მარტო უზრუნველყოფს ინფორმაციული გაცვლის ორგანიზაციას საბორტო კომპიუტერსა და გარე მოწყობილობებს შორის, როცა გადასაცემი შეტყობინებების სიგრძე მოითხოვს შესრულებაში მყოფი პროგრამის შეწყვეტას, ასევე უზრუნველყოფს პროცესორის დროულ რეაქციას სხვადასხვა სიტუაციების წარმოშობის შემთხვევაში, როგორც პროცესორში, ისე გარე გარემოში. წყვეტის სიტემის საჭიროება და მნიშვნელობა მოითხოვს მოხდეს მისი აგების პრინციპების და ფუნქციონირების უფრო დეტალური განხილვა.

საბორტო კომპიუტერის მიერ მიმდინარე პროგრამის შესრულების დროს მის შიგნით ან მასთან დაკავშირებულ გარე გარემოში (მაგალითად, რადიოელექტრონულ დანადგარში, რომელიც იმართება საბორტო კომპიუტერით) შესაძლოა აღიმართოს მოვლენები, რომლებიც მოითხოვენ დაუყოვნებელ რეაქციას საბორტო კომპიუტერის მხრიდან. ეს რეაქცია გულისხმობს იმას, რომ ასეთი მოვლენების წარმოშობისას საბორტო კომპიუტერმა უნდა შეწყვიტოს მიმდინარე პროგრამის დამუშავება და უნდა გადავიდეს რაღაც სხვა პროგრამის შესრულებაზე, რომელიც სპეციალურადაა გათვალისწინებული იმ სიტუაციისათვის, რომელიც დაკავშირებულია ამ მოვლენის გამოჩენასთან. ამ პროგრამის დასრულების შემდეგ კომპიუტერი აუცილებლად უნდა დაბრუნდეს შეწყვეტილი პროგრამის შესრულებაზე.

განსახილველ პროცესს, რომელიც იწოდება პროგრამების წყვეტად, შეგვიძლია ილუსტრაცია გავუკეთოთ ნახ.5.11-ით.



ნახ.5.11. პროგრამების წყვეტა

პრინციპულად უმნიშვნელოვანესს წარმოადგენს ის, რომ მოვლენათა წარმოშობის მომენტები, რომლებიც მოითხოვენ პროგრამების წყვეტას, არაა ცნობილი და ამიტომ ისინი შეუძლებელია წინასწარ იქნეს გათვალისწინებული დაპროგრამებისას.

თითოეული მოვლენა, რომელიც მოითხოვს წყვეტას, უნდა შეცავდეს სიგნალს, რომელიც შეატყობინებს საბორტო კომპიუტერს მისი წარმოშობის შესახებ. ეს სიგნალი იწოდება წყვეტის მიმართვად. პროგრამები, რომლებიც მოთხოვნილია წყვეტის მიმართვებით, იწოდებიან შემწყვეტ პროგრამებად, განსხვავებით შეწყვეტადი პროგრამებისგან, რომლებიც სრულდებიან საბორტო კომპიუტერის მიერ მიმართვების გამოჩენამდე.

როგორც ნახ.5.11-იდან ჩანს, მიმართვები წყვეტაზე შეიძლება წარმოიშვას (აღიძვრას) თვითონ შიგნით და მის გარე გარემოში. პირველს მიეკუთვნება, მაგალითად, საბორტო კომპიუტერში წარმოშობილი ისეთი მოვლენები, როგორიცაა შეერთებები მის აპარატურაში, თანრიგთა ბადის გადავსება, ნულზე გაყოფის მცდელობა, დამისამართების წესების დარღვევა, შეტანა-გამოტანის ოპერაციების დასრულება და ა.შ.

წყვეტის მიმართვები გარემოში შესაძლებელია აღიძვრას სხვა საბორტო კომპიუტერებიდან, რადიოლოკაციური, ოპტიკური და თვალმიდევნების სხვა სისტემებისგან, ასევე სხვადასხვა ავარიული გადამწოდებიდან.

საბორტო კომპიუტერმა წყვეტისას, დროებით უნდა შეწყვიტოს მიმდინარე პროგრამის შესრულება და გადავიდეს იმ პროგრამის შესრულებაზე, რომელიც შეესაბამება წყვეტის მიმართვას, ხოლო მისი დასრულების შემდეგ უნდა დაუბრუნდეს შეწყვეტილი პროგრამის შესრულებას. ჩამოთვლილი მოქმედებების რეალიზაციისათვის საბორტო კომპიუტერის მართვის მოწყობილობის შემადგენლობაში შედის სპეციალური აპარატული და პროგრამული საშუალებები, რომლებიც იწოდებიან წყვეტის სისტემად. იმ შემთხვევაში როცა ადგილი აქვს წყვეტის სხვადასხვა წყაროებიდან წყვეტაზე ერთდროულ მიმართვას მათი დამოუკიდებლად გამომუშავების დროს, საჭიროა აუცილებლად დადგინდეს მათი (მიმართვების) მომსახურების განსაზღვრული რიგი.

არსებობენ სისტემები წყვეტისა მიმართვების თანმიმდევრობითი ნახვით (სკანირებით) და მათი მომსახურებით მათდამი მინიჭებული პრიორიტეტის რიგის მიხედვით. ასეთი სისტემები იწოდებიან პრიორიტეტულ სისტემებად.

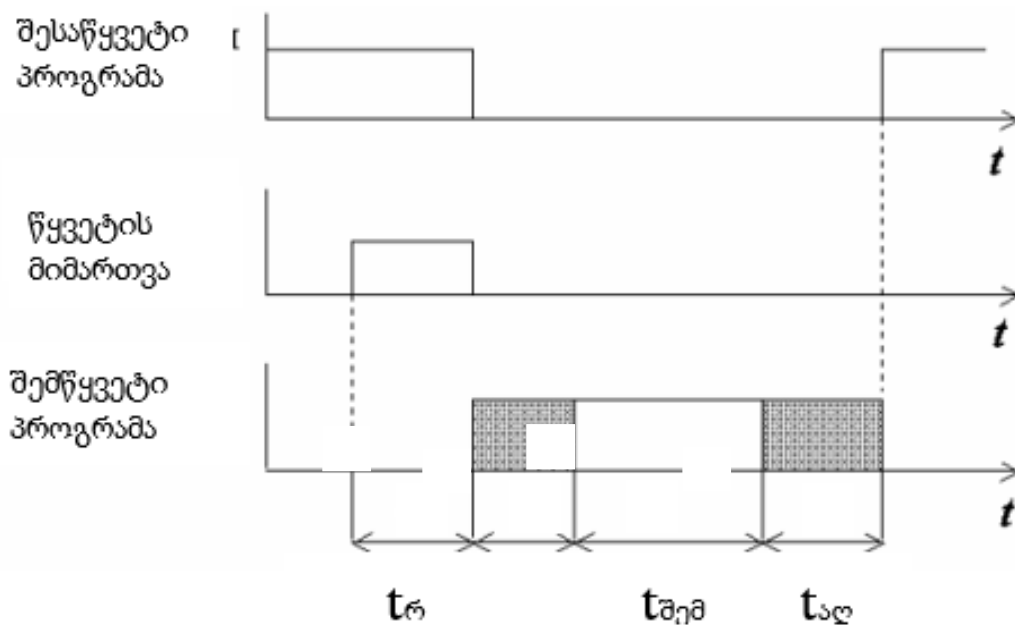
წყვეტის სისტემები ასრულებენ შემდეგ ფუნქციებს:

- შესაწყვეტი პროგრამის მდგომარეობის დამახსოვრება;
- შეწყვეტაზე შემოსული მიმართვების ჯგუფიდან მიმართვის პრიორიტეტის მიხედვით არჩევა და შესაწყვეტ პროგრამაზე გადასვლის ორგანიზაცია;
- შეწყვეტილი პროგრამის მდგომარეობის აღდგენა და მასზე გადასვლა;
- მიმართვების პრიორიტეტების პროგრამული ცვლილება.

თუ მდგომარეობის დამახსოვრების მართვა და პროგრამაზე დაბრუნების მართვა დაკისრებული აქვს თვით მმართველ პროგრამას, მაშინ ის უნდა შედგებოდეს სამი ნაწილისაგან: მომზადებითი და დამამთავრებელი ნაწილებისგან, რომლებიც უზრუნველყოფენ პროგრამების გადართვას, ხოლო მესამე ნაწილი წარმოადგენს საკუთრივ შემწყვეტ პროგრამას, რომელიც ასრულებს მიმართვის მიერ მოთხოვნილ სამუშაოს. ნახ.5.12-ზე მოყვანილია წყვეტის პროცესის გამარტივებული დროითი დიაგრამა.

წყვეტის შეწყვეტის ეფექტურობის შესაფასებლად გამოიყენება შემდეგი მახასიათებლები: რეაქციის დრო, წყვეტის სიღრმე (წყვეტის დონეების რაოდენობა), წყვეტის სისტემის გაჯერება (რიგისობის არსებობა), პროგრამების წყვეტის დასაშვები მომენტები.

რეაქციის დრო - ეს არის დრო წვეტის მიმართვის გამოჩენასა და შემწყვეტი პროგრამის შესრულების დასაწყისის შორის. ნახ.5.12-ზე ეს დრო აღნიშნულია T_r -ით.



ნახ.5.12. წვეტის პროცესის გამარტივებული დროითი დიაგრამა: T_r - არის რეაქციის დრო, $T_{აღ}$ - შეწყვეტილი პროგრამის მდგომარეობის დამახსოვრების დრო, $T_{შეგ}$ - შემწყვეტი პროგრამის შესრულების დრო, $T_{აღ}$ - შეწყვეტილი პროგრამის მდგომარეობის აღდგენის დრო. ერთი და იგივე მიმართვის დაყოვნება შემწყვეტი პროგრამის შესრულებაში დამოკიდებულია იმაზე, თუ რამდენი პროგრამა უფროსი პრიორიტეტით ელოდება მომსახურებას. ამიტომ რეაქციის დროს განსაზღვრავენ უმაღლესი დროების ჯამი, რომელიც იხარჯება შეწყვეტილი პროგრამის დამახსოვრებაზე ($T_{დაგ}$) და შემდეგ მასზე დაბრუნებაზე ($T_{დაბ}$), იწოდება მომსახურების დროდ ($T_{მოგ}$):

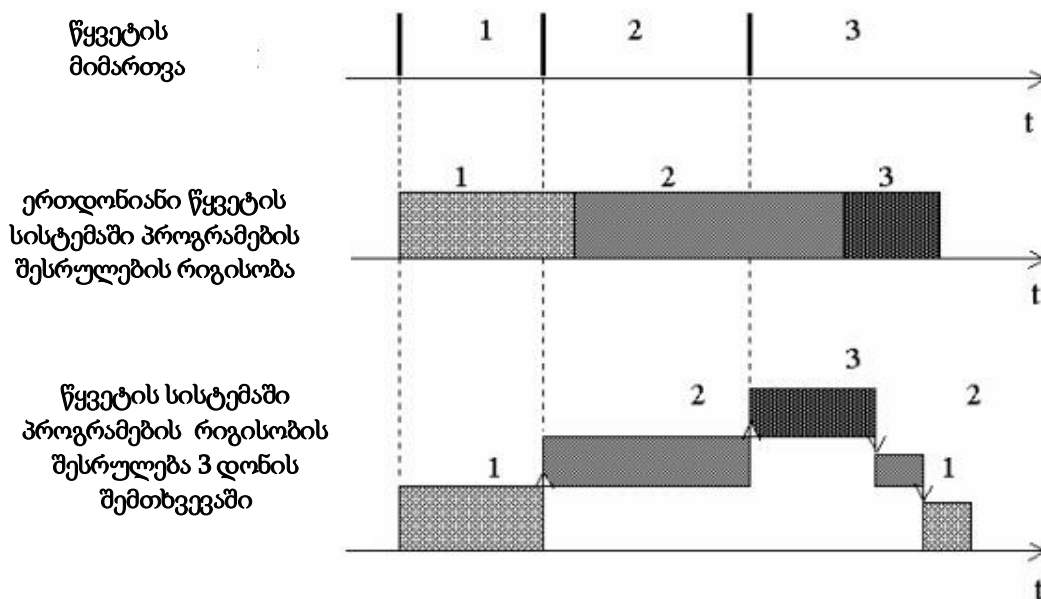
$$T_{მოგ} = T_{დაგ} + T_{დაბ}.$$

ნახ.5.12-ზე გამოსახულია წვეტის პროცესის გამარტივებულ დროითი დიაგრამა.

წყვეტის სიღრმე (დონეების რაოდენობა) - ეს არის პროგრამების მაქსიმალური რიცხვი, რომლებსაც შეუძლიათ შეწყვიტონ ერთმანეთი. თუ შემწყვეტ პროგრამაზე გადასვლის შემდეგ და მის დამთავრებამდეც სხვა მიმართვების მიღება აკრძალულია, მაშინ ამბობენ რომ სისტემას გააჩნია წყვეტის ერთი დონე. სიღრმე ტოლია n - ის, თუ დაშვებულია თანმიმდევრობითი წყვეტა n პროგრამებამდე.

ნახ.5.13-ზე ნაჩვენებია წყვეტა წყვეტის მრავალდონიან სისტემებში (ითვლება, რომ თითოეული შემდეგი მიმართვის პრიორიტეტი მაღალია წინაზე). წყვეტის სიღრმის დიდი მნიშვნელობების მქონე სისტემები უზრუნველყოფენ სასწრაფო მიმართვებზე სწრაფ რეაქციას. წყვეტის ერთნაირ დონეს ჩვეულებრივ მიაკუთვნებენ არა ერთ, არამედ რამდენიმე მიმართვას, რომლებიც ახლოს არიან მოვლენების მნიშვნელოვნობის ხარისხის მიხედვით.

ერთი დონის (კლასის) წყვეტის პროგრამების შესრულება ხორციელდება, როგორც ერთდონიან წყვეტის სისტემაში, ე.ი. რიგის მიხედვით.



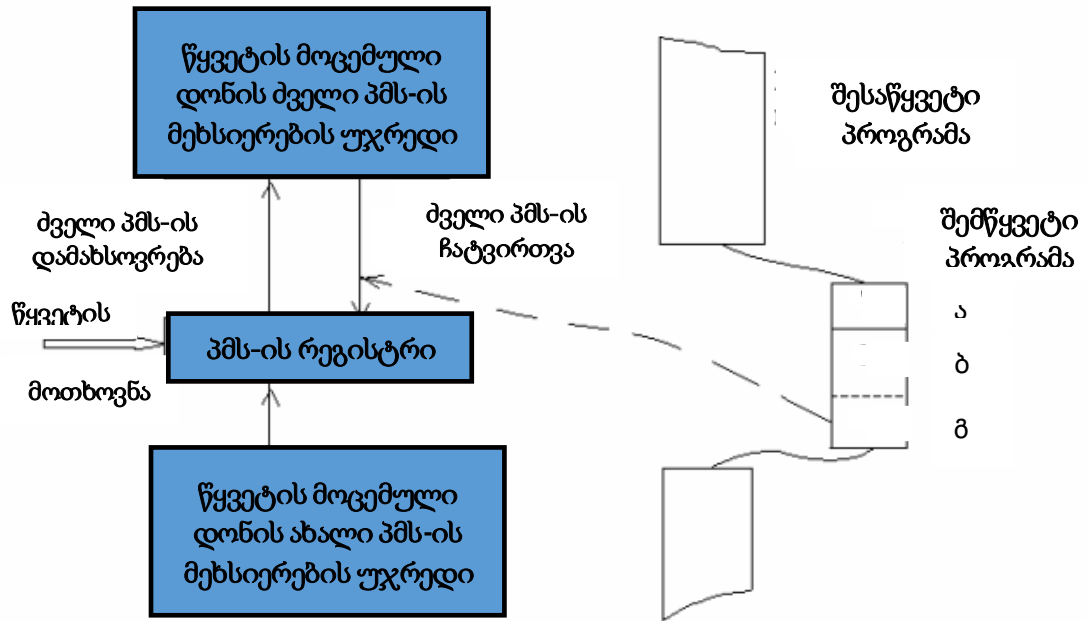
ნახ.5.13. წყვეტა სისტემებში წყვეტის სხვადასხვა სიღრმის დროს

წყვეტის სისტემის გაჯერება წარმოიშობა მაშინ, როცა რეაქციის დრო იმდენად მაღალია, რომ მიმდინარე მიმართვა აღმოჩნდება მოუმსახურებელი ახალი მიმართვის მოსვლის მომენტისათვის იმავე წყაროდან. ასეთ შემთხვევაში წყვეტის ერთი მიმართვა

მოცემული წყაროდან სიტემის მიერ იქნება დაკარგული, რაც დაუშვებელია. საბორტო კომპიუტერის სწრაფქმედება, წყვეტის სისტემის ლოგიკური შესაძლებლობები და წყვეტის წყაროები ისეთნაირად უნდა იყოს შეთანხმებული, რომ ადგილი არ უნდა ჰქონდეს გაჯერებას.

პროგრამების წყვეტის დასაშვები მომენტები. დიდი ნაწილით წყვეტა დასაშვებია ნებისმიერი მიმდინარე ბრძანების დასრულების შემდეგ. ამ შემთხვევაში აუცილებელია მოხდეს პროგრამის მდგომარეობის დამახსოვრება, ე.ი. ყველა პროგრამულად დაშვებადი რეგისტრების შეგთავსის დამახსოვრება, რადგან შესაძლებელ იყოს შეიცვალონ ისინი შემწყვეტი პროგრამით.

რეაქციის დრო განისაზღვრება ძირითადად ერთი ბრძანების შესრულების ხანგრძლივობით. რეაქციის ეს დრო შეიძლება აღმოჩნდეს დაუშვებლად დიდი საბორტო კომპიუტერისთვის, იმდენად რამდენადაც ისინი მუშაობენ დროის რეალურ მასშტაბში. ამიტომ ხანდახან დასაშვებია წყვეტა ბრძანების შესრულების ნებისმიერი ტაქტის ბოლოს. თუმცა ასეთ შემთხვევაში იზრდება ინფორმაციის რაოდენობა, რომელიც უნდა დაეკუმულებაროს დამახსოვრებას და აღდგენას პროგრამების გადართვების დროს. შეწყვეტილი პროგრამის მდგომარეობის დამახსოვრებასთან ერთად, აუცილებელია ასევე შენარჩუნებულ იქნეს პროცესორის მდგომარეობაც წყვეტის მომენტებში (ბრძანებების მთვლელის, ტაქტების მთვლელის, ბრძანებების რეგისტრის და ა.შ. შიგთავსები). ამიტომ წყვეტის ასეთი ორგანიზაცია შესაძლებელია მხოლოდ საბორტო კომპიუტერში ზეოპერატიული მეხსიერებით.



ნახ.5.14. შემწყვეტ პროგრამაზე გადასვლის პროცედურა და მისგან გამოსვლა: ა-რეგისტრების შიგთავსების დამახსოვრება; ბ-შემწყვეტი პროგრამა; გ-რეგისტრების შიგთავსების აღდგენა

პროგრამების წვეტაში პრიორიტეტის ცნებას გააჩნია ორი მნიშვნელობა: პრიორიტეტი წვეტის მიმართებს შორის და პრიორიტეტი შემწყვეტ პროგრამებს შორის.

პირველ შემთხვევაში ხდება ერთდროულად შემოსული მიმართვების აღქმის რიგისობის დადგენა სხვადასხვა დონეებიდან, ხოლო მეორე შემთხვევაში ხდება სხვადასხვა დონის შემწყვეტი პროგრამების შესრულებაში უფრო სწრაფად მომსახურების ხარისხის დადგენა. სხვა სიტყვებით რომ ვთქვათ, მეორე შემთხვევაში განისაზღვრება, გააჩნია თუ არა მოცემულ შემწყვეტ პროგრამას, შეწყვიტოს დამუშავებაში არსებული პროგრამა მოცემულ მომენტში.

K მიმართვებს შორის პრიორიტეტი (დიდი პრიორიტეტი გააჩნია დონეებს მცირე ნომრებით) შეიძლება განხორციელდეს აპარატულად, მაგალითად, მიმდევრობითი ძიების სქემების საშუალებით ან პროგრამულად. პროგრამების მნიშვნელოვნობის ფარდობითი ხარისხი, მათი სიგრძე და გამეორების სიხშირე გამოთვლითი პროცესის მიმდინარეობისას შესაძლებელია იცვლებოდეს, მოითხოვდეს ახალი პრიორიტეტული დამოკიდებულებების დადგენას. ამიტომ მიმართვებს შორის პრიორიტეტი შესაძლებელია დადაგენილ იქნეს ხისტად, ამისგან განსხვავებით შემწყვეტ პროგრამებს

შორის პრიორიტეტი არ შეიძლება დაფიქსირდეს ერთხელ და სამუდამოდ. აუცილებელია არსებობდეს შესაძლებლობები შეიცვალოს საჭიროების შეთხვევაში პრიორიტეტული თანაფარდობები პროგრამული გზით, სხვა სიტყვებით, რომ ვთქვათ, შემწყვეტ პროგრამებს შორის პრიორიტეტი უნდა იყოს პროგრამულად მართვადი. თანამედროვე საბორტო კომპიუტერებში ყველაზე ფართო გავრცელება ჰპოვა პრიორიტეტის პროგრამულმა მართვამ წყვეტის ნილბის საფუძველზე.

წყვეტის ნილაბი წარმოადგენს ორობით კოდს, რომლის ზომები დასმულია წყვეტის დონეების ან წყაროების (მიზეზების) შესაბამისად. ნილაბი ყენდება სპეციალური ბრძანებით ნილბების რეგისტრში. ნილბების რეგისტრის ამ თანრიგში მდგომარეობა „1“ ნებას რთავს, ხოლო მდგომარეობა „0“ კრძალავს („ნილბავს“) მიმდინარე პროგრამის წყვეტას წყვეტის შესაბამისი დონის მიმართვებისგან. ნილბების რეგისტრში თანრიგების განლაგების წესრიგს და დონეების ნუმერაციას არ აქვს მნიშვნელობა. ამრიგად, თუ პროგრამული ხერხით შეიცვლება ნილბების რეგისტრის შიგთავსი, მაშინ შესაძლებელია დადგინდეს ნებისმიერი პრიორიტეტული თანაფარდობები ნებისმიერ ნომრიან დონეებს შორის წყვეტის მიმართვების გამტარების გადაკომუტირების (გადართვების) გარეშე.

შენილბული მიმართვით მოსვლა ხორციელდება წყვეტის მიზეზით, ან მისი იგნორირებით, ან დამახსოვრებით. მაგალითად, თუ წყვეტა გამოწვეულია პერიფერიულ მოწყობილობაში ოპერაციის დამთავრებით, მაშინ საჭიროა ის დამახსოვრებულ იქნეს, იმისათვის რომ საბორტო კომპიუტერს შეატყობინოს პერიფერიული მოწყობილობის განთავისუფლების შესახებ. თუ შენილბული წყვეტა გამოწვეულია თანრიგთა ბადის გადავსებით არითმეტიკული ოპერაციის დროს, მაგალითად მუდმივი მეხსიერების ყველა უჯრედების საკონტროლო ჯამის მიღების შემთხვევაში, მაშინ შესაძლებელია ის იგნორირებულ იქნეს.

ყველა შესასწავლ საბორტო კომპიუტერებს გააჩნიათ წყვეტის მრავალდონიანი სისტემები (A-15A -4 დონე, C100 -7 დონე, ორობიტა10 - 2დონე) და აქვთ შესაძლებლობა შესასრულებელი პროგრამების პრიორიტეტები შეცვალონ პროგრამულად შენილბვის საშუალებით.

წყვეტის შესრულებისათვის წყვეტის სისტემამ უნდა უზრუნველყოს შესაწყვეტი პროგრამის დამახსოვრება, ორგანიზაცია გაუკეთოს შემწყვეტ პროგრამაზე გადასვლას და შეწყვეტილ პროგრამაზე დაბრუნებას.

შემწყვეტ პროგრამების საწყისი მისამართების მითითების უმარტივესი ხერხი მდგომარეობს იმაში, რომ თითოეულ დონეს მეხსიერებაში გამოეყოფა ფიქსირებული უჯრედი, რომელშიც ინახება ინფორმაცია შემწყვეტი პროგრამის საწყისი მისამართის შესახებ.

პროგრამის წვეტისას ინფორმაცია, რომელიც უნდა დაეკავშირებოდეს დამახსოვრებას, შეიძლება დაიყოს ორ ნაწილად: პირველი - ძირითადი ინფორმაცია, რომელიც დაიმახსოვრება ყოველთვის (შეწყვეტილი პროგრამის მომდევნო შეუსრულებელი ბრძანების მისამართი, მისი წვეტის ნილაბი და სხვა); მეორე - დამატებითი ინფორმაცია, რომელიც შეესაბამება იმ რეგისტრების შიგთავსს, რომლებსაც მოცემული შემწყვეტი პროგრამა იყენებს.

პროგრამის მდგომარეობის შესახებ ძირითადი ინფორმაცია (მომდევნო ბრძანების ბრძანების მისამართი, მმართველი ტრიგერების მდგომარეობები, წვეტის ნილაბი და სხვა.) ფორმირდება ეგრეთწოდებულ პროგრამის მდგომარეობის სიტყვად (პმს), რომელიც იკავებს ერთ ან რამდენიმე მანქანურ სიტყვას. პროცედურის ილუსტრაცია შემწყვეტ პროგრამაზე გადასვლაზე და იქიდან სისტემაში დაბრუნებაზე პმს-ის გამოყენებით წარმოდგენილია ნახ.5.14-ზე. თითოეულ წვეტის დონეს გამოეყოფა ოპერაციული დამხსომებელი მოწყობილობის ორი ფიქსირებული უჯრედი „ძველი პმს“ და „ახალი პმს“-ის შენახვისათვის. თვითონ პროგრამის წვეტა ხორციელდება ძველი პმს შეცვლით ახალი პმს-ით, რომელიც მიეკუთვნება შემწყვეტ პროგრამას. მიმდინარე პროგრამის შესრულებისას მისი პმს მოთავსებულია მიმდინარე პმს-ის რეგისტრში და მონაწილეობს გამოთვლითი პროცესის მართვაში. ამასთან პმს-ის ცალკეული ველები (მაგალითად, ბრძანებების მთვლელი) საჭირო სახით იცვლებიან. ახალ პმს-ის უჯრედებში წვეტის ყველა დონეებისთვის ინახება პმს-ები, რომლებიც შეიცავენ ინფორმაციას - დაშვებითს შესაბამისი შემწყვეტი პროგრამების ფუნქციონირების დაწყებისათვის.

წვეტის მოცემული დონიდან მიმართვის შემოსავლისას, თუ ეს მიმართვა არაა შენიღბული წვეტის ნილაბში, აპარატული საშუალებები ასრულებენ შემდეგ პროცედურას. პროგრამის მდგომარეობის სიტყვის რეგისტრის თანრიგების შესაბამის ჯგუფში ჩაიწერება წვეტის კოდი, რომელიც შეიცავს ინფორმაციას წვეტის

კონკრეტული მიზეზის შესახებ. შემდეგ პროგრამის მდგომარეობის სიტყვის რეგისტრიდან წყვეტის მოცემული დონისათვის პროგრამის სიტყვის მდგომარეობა გადაიცემა ოპერატიული დამხმომებელი მოწყობილობის უჯრედში ძველი პმს-ისთვის, ხოლო ამ დონის ახალი პმს-ისთვის (შემწყვეტი პროგრამის პმს, რომელიც შეიცავს მის საწყისს მისამართს) ჩაიტვირთება პმს-ის რეგისტრში. ამ მომენტიდან მართვა გადადის შემწყვეტ პროგრამაზე. პროგრამის მდგომარეობის სიტყვის შეცვლის პერიოდში წყვეტა ნებისმიერი დონიდან აკრძალულია.

შემწყვეტი პროგრამის შესრულება იწყება მეხსიერების უჯრედებში დამახსოვრებული ბრძანებებიდან პროგრამის მდგომარეობის შესახებ. შემდეგ სრულდება საკუთრივ შემწყვეტი პროგრამა. ეს ნაწილი იწყება წყვეტის კოდის ანალიზით, წყვეტის კონკრეტული მიზეზის განსაზღვრით და ქვეპროგრამის ინიცირებით, რომელიც შეესაბამება ამ მიზეზს. შემწყვეტი პროგრამის დამასრულებელი ნაწილი აღადგენს ოპერატიულ დამხმომებელ მოწყობილობაში შემწყვეტ პროგრამის მიერ გამოყენებული რეგისტრების შიგთავსს და ამთავრებს თავის მუშაობას შეწყვეტილი პროგრამის მდგომარეობის სიტყვის ჩატვირთვის ბრძანებით ძველი პმს-ის უჯრედიდან პმს-ის რეგისტრში. მართვა გადადის შეწყვეტილ პროგრამაზე.

მიგ - 31 თვითმფრინავის საბორტო ციფრული გამოთვლითი სისტემა B5 შეიცავს საბორტო კომპიუტერს „არგონი“ A-15A და შემტან-გამომტან მოწყობილობას B5.01. შემტან-გამომტანი მოწყობილობის ძირითადი ამოცანაა - უზრუნველყოს სიგნალების ანალოგური ფორმიდან ციფრულში გარდაქმნა და პირიქით ციფრულიდან ანალოგურში. ასეთი გარდაქმნები ხორციელდება ანალოგურ-ციფრული და ციფრულ-ანალოგური გარდამქმნელების საშუალებით.

ლექცია 6

საბორტო კომპიუტერების მეხსიერების მოწყობილობები

6.1. დამხსომეხელი მოწყობილობების დანიშნულება, ძირითადი მახასიათებლები და კლასიფიკაცია

დამხსომეხელი მოწყობილობა (დმ)-ეს არის კომპიუტერის ნაწილი, რომლის დანიშნულებაც კოდების სახით წარმოდგენილი ინფორმაციის (პროგრამის ბრძანებების, საწყისი მონაცემების, საშუალო და საბოლოო შედეგების) ჩაწერა, შენახვა და გაცემა.

დამხსომეხელი მოწყობილობის ძირითად მახასიათებლებს წარმოადგენს ტევადობა და სწრაფქმედება.

ტევადობის ცნების ქვეშ იგულისხმება კოდების (მანქანური სიტყვების) მაქსიმალური რაოდენობა, რომელთა ერთდროული შენახვა შესაძლებელია დამხსომეხელ მოწყობილობაში. თითოეული კოდის განლაგება ხდება მეხსიერების ერთ უჯრედში. თუ მეხსიერების ტევადობას შევადგენთ ბიტებში, მაშინ ის ტოლია მეხსიერების უჯრედების რიცხვის ნამრავლისა მის სიგრძეზე (თანრიგობა):

$$Q = N * n,$$

სადაც Q -ტევადობაა ბიტებში; N -უჯრედების რაოდენობა; n -უჯრედების თანრიგობა.

სწრაფქმედება ხასიათდება დამხსომეხელ მოწყობილობასთან მიმართვის დროით, რომელიც აუცილებელია ინფორმაციის ჩაწერისა და ამოკითხვისთვის.

ინფორმაციის ამოკითხვის შემთხვევაში მიმართვის დრო წარმოდგება შესაბამისი უჯრედის მოძებნისათვის საჭირო დროის $t_{\text{მოძ}}$, ამოკითხვის დროის $t_{\text{ამო}}$ და აღდგენის დროის $t_{\text{აღდ}}$ (აუცილებლობის შემთხვევაში) შეჯამებით:

$$T_{\text{ამ}} = t_{\text{მოძ}} + t_{\text{ამო}} + t_{\text{აღდ}}.$$

მეხსიერების უჯრედში ჩაწერისას მიმართვის დრო განისაზღვრება მოძიების $t_{\text{მოძ}}$, ადრე ჩაწერილი ინფორმაციის წაშლის $t_{\text{წაშ}}$ და ჩაწერის დროის $t_{\text{ჩაწ}}$ -ის შეჯამებით:

$$T_{\text{ჩაწ}} = t_{\text{მოძ}} + t_{\text{წაშ}} + t_{\text{ჩაწ}}.$$

პრაქტიკულად $T_{\text{აგ}} \approx T_{\text{ჩაწ.}}$

გარდა განხილული ძირითადი მახასიათებლებისა დამხსომებელი მოწყობილობის ხარისხის მნიშვნელოვან მაჩვენებლებს წარმოადგენს:

საიმედოობა - დამხსომებელი მოწყობილობის უნარი შეინარჩუნოს თავისი პარამეტრები დასაშვებ საზღვრებში ექსპლუატაციის სხვადასხვა პირობების დროს.

ეკონომიურობა - დამხსომებელი მოწყობილობის ღირებულება, მისი მეხსიერების ზომასთან მიმართებაში.

გაბარიტები და მოთხოვნილი სიმძლავრე.

ერთდროულად დამხსომებელი მოწყობილობის ტევადობის გაზრდა, მისი ღირებულების შემცირება და სწრაფქმედების ამაღლება შეუძლებელია, იმდენად რამდენადაც, ეს მახასიათებლები ერთმანეთთან დაკავშირებული არიან პირდაპირი ან უკუ დამოკიდებულებით. მართლაც, რაც უფრო დიდია მეხსიერების მოცულობა, მაშინ, ბუნებრივია უფრო მეტი დროა საჭირო მონაცემების მოძებნისათვის. მოთხოვნები მეხსიერების ტევადობისა და სწრაფქმედების მიმართ თავისი არსით ურთიერთ საწინააღმდეგოა. რაც უფრო მაღალია (დიდია) სწრაფქმედება, მით უფრო ძნელი მისაღწევია ტექნიკურად და ძვირია ღირებულების მიხედვით მეხსიერება. მეხსიერების ღირებულება წარმოადგენს საბორტო კომპიუტერის საერთო ღირებულების მნიშვნელოვან ნაწილს. ამიტომ მეხსიერების ორგანიზება ხდება დამხსომებელი მოწყობილობის იერარქიული სტრუქტურის სახით, რომლებიც ერთმანეთისგან განსხვავდებიან თავიანთი სწრაფქმედებითა და ტევადობით.

ყველა დამხსომებელი მოწყობილობა თავისი ფუნქციური დანიშნულებით შესაძლებელია დაიყოს შემდეგ ჯგუფებად:

- **ზეოპერატიული დამხსომებელი მოწყობილობა (ზოდმ)**, რომელიც წარმოადგენს რეგისტრების ნაკრებს, რომელთა შიგთავსი უშუალოდ გამოიყენება პროცესორში ინფორმაციის დამუშავების დროს;

- **ოპერატიული დამხსომებელი მოწყობილობა (ოდმ)**, რომელიც გამოიყენება ოპერატიული ინფორმაციის (ოპერანდები, პროგრამის ნაწილი, რომელიც მოითხოვება მუშაობის პროცესში) შენახვისათვის.

მუდმივი დამხსომებელი მოწყობილობა (მდმ), რომლის დანიშნულებაც საბორტო კომპიუტერის მუშაობის პროცესში ინფორმაციის (პროგრამების, მიკროპროგრამების, კონსტანტების) ხანგრძლივი შენახვა;

ნახევრად მუდმივი (რეპროგრამირებადი) დამხსომებელი მოწყობილობა, რომელიც ფუნქციონირებს შესრულების თვალსაზრისით სრულად შეესაბამება მუდმივ დამხსომებელ მოწყობილობას, თუმცა უკანასკნელისგან განსხვავდება იმით, რომ მასში შესაძლებელია შედარებით სწრაფად მოხდეს მასში შენახული მთელი ინფორმაციის ერთდროული შეცვლა.

– **გარე დამხსომებელი მოწყობილობა (გდმ)**, რომლის დანიშნულებაც დიდი მოცულობის ინფორმაციის შენახვა შენახული ინფორმაციის ბიტის მცირე ხვედრითი ღირებულებით. გარე დამხსომებელი მოწყობილობა იმყოფება საბორტო კომპიუტერის კორპუსის გარეთ. ამჟამად გამანადგურებლის ბორტზე გარე დამხსომებელი მოწყობილობების განლაგება, რომლებიც შესრულებულია მაგნიტურ დისკებზე, მაგნიტურ ლენტებზე გამოუყენებადია ფუნქციონირების გაძნელებული პირობების გამო. ამიტომ გარე დამხსომებელი მოწყობილობები ჩვეულებრივ გამოიყენება მიწისზედა ოპერაციების შესასრულებლად, რომლებიც დაკავშირებულია საბორტო კომპიუტერის და მთლიანად მთელი რადიოელექტრონული კომპლექსის მუშაობის უნარიანობისა და დიაგნოსტიკის კონტროლთან.

დამხსომებელი მოწყობილობები შესაძლებელია რეალიზებულ იქნეს მხოლოდ ოპერატიული დამხსომებელი მოწყობილობის დიდ ინტეგრალურ მიკროსქემებზე. თუმცა მათი გამოყენებისას კვების ძაბვის გამორთვისას ადგილი აქვს ინფორმაციის დაკარგვას, ასევე საკმაოდ მაღალია შესაძლებელი ინფორმაციის ბიტის ხვედრითი (საკუთარი) ღირებულება, ამან და ასევე სხვა მიზეზებმა გამოიწვია მუდმივი და ნახევრად მუდმივი დამხსომებელი მოწყობილობების ფართო გამოყენება.

პროცესორებისა და ცალკეულ დამხსომებელ მოწყობილობებს შორის ინფორმაციის გაცვლის ორგანიზაციის ხერხის მიხედვით განასხვავებენ შემდეგ დამხსომებელ მოწყობილობებს:

1. დამხსომებელი მოწყობილობა თავისუფალი ამორჩევით;
2. დამხსომებელი მოწყობილობა თანმიმდევრობითი მიმართვით;

3. დამხსომებელი მოწყობილობა გაცვლის თანმიმდევრობით-პარალელური ორგანიზაციით.

საბორტი კომპიუტერის დამხსომებელ მოწყობილობათა უმრავლესი ნაწილი - ეს არის დამხსომებელი მოწყობილობა თავისუფალი ამორჩევით (ან თავისუფალი წვდომით, დაშვებით). თუ მეხსიერება ვერ უზრუნველყოფს მის ნებისმიერ ნაწილთან დაშვებას (მიმართვას), მაშინ ასეთი მეხსიერება წარმოადგენს მეხსიერებას თანმიმდევრობითი დაშვებით და ის ძალზე იშვიათად გამოიყენება საბორტო კომპიუტერის ძირითადი მეხსიერების სახით. ასეთი მეხსიერების მუშაობისას მეხსიერების საჭირო არესთან მიმართვისათვის აუცილებელია წინასწარ მოხდეს მიმართვა ყველა იმ არესთან, რომელიც მდებარეობს იმ არესა (რომელთანაც მოხდა მიმართვა მოცემულ მომენტში) და მოთხოვნილ არეს შორის. ინფორმაცია წარმოდგენილი თანმიმდევრობით ფორმაში ინახება მაგნიტურ ლენტზე. ინფორმაციასთან თანმიმდევრობითი დაშვების ხერხი გამოიყენება დიდი მოცულობის მონაცემების დამახსოვრებისთვის, რომელთათვისაც მიმართვის დრო არ წარმოადგენს კრიტიკულს.

დამხსომებელი მოწყობილობის ერთ-ერთი მახასიათებელია მისი უნარი-შეინარჩუნოს ინფორმაცია კვების წყაროს გამორთვის შემთხვევაში. ასეთ შემთხვევაში განასხვავებენ ენერგოდამოკიდებულ და ენერგოდამოუკიდებელ მეხსიერებას. ენერგოდამოუკიდებელ მეხსიერებაში კვების სისტემის მუშაობაში წარმოშობილი დარღვევების შედეგად მონაცემები არ იკარგება, ხოლო ენერგოდამოკიდებულ მეხსიერებაში კი ასეთ შემთხვევაში მონაცემები იკარგება.

თანამედროვე საბორტო კომპიუტერებისთვის დამახასიათებელია ოპერატიული დამხსომებელი მოწყობილობების გამოყენება, რომლებიც აგებულია უნიპოლარულ ველიან ტრანზისტორებზე.

ზოგიერთ მიკროპროცესორულ საბორტო კომპიუტერებში გამოყენებას პოულობს ასევე ორი სახის ნახევარგამტარული მეხსიერება:

1. მეხსიერება, რომელიც შესრულებულია ტევადური კავშირის მქონე ხელსაწყოებზე;
2. მეხსიერება, რომელიც შესრულებულია ცილინდრულ მაგნიტურ დომენებზე.

მეხსიერების ეს ორივე ტიპი ორიენტირებულია თანმიმდევრობით წვდომაზე. მათ, ისე როგორც სხვა თანმიმდევრობითი დაშვების მოწყობილობებს გააჩნიათ დაბალი სწრაფქმედება, თუმცა საშუალებას იძლევიან აგებულ იქნეს საკმაოდ დიდი მოცულობის დამხსომებელი მოწყობილობები, ვიდრე ეს შესაძლებელია ბიპოლარულ და უნიპოლარულ ტრანზისტორებზე აგებული დამხსომებელი მოწყობილობების შემთხვევაში.

ტევადური კავშირის ხელსაწყოებზე შექმნილი ნახევარგამტარული მეხსიერების ღირსებას წარმოადგენს მის მიერ ძალზე მცირე მოთხოვნილი სიმძლავრე და სიმარტივე გამოყენებაში. ასეთ ხელსაწყოებზე აგებული მეხსიერება ენერგოდამოკიდებულია, თუმცა მცირე მოთხოვნილი სიმძლავრე შესაძლებელს ხდის მასში ინფორმაცია შენახული იქნეს სათადარიგო აკუმულატორების დახმარებით, თუ მოხდება ძირითადი კვების წყაროს მწყობრიდან გამოსვლა ხანგრძლივი დროის განმავლობაში.

მეხსიერება, რომელიც აგებულია ცილინდრულ მაგნიტურ დომენებზე ერთის მხრივ მოითხოვს დამხმარე სქემების უფრო დიდ რიცხვს, ვიდრე მეხსიერება აგებული ტევადური კავშირის ხელსაწყოებზე, მაგრამ მეორეს მხრივ, ის წარმოადგენს ენერგოდამოუკიდებელს.

ამრიგად, მეხსიერების ეს ტიპი საშუალებას იძლევა აიგოს ენერგოდამოუკიდებელი დამხსომებელი მოწყობილობები დიდი მოცულობით და რომელთათვისაც არაა საჭირო რთული მექანიკური ლენტგამწევი მექანიზმები.

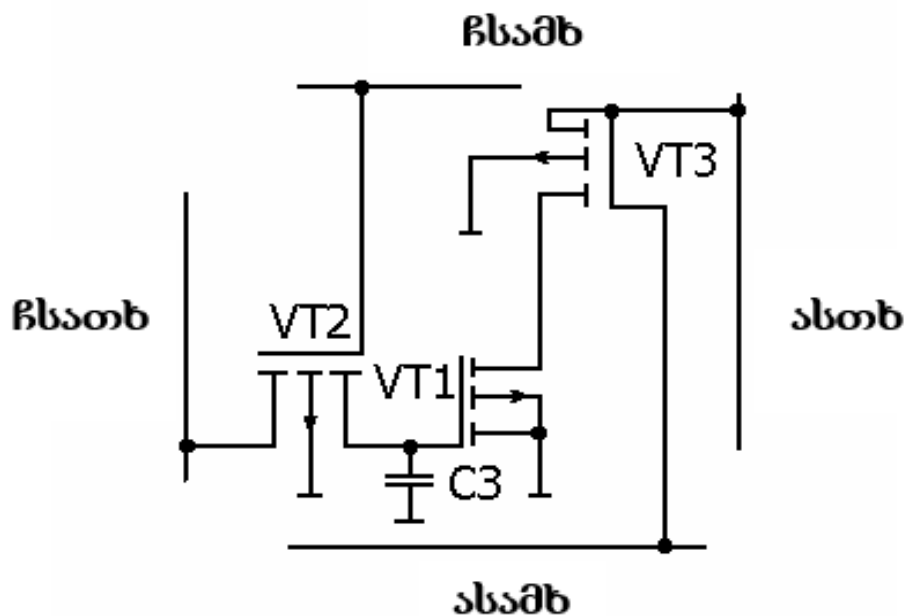
6.2. ოპერატიული დამხსომებელი მოწყობილობები

ოპერატიული მეხსიერება მიეუთვნება მეხსიერებას პირდაპირი დაშვებით (Random access memory-RAM). ინფორმაციის შენახვის პრინციპის მიხედვით ნახევარგამტარული დამხსომებელი მოწყობილობები იყოფა დინამიკურ და სტატიკურ ოპერატიულ დამხსომებელ მოწყობილობებად.

6.2.1. დინამიკური დამხსომებელი მოწყობილობები

დინამიკური დამხსომებელი მოწყობილობები (DRAM) აიგება ისეთი დამხსომებელი ელემენტის საფუძველზე, რომელიც თავის მდგომარეობას ინარჩუნებს მხოლოდ დროის განსაზღვრულ შუალედში. გამომდინარე აქედან ის მოითხოვს პერიოდულ აღდგენას. დინამიკურ ნახევარგამტარულ დამხსომებელ მოწყობილობაში დამხსომებელ ელემენტს წარმოადგენს კონდენსატორი, რომელშიც ინფორმაცია ინახება მუხტის არსებობის ან არარსებობის ფორმის სახით. დამხსომებელ კონდენსატორზე მუხტი მუდმივად მცირდება გადინების გამო. მუხტის აღსადგენად დამხსომებელი კონდენსატორი პერიოდულად უერთდება კვების წყაროს.

მეხსიერების დინამიკური უჯრედის უმარტივესი სქემა როცა ის შესრულებულია უნიპოლარულ ტრანზისტორებზე ნაჩვენებია ნახ.6.1-ზე.



ნახ.6.1. დინამიკური მეხსიერების უჯრედი

ინფორმაციის შენახვა უზრუნველყოფილია C კონდენსატორის დახმარებით. უჯრედში გამოიყენება ჩაწერის სამისამართო ხაზი (აქსამბ), ამოკითხვის სამისამართო

ხაზი (ასამხ), ჩაწერის სათანრიგო ხაზი (ჩსათხ) და ამოკითხვის სათანრიგო ხაზი (ასათხ).

ინფორმაციის ჩაწერის პროცესში ჩაწერის სათანრიგო ხაზიდან (ჩსათხ), როცა ჩაწერის სამისამართო ხაზზე (ჩსამხ) არსებობს ნებისდამრთველი სიგნალი VT2 გახსნილი ტრანზისტორის საშუალებით უზრუნველყოფილია C ტევადობის მუხტი. ინფორმაციის ამოკითხვა წარმოებს თანრიგობრივი ამოკითხვის ხაზიდან VT1 და VT3 ტრანზისტორების საშუალებით, როცა არსებობს ნებისდამრთველი - ამოკითხვის სმისამართო სიგნალი (ასს).

დინამიკური მეხსიერების უჯრედის რეგენერაცია (მონაცემების აღდგენა) წარმოებს მასთან ყოველი მიმართვის დროს. თუმცა საბორტო კომპიუტერის მუშაობის ჩვეულებრივი ინტენსიურობის დროს ასეთი რეგენერაცია არ იძლევა მასში შენახული ყველა ბიტის შენახვის გარანტიას. მაგალითად საბორტო კომპიუტერს შეუძლია დახარჯოს დრო, რომელიც გადააჭარბებს რამდენიმე მილიწამს მარტივი ციკლის შესრულებისთვის და ამ დროს გამოიყენოს მეხსიერების მხოლოდ რამდენიმე უჯრედი. მთელი ამ დროის შუალედში მეხსიერების ყველა სხვა სიტყვები არ დაექვემდებარებიან რეგენერაციას. ამიტომ დინამიკურ დამხსომებელ მოწყობილობებზე შესრულებული მეხსიერების მოწყობილობები მოითხოვენ რეგენერაციის ლოგიკურ სქემას, რომელიც ავტომატურად ახდენს მიმართვას მეხსიერების თითოეულ სვეტთან მილიწამის რამდენიმე ათეული ნაწილის ინეტერვალებით. დინამიკური დამხსომებელი მოწყობილობა აგებულია ისეთნაირად, რომ თვით მიმართვა სვეტთან უზრუნველყოფს ინფორმაციის რეგენერაციას მის ყველა უჯრედში. რეგენერაციის ლოგიკური სქემის მუშაობა კოორდინირებული უნდა იყოს პროცესორის სხვა მოქმედებებთან. მაგალითად, თუ პროცესორი ცდილობს მიმართოს მეხსიერებას იმ მომენტში, როცა მასში ხორციელდება რეგენერაცია, მაშინ რეგენერაციის სქემამ პრიორიტეტი უნდა მისცეს სახელდობრ პროცესორს.

დინამიკური ნახევარგამტარული დამხსომებელი მოწყობილობები თავისუფალი ამორჩევით გამოირჩევიან ინტეგრაციის მაღალი ხარისხით და დიდი სწრაფქმედებით, დაბალი ღირებულებით. ამიტომ ყოველივე ამის გამო ისინი ფართო გამოყენებას პოულობენ მიკროპროცესორულ სისტემებში. დამხსომებელი მოწყობილობები

აგებული დინამიკურ დიდ ინტეგრალურ სქემებზე, წარმოადგენენ ფუქციურად დასრულებულ მოწყობილობებს და მათგან შესაძლებელია შედარებით მარტივად აიგოს სხვადასხვა ინფორმაციული ტევადობის მქონე ოპერატიული დამხსომეხელი მოწყობილობები.

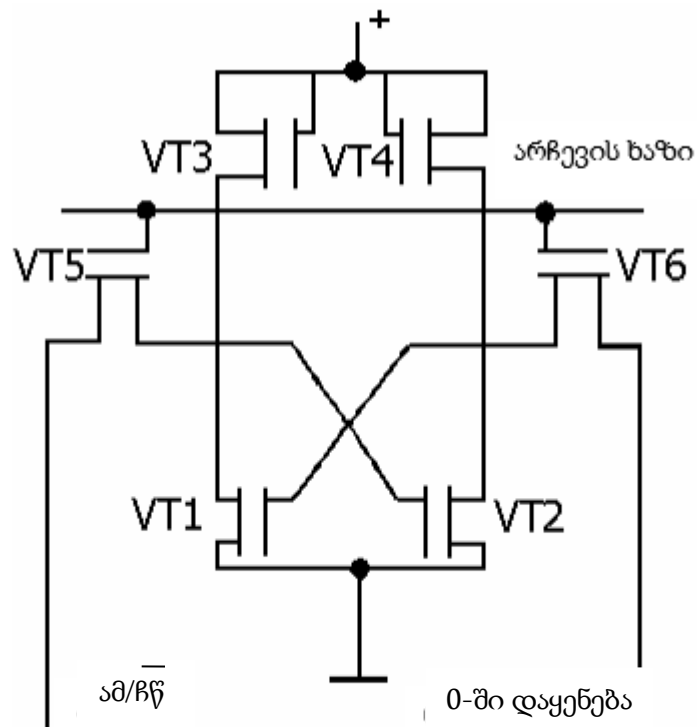
დინამიკური დამხსომეხელი მოწყობილობების ნაკლოვანი მხარე-რეგენერაციის აუცილებლობა კომპენსირდება უფრო დიდი ინფორმაციული ტევადობით ვიდრე ეს შესაძლებელია სტატიკურ დამხსომეხელ მოწყობილობებში.

6.2.2. სტატიკური დამხსომეხელი მოწყობილობები

სტატიკური დამხსომეხელი მოწყობილობები წარმოადგენენ მიკროპროცესორული სისტემების მეხსიერების ყველაზე უფრო გავრცელებულ სახეს. სტატიკური დამხსომეხელი მოწყობილობების უმრავლესობა რეალიზებულია მეტალი - ჟანგეული -ნახევარგამტარი (უნიპოლარული) ტექნოლოგიის საფუძველზე, თუმცა ასევე არსებობენ სტატიკური დამხსომეხელი მოწყობილობები ბიპოლარულ სქემებზე.

სტატიკური დამხსომეხელი მოწყობილობის მეხსიერების უჯრედი თავის მხრივ წარმოადგენს ჩვეულებრივ ტრიგერს. ის შეიძლება იმყოფებოდეს მდგომარეობა „0“-ში, ან მდგომარეობა „1“-ში. თუ ტრიგერი დაყენებულია მდგომარეობა „1“-ში, მაშინ ეს მდგომარეობა შენარჩუნებული იქნება მანამ, სანამ არ მოხდება ტრიგერის ჩამოყრა ან არ მოხდება კვების წყაროს გამორთვა.

თითოეული ტრიგერი ინახავს ინფორმაციის 1 ბიტს. ტრიგერებს გააჩნიათ ცალკეული შესასვლელები მათ დასაყენებლად „0“ ან „1“ მდგომარეობაში, ასევე ვენტილები მათი შეერთებისათვის მონაცემთა სალტის ხაზებთან. თითოეული უჯრედი მეხსიერებისა ჩვეულებრივ შეიცავს ექვს უნიპოლარულ ტრანზისტორს (ნახ.6.2).



ნახ.6.2. სტატიკური მეხსიერების უჯრედი

უჯრედთან მიმართვისას, როგორც მეხსიერების ელემენტთან ამორჩევის ხაზზე ყენდება მაღალი დონის სიგნალი. VT5 და VT6 ტრანზისტორები ჩაირთვება, ხოლო ამოკითხვა-ჩაწერის (ამ/ჩაწ) ხაზი უერთდება ჩამკეტს.

ლოგიკური „1“-ის ჩაწერისათვის ამ ელემენტში ამ/ჩაწ ხაზზე ყენდება მაღალი დონე, ხოლო ხაზზე „დაყენება 0“-ში კი სიგნალის დაბალი დონე. ამასთან VT2 ტრანზისტორი გაიხსნება, ხოლო VT1 ჩაიკეტება. ლოგიკური „0“-ის ჩაწერის შემთხვევაში სიგნალის დონეები ამ/ჩაწ და „დაყენება 0“-ში ხაზებზე იცვლება საპირისპიროდ. დაყენების სიგნალების დამთავრების შემდეგ VT1 და VT2 ტრანზისტორების მდგომარეობა არ იცვლება. ხაზებში ამ/ჩაწ და „დაყენება 0“-ში სიგნალების დონეების კომბინაციით მიიღწევა მეხსიერების უჯრედის სამი მდგომარეობა: ჩაწერა, ამოკითხვა და ინფორმაციის შენახვა.

6.2.3. მეხსიერების უჯრედების მატრიცულ სტრუქტურაში გაერთიანება

მეხსიერების უჯრედები გაერთიანება მატრიცულ სტრუქტურაში ნიშნავს მათ განლაგებას სტრიქონებისა და სვეტების მიხედვით. მეხსიერების თითოეულ ელემენტს ენიჭება მისამართი, რომელიც წარმოადგენს n -თანრიგა კოდს. ორობითი კოდის თანრიგების რაოდენობა განსაზღვრავს დამხსომეხელი მოწყობილობის მატრიცის ტევადობას და განისაზღვრება როგორც

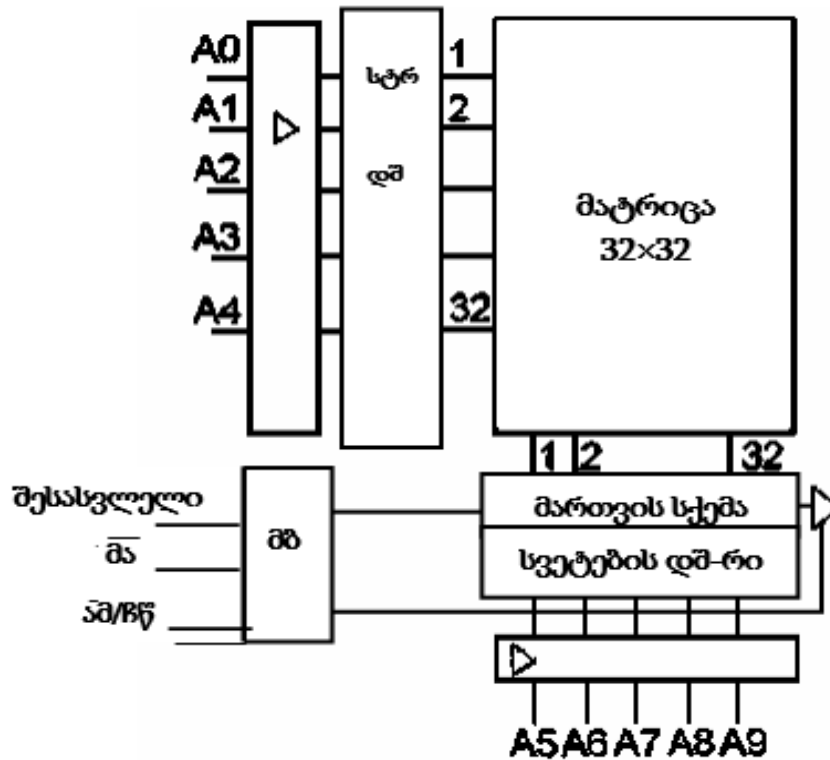
$$Q = 2^n ,$$

სადაც n - სამისამართო კოდის თანრიგობაა. სამისამართო კოდი დაყოფილია ორ ჯგუფად სტრიქონების მისამართების თანრიგები და სვეტების მისამართების თანრიგები. თანრიგების თითოეული ჯგუფი უერთდება შესაბამისი დეშიფრატორების შესასვლელებს, ხოლო მათი გამოსასვლელები უზრუნველყოფენ მეხსიერების უჯრედთან მიმართვას სტრიქონის მისამართისა და სვეტის მისამართის მითითებით.

ნახ.6.3-ზე გამოსახულია ოპერატიული დამხსომეხელი მოწყობილობის მიკროსქემის სტრუქტურა განცალკევებული ერთთანრიგიანი შესასვლელებით და გამოსასვლელებით.

მიკროსქემაში გამოიყენება სამისამართო სიგნალების გამაძლიერებლები და მისამართების დეშიფრატორი $5 \rightarrow 32$, რაც საშუალებას იძლევა მოხდეს მეხსიერების 1024 ერთთანრიგიანი უჯრედების დამისამართება. ოპერატიული დამხსომეხელი მოწყობილობის ინტეგრალურ სქემებთან მიმართვისათვის გამოიყენება მიკროსქემების ამორჩევის (მა) დაბალი დონის სიგნალი და ამ/ჩაწ მართვის სიგნალი. ინფორმაციის ბიტი, რომელიც უნდა დაექვემდებაროს ჩაწერას, მიეწოდება მართვის ბლოკის გამომყვანზე „შესასვლელი“, ხოლო ამოკითხვის დროს სიგნალი „ამოკითხვა“ ამოიკითხება გამოსასვლელი ხაზიდან.

განხილული ინტეგრალური მიკროსქემის სტრუქტურაა $1024 * 1$ ბიტი.



ნახ.6.3. ოდმ-ის ინტეგრალური სქემის სტრუქტურა

ამჟამად მრეწველობა უშვებს მეხსიერების ინტეგრალურ სქემებს ათეული და ასეული მეგაბაიტების ტევადობით. პრაქტიკული გამოყენების თვალსაზრისით უფრო მოსახერხებელია მეხსიერების ის ინტეგრალური სქემები, რომლებიც ჯერადია ან ტოლია პროცესორის თანრიგობასთან.

ასეთ შემთხვევაში მეხსიერების ინტეგრალურ სქემასთან ერთჯერადი მიმართვისას უზრუნველყოფილია გაცვლა მანქანური სიტყვების მიხედვით .

დინამიკური ოპერატიული დამხსომეხელი მოწყობილობის ტიპური დიდი ინტეგრალური სქემებია: KP565PY6-16384*1ბიტი ტევადობით; K565PY5-65535*1 ბიტი ტევადობით.

სტატიკური ოპერატიული დამხსომეხელი მოწყობილობების ტიპური დიდი ინტეგრალური სქემებია:

KP537PY14- 4096*1ბიტი ტევადობით;

KP132PY6A-16384*1ბიტი ტევადობით;

K537PY9- 2048 * 8 ბიტი ტევადობით.

6.3. მუდმივი დამხსომებელი მოწყობილობები

მუდმივი დამხსომებელი მოწყობილობები (ROM, Read Only Memory) - მეხსიერება მხოლოდ ამოკითხვისათვის - ეს არის ენერგოდამოუკიდებელი მეხსიერება, რომელიც გამოიყენება არაცვალებადი მონაცემების შესანახად: ქვეპროგრამების, მიკროპროგრამების, კონსტანტების და ა.შ. ასეთი დამხსომებელი მოწყობილობები მუშაობენ მხოლოდ მრავალჯერადი ამოკითხვის რეჟიმში. ისინი შეიძლება დავყოთ დაპროგრამირების ხერხის მიხედვით შემდეგ კატეგორიებად:

- **ნიღბური მდმ**, რომელთა დაპროგრამება ხდება დამზადების დროს. მდმ-ის მოცემული სახესხვაობა დაპროგრამირდება მხოლოდ ერთხელ და არ უშვებს ინფორმაციის შემდგომ ცვლილებას;

- **დაპროგრამირებადი მუდმივი დამხსომებელი მოწყობილობები (PROM-Programmable ROM)** - ესაა მუდმივი დამხსომებელი მოწყობილობები, რომლებშიც დასაშვებია ერთჯერადი ელექტრული დაპროგრამება. ისინი ნიღბური მუდმივი დამხსომებელი მოწყობილობებისგან განსხვავდებიან იმით, რომ საშუალებას იძლევიან მიკროსქემის გამოყენების შემთხვევაში ერთჯერადად შეცვალონ დამხსომებელი მატრიცის მდგომარეობა ელექტრული გზით მოცემული პროგრამის შესაბამისად;

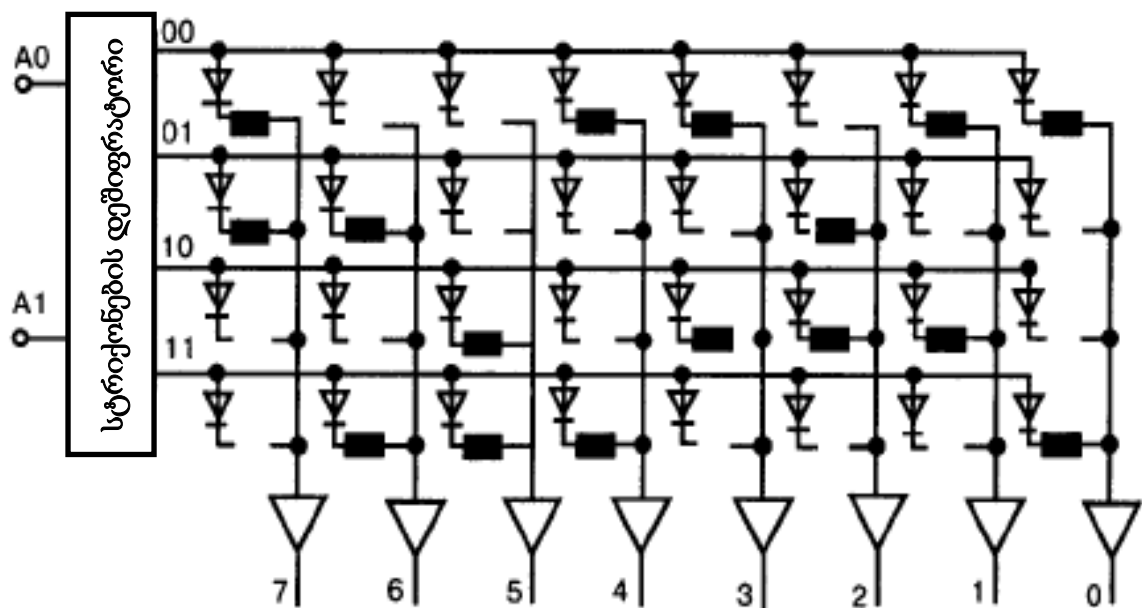
- **რეპროგრამირებადი (გადაპროგრამირებადი) მუდმივი დამხსომებელი მოწყობილობები (გმდმ)** ესენი არიან მუდმივი დამხსომებელი მოწყობილობები, რომლებშიც შესაძლებელია მრავალჯერადი ელექტრული გადაპროგრამება. რეპროგრამირებად მუდმივი დამხსომებელი მოწყობილობაში შენახული ინფორმაციის წაშლა ახალი ინფორმაციის ჩაწერის წინ შესაძლებელია განხორციელდეს სხვადასხვანაირად. ამას ახერხებენ ან ელექტრული სიგნალების დახმარებით ან ულტრაიისფერი გამოსხივების დახმარებით. ამ უკანასკნელ შემთხვევაში აღნიშნული ოპერაციების შესასრულებლად მიკროსქემის კორპუსში გათვალისწინებულია კვარცული მინისგან დამზადებული ფანჯარა.

მუდმივი დამხსომებელი მოწყობილობები განსხვავებით ოპერატიული დამხსომებელი მოწყობილობებისგან არ კარგავენ ინფორმაციას ელექტრული კვების გამორთვის დროს და უზრუნველყოფენ ინფორმაციის მხოლოდ წაკითხვას. მუდმივი დამხსომებელი მოწყობილობების ყველა ინტეგრალური მიკროსქემა შეიძლება დაიყოს

ორ ტიპად: ერთჯერადი დაპროგრამებით და გადაპროგრამებით. პირველ ტიპს მიეკუთვნება ნილბური დამხსომებელი მოწყობილობები, რომლებიც პროგრამდებიან მიკროსქემის დამზადების პროცესში, ხოლო მეორე ტიპის მუდმივი დამხსომებელი მოწყობილობები პროგრამდებიან დნობადი შესაკრავების დახმარებით.

მუდმივი დამხსომებელი მოწყობილობების დამგროვებელს ჩვეულებრივ ასრულებენ ურთიერთპერპენდიკულარული სალტების სისტემის სახით, რომელთა გადაკვეთებში მოთავსებულია ლოგიკური ელემენტი, რომელიც აკავშირებს ორ სალტს და განსაზღვრავს უჯრედში ჩაწერილია ნული თუ ერთიანი. მუდმივი დამხსომებელი მოწყობილობაში სიტყვის ამორჩევა ხორციელდება მისამართის დეშიფრატორის დახმარებით. ნილბურ მუდმივი დამხსომებელი მოწყობილობებში ინფორმაცია მიკროსქემაში შეიტანება დამზადებისას სქემის შრეებიდან ერთ-ერთი შრის აგებისას ფოტომაზლონის დახმარებით, რაც განაპირობებს სწორედ მის დასახელებას.

ნახ.6.4-ზე წარმოდგენილია ნილბური მუდმივი დამხსომებელი მოწყობილობა, რომელშიც ჩაწერა წარმოებს იმ უბნების მეტალიზაციის გზით, რომლებიც აერთებენ სტრიქონებისა და სვეტების შესაბამის ხაზებს დიოდების საშუალებით.



ნახ.6.4. ნილბური მუდმივი დამხსომებელი მოწყობილობა

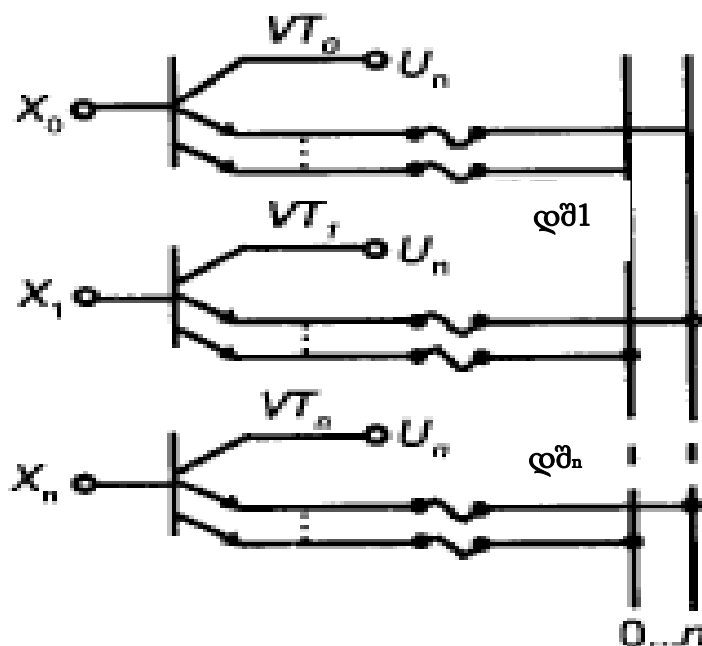
ცხრილში 6.1 წარმოდგენილია მუდმივი დამხსომებელი მოწყობილობა 4 ბაიტი ტევადობით.

მდმ-ის შიგთავსი

ცხრილი 6.1

მისამართი	შესანახი მნიშვნელობა	
00	10011011	9B
01	11000100	C4
10	00101110	2E
11	01110001	71

დიდი გავრცელება ჰპოვეს მუდმივი დამხსომებელი მოწყობილობების მიკროსქემებმა დნობადი შესაკრავების საფუძველზე. ასეთ მიკროსქემებში კავშირის ელემენტს წარმოადგენს ბიპოლარული ტრანზისტორი ამოწვადი შესაკრავით (ნახ.6.5).



ნახ. 6.5. მდმ-ის მეხსიერების ელემენტის სქემა დნობადი შესაკრავებით

ასეთი სქემის გამოყენების შემთხვევაში ნულის ჩაწერისთვის დაპროგრამირების დროს ტრანზისტორის შესაბამის გადასასვლელზე ატარებენ დენის იმპულსს, რომელიც აუცილებელია შესაკრავის გაწყვეტისათვის.

რეპროგრამირებად მუდმივი დამხსომებელი მოწყობილობებს ჩვეულებრივ მიაკუთვნებენ მუდმივი დამხსომებელი მოწყობილობების მიკროსქემებს ელექტრული

დაპროგრამებით და ულტრაიისფერი წაშლით. ასევე მუდმივი დამხსომებელი მოწყობილობებს ელექტრული დაპროგრამებით და ელექტრული წაშლით.

რეპროგრამირებად მუდმივი დამხსომებელი მოწყობილობების ორგანიზაცია განსხვავდება პროგრამირებადი მუდმივი დამხსომებელი მოწყობილობებისგან იმით, რომ სტრიქონებისა და სვეტებს ხაზებს შორის დაყენებულია არა დიოდები დნობადი შესაკრავებით, არმედ სპეციალური ველიანი (საველე) ტრანზისტორები იზოლირებული საკეტით (მეტალი - ჟანგეული - ნახევარგამტარი სტრუქტურით). დამზადების შემდეგ ყველა ველიანი ტრანზისტორი აღჭურვილია ძალიან მაღალი წინაღობით (ე.ი. ჩაკეტილი არიან). დიდი ამპლიტუდის იმპულსის მიწოდებით ველიანი ტრანზისტორი გადადის გამტარ (დია) მდგომარეობაში, რომელიც მას შეუძლია შეინარჩუნოს 10 წლის განმავლობაშიც კი. ველიანი ტრანზისტორების საწყის მდგომარეობაში (ჩაკეტილი მდგომარეობა) დაბრუნებისათვის მათზე უნდა მოხდეს ულტრაიისფერი სხივებით ხანგრძლივი ზემოქმედება. ყველა ველიანი ტრანზისტორის ჯგუფური დასხივება ხდება სპეციალური დანადგარების საშუალებით 10-30 წუთის განმავლობაში გამჭვირვალე (კვარცხული) ფანჯრის საშუალებით დიდი ინტეგრალური მიკროსქემის კორპუსში. ამის შემდეგ რეპროგრამირებადი მუდმივი დამხსომებელი მოწყობილობის დიდი ინტეგრალური სქემა აღმოჩნდება საწყის მდგომარეობაში და შესაძლებელია მოხდეს მისი ხელახლა დაპროგრამება.

მუდმივი დამხსომებელი მოწყობილობაში ელექტრული დაპროგრამებით და ელექტრული წაშლით დაპროგრამების შემდეგ შეიძლება ცალკეულად აღებული ნებისმიერი ველიანი ტრანზისტორი დაბრუნდეს საწყის მდგომარეობაში (ინფორმაციის წაშლა). ასეთი ტიპის გადაპროგრამირებად მუდმივი დამხსომებელი მოწყობილობას მიეკუთვნება ედნერგოდამოუკიდებელი მეხსიერება Flash Memory, რომელიც უშვებს თავისი შიგთავსის მრავალჯერად გადაწერას დისკეტიდან. Flash მეხსიერება მისი კომპაქტურობის სიიაფის, მექანიკური სიმტკიცის, დიდი ტევადობის (მეხსიერების ინფორმაციული ტევადობა აღწევს 1024 მბაიტს), მუშაობის მაღალი სიჩქარის და დაბალი ენერგომომარაგების გამო ფართოდ გამოიყენება ციფრულ პორტატულ მოწყობილობებში და ინფორმაციის მატარებლებში.

6.4. გარე დამხსომებელი მოწყობილობები

გარე დამხსომებელი მოწყობილობები (გდმ) - ეს არის მოწყობილობები დიდი მოცულობით (ასეული მილიონობით ბაიტები და უფრო მეტი) რომელთა დანიშნულებაა მონაცემების ხანგრძლივი შენახვა, მონაცემთა არქივის შექმნა (მათ რიცხვში მოხსნად ინფორმაციულ მატარებლებზეც) და ამ მონაცემების გაცვლა ოპერატიულ დამხსომებელ მოწყობილობებთან. ზოგადად გარე დამხსომებელი მოწყობილობა შედგება დამგროვებისა და კონტროლერისგან.

გარე დამხსომებელი მოწყობილობების დამგროვებლად იწოდება გარე დამხსომებელი მოწყობილობის ის ნაწილი, რომელიც უშუალოდ იღებს ინფორმაციას, გარდაქმნის მას მოხერხებულ ფორმაში, ახორციელებს ამ ინფორმაციის შენახვას და გარდაქმნას ელექტრულ სიგნალებში ინფორმაციის ამოკითხვის დროს. გარე დამხსომებელი მოწყობილობების დამგროვებელი შედგება ინფორმაციის გარდაქმნის მოწყობილობისგან, ამძრავი მექანიზმისა და ინფორმაციის მატარებლისგან.

ინფორმაციის გარდაქმნის მოწყობილობა თავის შემადგენლობაში შეიცავს ჩაწერისა და ამოკითხვის გამაძლიერებლებს, მაგნიტურ თავაკებს (მაგნიტური გარე დამხსომებელი მოწყობილობების შემთხვევაში) და სხვა საშუალებებს, რომლებიც აუცილებელია ციფრული კოდების პირდაპირი და უკუ გარდაქმნებისთვის ისეთ ფორმაში, რომელიც მოსახერხებელია შენახვისათვის. ამძრავი მექანიზმი უზრუნველყოფს ინფორმაციის მატარებლის გადაადგილებას ჩაწერისა და ამოკითხვის ორგანოებთან მიმართებაში.

ინფორმაციის მატარებელს წარმოადგენს ნაკეთობა, რომელიც განსაზღვრული ფიზიკური მოვლენების საფუძველზე ახორციელებს ინფორმაციის მატარებელი შეიძლება იყოს - ლენტა, დისკი, დოლი, ფოტოფირი, ჰოლოგრამა და ა.შ.

გარე დამხსომებელი მოწყობილობის კონტროლერად იწოდება მოწყობილობა, რომელიც უზრუნველყოფს დამგროვებლის მუშაობის მართვას და საბორტო გამოთვლითი მანქანის ინტერფეისთან შეუღლებას.

გარე დამხსომებელი მოწყობილობების სპეციფიკა მდგომარეობს მის დამგროვებელში. გარე დამხსომებელი მოწყობილობების ტიპიურ დამგროვებლებს წარმოადგენენ:

- დამგროვებული მაგნიტურ ლენტზე;
- დამგროვებული მაგნიტურ დისკებზე;
- დამგროვებული მოქნილ მაგნიტურ დისკებზე;
- დამგროვებული მაგნიტურ ბარათებზე;
- დამგროვებული ცილინდრულ მაგნიტურ დომენებზე.

ლექცია 7

მეხსიერების ორგანიზაციის ხერხები ოპერატიულ დამხსომებელ მოწყობილობაში

7.1.მისამართიანი მეხსიერება

დამხსომებელი მოწყობილობის დანიშნულებაა დიდი მოცულობის ორობითი მონაცემების შენახვა, ამიტომ იგი შეიცავს ერთნაირი დამხსომებელი ელემენტების სიმრავლეს, რომლებიც ქმნიან დამხსომებელ მასივს (დმ). მასივი დაყოფილია ცალკეულ უჯრედებად, რომელთაგან თითოეულის დანიშნულებაა ორობითი კოდის შენახვა, რომელშიც თანრიგების რაოდენობა განისაზღვრება მეხსიერების ამორჩევის სიგანით, (კერძოდ, ის შეიძლება იყოს ერთი, ნახევარი ან რამდენიმე მანქანური სიტყვა).

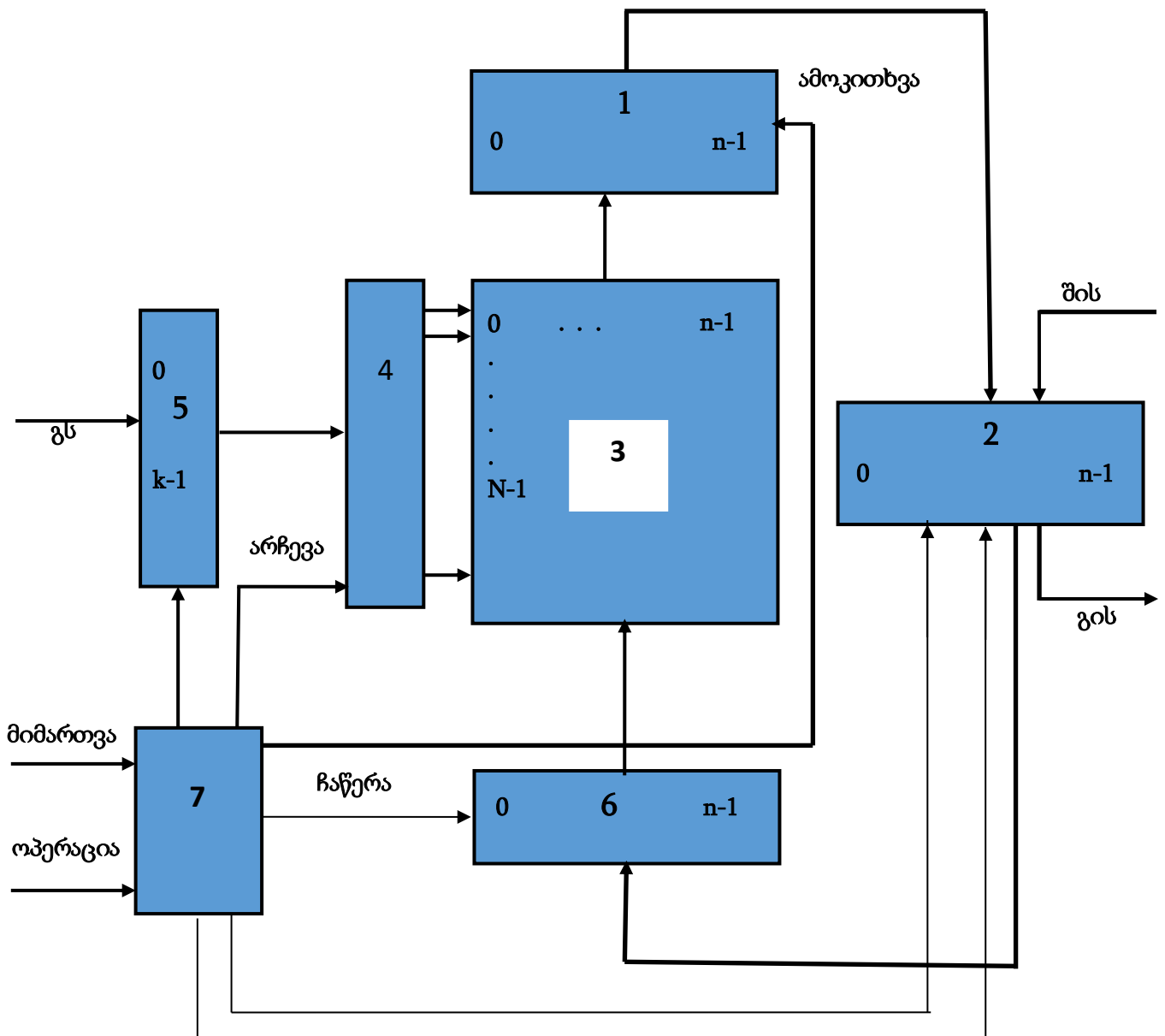
მეხსიერების ორგანიზაციის ხერხი დამოკიდებულია დამხსომებელ მასივში ინფორმაციის განლაგებისა და ძიების მეთოდებზე. ამ ნიშნის მიხედვით განასხვავებენ მისამართიან, ასოციაციურ და სტეკურ მეხსიერებას.

მისამართიანი მეხსიერება. მისამართიანი ორგანიზაციის მეხსიერებაში ინფორმაციის განლაგება და ძიება დამხსომებელ მასივში დაფუძნებულია სიტყვის (რიცხვები, ბრძანებები და ა.შ.) შენახვის მისამართის გამოყენებაზე, რომელსაც წარმოადგენს დამხსომებელი მასივის იმ უჯრედის ნომერი, რომელშიც განლაგებულია ეს სიტყვა.

დამხსომებელ მასივში სიტყვის ჩაწერისას (ან ამოკითხვისას), რომელიც ინიცირებას უკეთებს ამ ოპერაციას, ბრძანებამ უნდა მიუთითოს მისამართი (უჯრედის ნომერი), რომლის მიხედვითაც უნდა მოხდეს ჩაწერა (ან ამოკითხვა).

მისამართიანი მეხსიერების ტიპური სტრუქტურა მოცემულია ნახ.7.1-ზე. ის შეიცავს დამხსომებელ მასივს (დმ) შემდგარს $N \times n$ -თანრიგოვან უჯრედებისგან და მის აპარატულ მოჭარჩოებას, რომელიც მოიცავს მისამართის რეგისტრს (მრგ), ინფორმაციულ რეგისტრს (ირგ), სამისამართო ამორჩევის ბლოკს (საბ) მისამართის

დეშიფრატორით (მდშ), ამოკითხვის გამაძლიერებლების ბლოკს (აგბ), სათანრიგო გამაძლიერებლების ბლოკს (სგბ) - ჩაწერის სიგნალების მაფორმირებლებს და მეხსიერების მართვის ბლოკს (მმბ).



ნახ.7.1 მისამართიანი მეხსიერების სტრუქტურა

1. ამოკითხვის გამაძლიერებლების ბლოკი
2. ინფორმაციული რეგისტრი

3. დამხსომეხელი მასივი
4. მისამართების ამორჩევის ბლოკი
5. მისამართის რეგისტრი
6. ჩაწერის სიგნალების მაფორმირებელი ბლოკი
7. მეხსიერების მართვის ბლოკი

მისამართის რეგისტრის თანრიგობა განისაზღვრება გამოსახულებით

$$R \geq \text{int}(\log_2 N).$$

მისამართის რეგისტრში მისამართის კოდის მიხედვით მისამართის ამორჩევის ბლოკი (მაბ) აფორმირებს მეხსიერების შესაბამის უჯრედში სიგნალებს, რომლებიც საშუალებას იძლევა განახორციელდეს უჯრედში სიტყვის ამოკითხვა ან ჩაწერა.

მეხსიერებასთან მიმართვის ციკლი ინიცირდება მეხსიერების მართვის ბლოკში გარედან „მიმართვა“ სიგნალის მიწოდებით. ციკლის მიმართვის საერთო ნაწილი მოიცავს მისამართის რეგისტრში (მრ) სამისამართო სალტიდან (სს) საჭირო უჯრედის მისამართის მიღებას და მეხსიერების მართვის ბლოკში მმართველი სიგნალის „ოპერაცია“ მიღებას და გაშიფვრას, რომელიც მიუთითებს (სიგნალი „ოპერაცია“) მოთხოვნილი ოპერაციის სახეს (ამოკითხვა ან ჩაწერა).

მისამართის მართვის ბლოკი ახდენს მისამართის დეშიფრირებას და ამოკითხვის სიგნალს აგზავნის მისამართით მოცემულ დამხსომეხელი მასივის უჯრედში. ამასთან უჯრედში ჩაწერილი სიტყვის კოდი ამოკითხება ამოკითხვის გამაძლიერებლის ბლოკის გამაძლიერებლებით და გადაიცემა ინფორმაციულ რეგისტრში (ირგ).

შემდეგ, თუ მეხსიერება არის დამრღვევი ამოკითხვით (ასეთ შემთხვევაში უჯრედის დამხსომეხელი ელემენტები დგებიან ნულოვან მდგომარეობაში), მაშინ წარმოებს უჯრედში ინფორმაციის რეგისტრიდან ამოკითხული სიტყვის რეგენერაცია. ამოკითხვის ოპერაცია მთავრდება ინფორმაციული რეგისტრიდან სიტყვის გაცემით გამოსასვლელ ინფორმაციულ სალტეზე.

ჩაწერისას მიმართვის ციკლის ზემოთ აღნიშნული საერთო ნაწილის შესრულების შემდეგ მიმდინარეობს ჩასაწერი სიტყვის მიღება შესასვლელი ინფორმაციული სალტიდან ინფორმაციულ რეგისტრში. თვით ჩაწერა შედგება ორი ოპერაციისგან: უჯრედის გასუფთავება (ნულში ჩამოყრა) და საკუთრივ ჩაწერა.

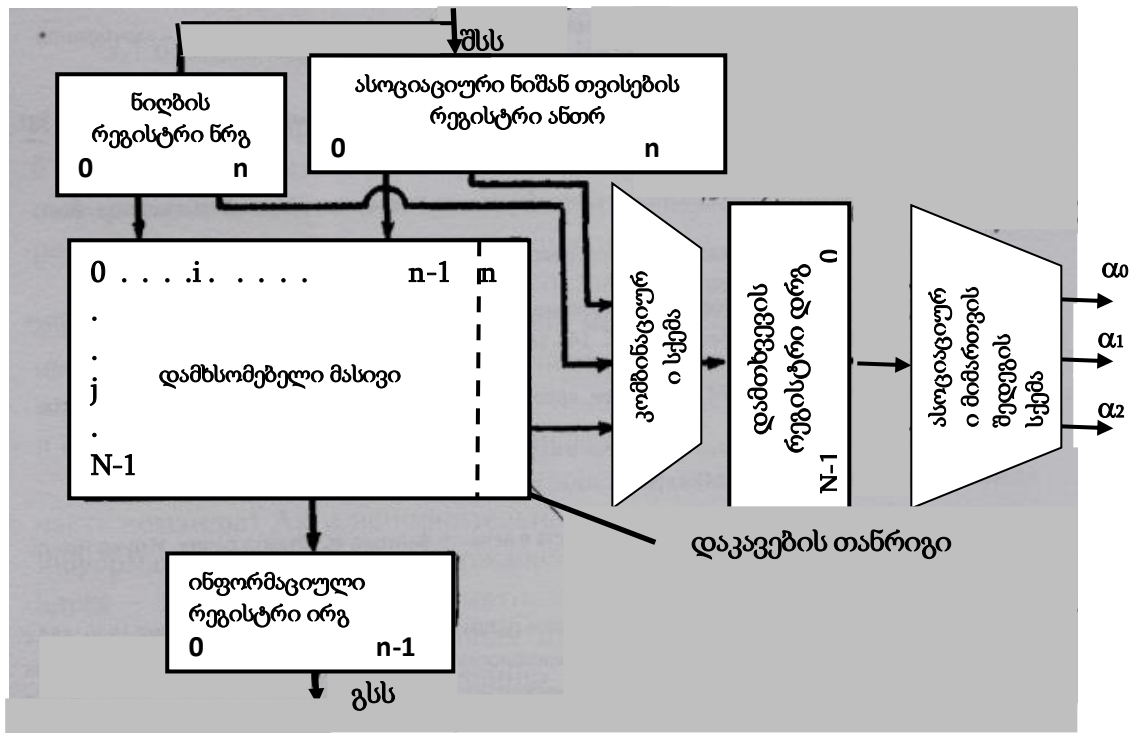
ამისთვის მისამართის არჩევის ბლოკი დასაწყისში ახდენს იმ უჯრედის ამორჩევას და გასუფთავებას, რომელიც მისამართის რეგისტრშია დაფიქსირებული. გასუფთავება ხდება უჯრედში სიტყვის ამოკითხვის სიგნალებით, მაგრამ ამასთან ერთად იბლოკება ამოკითხვის მაძლიერებლები და ამოკითხვის მაძლიერებლების ბლოკიდან (ირგ) - ში ინფორმაცია არ მიეწოდება. ამის შემდეგ სამისამართო არჩევის ბლოკის მიერ არჩეულ უჯრედში ჩაიწერება სიტყვა ინფორმაციული რეგისტრიდან.

მეხსიერების მართვის ბლოკი გამოიმუშავებს (გენერირებს) მმართველი სიგნალების აუცილებელ თანმიმდევრობას, რომელიც ინიცირებას უკეთებს მეხსიერების ცალკეული კვანძების მუშაობას.

7. 2. ასოციაციური მეხსიერება

ამ ტიპის მეხსიერებაში საჭირო ინფორმაციის მოძიება წარმოებს არა მისამართის მიხედვით, არამედ მისი შინაარსის (ასოციაციური ნიშან-თვისების მიხედვით). ამასთან ძიება ასოციაციური ნიშან-თვისების მიხედვით (ან თანმიმდევრობით ამ ნიშან-თვისებების ცალკეული თანრიგების მიხედვით) მიმდინარეობს პარალელურად დროში დამხსომეხელი მასივის ყველა უჯრედებისთვის. ბევრ შემთხვევაში ასოციაციური ძიება საშუალებას იძლევა არსებითად გამარტივდეს და დაჩქარდეს მონაცემთა დამუშავება. ეს კი მიიღწევა იმის შედეგად, რომ ასეთი ტიპის მეხსიერებაში ინფორმაციის ამოკითხვის ოპერაცია შეთავსებულია რიგი ლოგიკური ოპერაციების შესრულებასთან.

ასოციაციური მეხსიერების ტიპური სტრუქტურა წარმოდგენილია ნახ.7.2-ზე.



ნახ. 7.2. ასოციაციური მეხსიერების სტრუქტურა

შსს---შესასვლელი საინფორმაციო სალტე

გსს--გამოსასვლელი საინფორმაციო სალტე

დამსომეხელი მასივი შეიცავს $N * (n+1)$ -თანრიგების უჯრედებს (თანრიგიან უჯრედებს). უჯრედის დაკავებულობის ჩვენებისთვის გამოიყენება სამომსახურო Π -ური თანრიგი (0-უჯრედი თავისუფალია, 1-უჯრედში ჩაწერილია სიტყვა).

შესასვლელ საინფორმაციო სალტეზე (შსს) ასოციაციური ნიშან-თვისების რეგისტრზე (ანთრ) მიეწოდება Π -თანრიგიანი ასოციაციური მიმართვა, ხოლო ნიღბის რეგისტრზე (წრ)-ძიების ნიღბის კოდი, ამასთან ერთად ნიღბის რეგისტრის Π -ური თანრიგი დგება 0-ში. ასოციაციური ძიება წარმოებს ასოციაციური ნიშან-თვისების რეგისტრის მხოლოდ იმ თანრიგების ერთობლიობისთვის, რომელთაც შეესაბამებათ 1 ნიღბის რეგისტრში (ასოციაციური მეხსიერების რეგისტრის შეუნიღბავი თანრიგები).

იმ სიტყვებისთვის, რომლებშიც ციფრები თანრიგებში დაემთხვევა ასოციაციური მეხსიერების რეგისტრის არაშენიღბულ თანრიგებს, მაშინ კომბინაციური სქემა (კს)

დამთხვევის რეგისტრის (დრგ) შესაბამის თანრიგებში დააყენებს 1-ს, ხოლო სხვა დარჩენილ თანრიგებში 0-ს.

ამრიგად, დამთხვევის რეგისტრის i -ური თანრიგის მნიშვნელობა განისაზღვრება გამოსახულებით

$$i = n - 1$$

$$\text{დრ}(j) = \bigwedge_{i=0}^{\text{ანთრგ}[i] + \text{დმ}[j,i] \vee \text{ნრგ}[i]} 1, 0 \leq j \leq N - 1,$$

$$i = 0$$

სადაც ანთრგ $[i]$, ნრგ $[i]$ და დმ $[j,i]$ - შესაბამისად i -ური თანრიგის მნიშვნელობაა ანთრგ, ნრგ რეგისტრებში და დამხსომეხელი მასივის j -ურ უჯრედში.

ასოციაციური მიმართვის (ამ) შედეგის ფორმირების კომბინაციური სქემა სიტყვისაგან რომელიც წარმოშობილია დამთხვევის რეგისტრში აფორმირებს სიგნალებს

$$\alpha_0 \alpha_1 \alpha_2,$$

რომლებიც შესაბამისად შეესაბამებიან შემთხვევებს: დამხსომეხელ მასივში სიტყვები არ არსებობს, სიტყვები აკმაყოფილებენ ასოციაციურ ნიშან –

თვისებებს, ადგილი აქვს ერთ ან ერთზე უფრო მეტი ასეთი სიტყვის არსებობას.

ამისათვის კომბინაციური სქემა რეალიზებას უკეთებს შემდეგ ბულებს ფუნქციებს:

$$\alpha_0 = \bigwedge_{j=0}^{N-1} \overline{\text{დრგ}[j]}$$

$$\alpha_1 = \text{დრგ}[0] \cdot \overline{\text{დრგ}[1]} \dots \overline{\text{დრგ}[j]} \dots \overline{\text{დრგ}[N-1]} \vee \overline{\text{დრგ}[0]} \cdot \overline{\text{დრგ}[1]} \dots \overline{\text{დრგ}[j]} \dots \overline{\text{დრგ}[N-1]} \vee \overline{\text{დრგ}[0]} \cdot \overline{\text{დრგ}[1]} \dots \overline{\text{დრგ}[j]} \dots \overline{\text{დრგ}[N-1]}$$

$$\alpha_2 = \overline{\alpha_1} \cdot \overline{\alpha_0}$$

დამთხვევის რეგისტრის (დრგ) შიგთავსის ფორმირება და $\alpha_0, \alpha_1, \alpha_2$ სიგნალების შესაბამისად ასოციაციური ნიშან-თვისების რეგისტრის (ანთრგ), ნილბის რეგისტრის (ნრგ) და დამხსომეხელი მასივის (დმ) შიგთავსის მიხედვით იწოდება ასოციაციის კონტროლის ოპერაციად. ეს ოპერაცია წარმოადგენს ჩაწერისა და ამოკითხვის ოპერაციების შემადგენელ ნაწილს, თუმცა მას გააჩნია თავისი დამოუკიდებელი მნიშვნელობაც.

ამოკითხვისას დასაწყისში ხდება ასოციაციის კონტროლი ასოციაციური ნიშან-თვისების მიხედვით ასოციაციური ნიშან-თვისების რეგისტრში. შემდეგ, როცა $\alpha_0 = 1$,

ამოკითხვა უგულებელყოფილი ხდება სამიუბო ინფორმაციის არ არსებობის გამო, როცა $\alpha_1 = 1$, ამოკითხვა ინფორმაციულ რეგისტრში (ირგ) ნაპოვნი სიტყვა, როცა $\alpha_2 = 1$, მაშინ ინფორმაციულ რეგისტრში ამოკითხვა სიტყვა იმ უჯრედიდან, რომელსაც გააჩნია ყველაზე ნაკლები ნომერი იმ უჯრედებს შორის, რომლებიც მონიშნულია 1-ით დამთხვევის რეგისტრში. ინფორმაციული რეგისტრიდან ამოკითხული სიტყვა გაიცემა გამოსასვლელ სალტზე (გს).

ჩაწერის შემთხვევაში თავდაპირველად იძებნება თავისუფალი უჯრედი. ამისათვის სრულდება ასოციაციის კონტროლის ოპერაცია როცა ასოციაციის ნიშან-თვისების რეგისტრში დაფიქსირებულია ანთრგ = 111...10 და ნილბის რეგისტრში ნრგ = 000...01. ამასთან დამთხვევის რეგისტრში თავისუფალი უჯრედები მონიშნებიან 1-ით. ჩაწერისათვის ამოირჩევა თავისუფალი უჯრედი ყველაზე უფრო მცირე ნომრით. მასში იწერება სიტყვა, რომელიც მიწოდებულია შესასვლელი საინფორმაციო სალტიდან ინფორმაციის რეგისტრში.

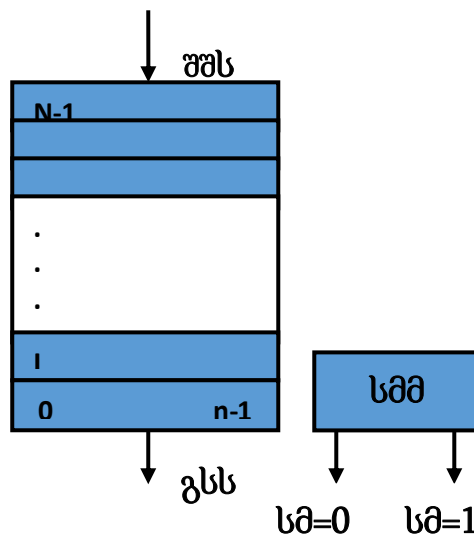
ასოციაციის კონტროლის ოპერაციის საშუალებით შესაძლებელია მეხსიერებიდან სიტყვის ამოკითხვის გარეშე დამთხვევის რეგისტრის შიგთავსის მიხედვით განვსაზღვროთ თუ მეხსიერებაში რამდენია ისეთი სიტყვა, რომლებიც აკმაყოფილებენ ასოციაციურ ნიშან-თვისებას, მაგალითად, რეალიზებულ იქნეს ასეთი ტიპის შეკითხვები: მოცემულ დისციპლინაში ჯგუფში რამდენ სტუდენტს აქვს ფრიადი შეფასება. შესაბამისი კომბინაციური სქემების გამოყენებისას ასოციაციურ მეხსიერებაში შესაძლებელია შესრულდეს საკმარისად რთული ლოგიკური ოპერაციები, ისეთები როგორიცაა უდიდესი (უმცირესი) რიცხვის მოძებნა, სიტყვების მოძებნა, რომლებიც მოთავსებულია განსაზღვრულ საზღვრებში და სხვ.

უნდა ავღნიშნოთ, რომ ასოციაციური მეხსიერებისთვის აუცილებელია დამხსომეხელი ელემენტები, რომლებიც უშვებენ მასში ჩაწერილი ინფორმაციის ამოკითხვას ინფორმაციის დარღვევის გარეშე. ეს დაკავშირებულია იმასთან, რომ ასოციაციური ძიების დროს ამოკითხვა მიმდინარეობს მთელი დამხსომეხელი მასივის მიხედვით ყველა შეუნიღბავი თნრიგებისთვის და გამომდინარე აქედან, ამიტომ შეუძლებელია სადმე იქნეს შენახული ამოკითხვის შედეგად დროებით დარღვეული ინფორმაცია.

7. 3. სტეკური მეხსიერება

სტეკური მეხსიერება, ისე როგორც ასოციაციური წარმოადგენს უმისამართოს.

სტეკური მეხსიერება ორი სახისაა: 1. FIFO -პირველი მოვიდა, პირველი გავიდა და FILO-პირველი მოვიდა, ბოლო წავიდა. FIFO-ს ტიპის სტეკურ მეხსიერებაში (ნახ.7.3) უჯრედები ქმნიან ერთგანზომილებიან მასივს, რომელშიც მეზობელი უჯრედები შეერთებულია სიტყვების გადაცემის თანრიგობრივი წრედებით.



ნახ. 7 . 3

სმ --- სტეკის მოვლელ- მაჩვენებელი

შსს ---შესასვლელი საინფორმაციო სალტე

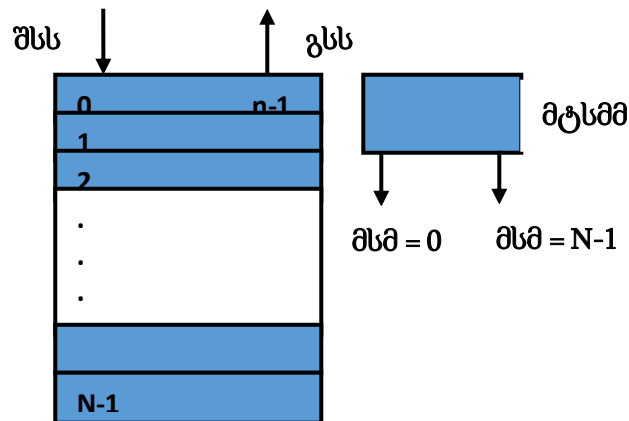
გსს --გამოსასვლელი საინფორმაციო სალტე

სმ ---სტეკის მაჩვენებელი

სტეკის შევსება ხდება ერთი მიმართულებით, ამასთან თავისუფალ უჯრედში სიტყვები ჩაიწერება მიმდევრობითი ნომრებით, დაწყებული ზედა უჯრედიდან, ხოლო ამოკითხვისას მეხსიერებიდან სიტყვის მოშორება წარმოებს სტეკის მეორე მხრიდან და მხოლოდ ქვედა უჯრედიდან (უჯრედი 0). დანარჩენი სიტყვების დაძვრა სტეკში ხდება ქვევით მეზობელ უჯრედებში კლებადი ნომრებით. ამრიგად, ამოკითხვის რიგისობა ექვემდებარება წესს: პირველი მოვიდა-მოემსახურეთ პირველ რიგში.

სტეკური მეხსიერების შემადგელობაში შედის სტეკის მთვლელ-მაჩვენებელი (სმმ). ახალი სიტყვის ჩაწერისას სტეკში სტეკის მთვლელ-მაჩვენებლის შიგთავსი იზრდება ერთით, ხოლო სიტყვის ამოკითხვისას კი მცირდება ერთით. როცა სტეკის მთვლელ-მაჩვენებლის სმმ მნიშვნელობა ნულია, ე.ი. $სმმ = 0$, მაშინ გამოძუწავდება სიგნალი $სმ = 0$, რომელიც მიუთითებს რომ სტეკი ცარიელია. როცა $სმმ = N-1$ ფორმირდება სიგნალი $სმ = 1$, რომელიც მიუთითებს, რომ სტეკი შევსებულია. კომპიუტერში სტეკური მეხსიერება გამოიყენება სხვადასხვა რიგების აპარატული ორგანიზაციისთვის.

FIFO-ს სტეკურ მეხსიერებაში ახალი სიტყვის ჩაწერა მიმდინარეობს ზედა უჯრედში (ნახ.7.4).



ნახ.7.4

მტმმ - მაღაზიის ტიპის სტეკის მთვლელ - მაჩვენებელი

ამასთან ყველა ადრე ჩაწერილი სიტყვები (0-ვან უჯრედში მყოფი სიტყვის ჩათვლით) დაიდვრებიან ქვევით მეზობელ უჯრედებში, რომელთაც გააჩნიათ ერთით მეტი ნომრები. ამოკითხვა შეიძლება მხოლოდ ზედა (ნულოვანი) უჯრედიდან. ამასთან თუ ხდება ამოკითხვა წაშლით, მაშინ ყველა სხვა დანარჩენი სიტყვები მეხსიერებაში დაიდვრება ზევით მცირე ნომრებიან მეზობელ უჯრედებში. ამ მეხსიერებაში სიტყვების ამოკითხვის რიგისობა შეესაბამება შემდეგ წესს: ბოლო მოვიდა - მოემსახურეთ პირველად. ასეთი პრინციპითაა აგებული ავტომატური ცეცხლსასროლი იარაღის მაღაზიიდან პატრონების მიწოდების ორგანიზაცია. ამიტომ **FIFO** ტიპის სტეკური მეხსიერებას ასევე უწოდებენ მაღაზიური სახის მეხსიერებას. მაღაზიური მეხსიერება

მარაგდება მაღაზიის მრიცხველ-მაჩვენებლით, რომლის მოქმედება ანალოგიურია სტეკის მაჩვენებლისა. მაღაზიისმაგვარი მეხსიერება ეფექტურია ჩაშენებული ალგორითმებისა და წყვეტების სისტემის რეალიზაციისას.

თანამედროვე საბორტო გამოთვლით სისტემებში სტეკური მეხსიერება ფართოდ გამოიყენება წყვეტის სისტემის ორგანიზაციის დროს წყვეტის ვექტორისა და რეგისტრების შიგთავსის შენახვისათვის, რაც მნიშვნელოვნად ამცირებს ინფორმაციის ჩაწერისა და აღწარმოების დროს პროგრამის წყვეტის პროცესში.

სტეკური მეხსიერება გამოიყენება მიკროპროცესორების დაპროგრამებისას ქვეპროგრამებთან მუშაობის დროს. ამ შემთხვევაში ქვეპროგრამის გამოძახების წინ სტეკში ინახება პროცესორების იმ რეგისტრების შიგთავსი, რომლებიც შესაძლებელია გამოყენებულ იქნეს ქვეპროგრამაში.

7.4 საბორტო კომპიუტერების სამისამართო სივრცე. დამისამართების სახეები და მათი რელიზაციის ხერხები

7.4.1. საბორტო კომპიუტერის სამისამართო სივრცე

სამისამართო სივრცე არის ოპერატიული ან მუდმივი მეხსიერების მისამართების სიმრავლე, რომელიც ხელმისაწვდომია პროცესორის მიერ შესრულებადი პროგრამებისათვის. სამისამართო სივრცის ზომა განისაზღვრება მისამართის კოდის მაქსიმალური თანრიგობით და შესაძლოა გამოისახოს მეხსიერების მინიმალურად დამისამართებადი ელემენტების ერთობლიობაში ან უფრო მსხვილ ერთეულებში. მისამართად იწოდება სპეციალური რიცხვი, რომელიც მიენიჭება მეხსიერების უჯრედს ინფორმაციის ერთეულის შენახვისათვის. ამრიგად, მისამართი განსაზღვრავს ინფორმაციის შენახვის ადგილს მეხსიერებაში.

მისამართი შეიძლება წარმოადგენდეს ბრძანების ნაწილს, რისგანაც შედგება ამოცანის ამოხსნის პროგრამები. თუ ბრძანებაში მითითებული მისამართი უშუალოდ განსაზღვრავს ინფორმაციის განლაგების ადგილს, მაშინ ასეთი მისამართი იწოდება შემსრულებელ მისამართად. თუ შემსრულებელი მისამართი შედგება 16 თანრიგისგან (2 ბაიტი) მეხსიერების მინიმალურად დამისამართებადი ელემენტის 1 ბაიტის შემთხვევაში სამისამართო სივრცე უზრუნველყოფს დაშვებას

$$2^{16}=2^{10} * 2^6=1024 * 64 = 65536 \text{ ბაიტი} = 64 \text{ კ.ბაიტი} = 32 \text{ კილო სიტყვასთან} \\ (k=1024).$$

მოცემულ შემთხვევაში ერთი სიტყვა შეესაბამება 2 ბაიტს.

როგორც წესი, საბორტო კომპიუტერებში მეხსიერების მინიმალურ დამისამართებად ერთეულს წარმოადგენს ბაიტი. სიტყვის მისამართს წარმოადგენს ან მარცხენა (უფროსი) ბაიტის მისამართი ან მარჯვენა (უმცროსი) ბაიტის მისამართი. მაგალითად თუ სიტყვა შედგება 2 ბაიტისგან მისამართები 20 და 21, მაშინ ხშირად სიტყვის მისამართად მიიღება 20 (უფროსი ბაიტის მისამართი).

7.4.2. დამისამართების სახეები და მათი რეალიზაციის ხერხები

საბორტო კომპიუტერში

თანამედროვე საბორტო კომპიუტერებში ფართო გამოყენება ჰპოვა დამხსოვრებად მოწყობილობებმა სამისამართო ორგანიზაციით. ეს იმას ნიშნავს, რომ ინფორმაციის თითოეულ შენახულ ერთეულზე განკუთვნილია შესაბამისი სპეციალური რიცხვი-მისამართი, რომელიც განსაზღვრავს მათი შენახვის ადგილს მეხსიერებაში.

ცნობები მისამართის ან მისი შემადგენელი ნაწილების შესახებ მოცემულია ბრძანებებში. ამ ინფორმაციის გამოყენებით საბორტო კომპიუტერი აწარმოებს მეხსიერებიდან საჭირო პროგრამული მონაცემების ამორჩევას. თავის მხრივ, თვითონ მისამართები შესაძლებელია განვიხილოთ როგორც მონაცემები, რომლებიც ექვემდებარებიან კომპიუტერში დამუშავებას და მიიღებიან გამოთვლების შედეგად.

აუცილებელია ერთმანეთისგან განვასხვავოთ ცნებები: სამისამართო კოდი ბრძანებებში (ბრძანების სამისამართო ნაწილი) და შემსრულებელი მისამართი (შმ).

სამისამართო კოდი (სკ) -ეს არის ინფორმაცია მისამართის შესახებ, რომელიც მოცემულია ბრძანებაში. შემსრულებელი მისამართი - კი ეს არის მეხსიერების უჯრედი, რომელთანაც ფაქტობრივად ხდება მიმართვა. თანამედროვე საბორტო კომპიუტერებში სწრაფვას შემციერებულ იქნეს ბრძანებების კოდის სიგრძე და მოხდეს დაპროგრამებისთვის ხელსაყრელი პირობების შექმნა, მივყავართ იქმადე, რომ სამისამართო კოდი, როგორც წესი, არ ემთხვევა შემსრულებელ მისამართს.

შემსრულებელი მისმართის ფორმირება შესაძლებელია განხორციელდეს სხვადასხვა ხერხებით. ყველაზე უფრო ხშირად გამოიყენება შემდეგი ხერხები:

- საგულისხმო ოპერანდი;
- საგულისხმო მისამართი;
- უშუალო დამისამართება;
- პირდაპირი დამისამართება;
- ფარდობითი დამისამართება ან ბაზირება;
- ორიბი დამისამართება;
- შემოკლებული დამისამართება;
- ცვლადი სიგრძის სიტყვების დამისამართება.

საგულისხმო ოპერანდი ასეთ შემთხვევაში ბრძანებაში ცხადად არაა მითითებული ოპერანდის მისამართის შესახებ, იგი იგულისხმება და ფაქტობრივად მოიცემა ბრძანების კოდის ოპერაციით. ეს მეთოდი გამოიყენება იშვიათად, თუმცა უნდა აღინიშნოს მისი გამოყენების რამდენიმე საჭირო შემთხვევა. მაგალითის სახით შეგვიძლია მოვიყვანოთ დათვლის ბრძანება, რომელშიც რაღაც რიცხვს (მრიცხველის შიგთავსი) ემატება ფიქსირებული ნაზრდი, ხშირად უმცროსი თანრიგის ერთიანის სახით. ოპერანდებიდან ერთი-რიცხვი მრიცხველში ჩვეულებრივ დამისამართებულია ცხადი მეთოდით, მეორე ოპერანდი - ნაზრდი არაა დამისამართებული, კომპიუტერის მეხსიერებაში არ იმყოფება და წარმოადგენს საგულისხმოს.

საგულისხმო მისამართი ბრძანებაში ცხადად არაა მოცემული ოპერაციაში მონაწილე ოპერანდის მისამართის შესახებ ან მისამართის შესახებ რომელშიც უნდა მოთავსდეს ოპერაციის შედეგი, თუმცა ეს მისამართი იგულისხმება. მაგალითად, ბრძანება შეიძლება შეიცავდეს ორივე ოპერანდის მისამართებს, რომლებიც მონაწილეობას იღებენ ოპერაციაში, ამასთან ერთად იგულისხმება, რომ შედეგი ოპერაციისა თავსდება ოპერანდებიდან ერთ-ერთის მისამართზე. სხვა შემთხვევაში ბრძანებაში შესაძლებელია მითითებული იყოს მხოლოდ ერთი ოპერანდის მისამართი, ხოლო მეორის მისამართი რომელსაც წარმოადგენს სპეციალური რეგისტრის შიგთავსი (ეს სპეციალური რეგისტრი იწოდება შედეგის რეგისტრად) იგულისხმება.

უშუალო დამისამართება ბრძანებაში მოცემულია ოპერანდის არა მისამართი, არამედ უშუალოდ თვითონ ოპერანდი. უშუალო დამისამართების დროს არ მოითხოვება მეხსიერებისადმი მიმართვა ოპერანდის ამორჩევისთვის, ასევე მისი შენახვისთვის არ მოითხოვება მეხსიერების უჯრედის მისამართი. ეს საშუალებას იძლევა ამაღლებულ იქნეს კომპიუტერის სწრაფქმედება და შემცირდეს მეხსიერების მოცულობა, რომელიც საჭიროა მოცემული ამოცანებისთვის. უშუალო დამისამართება მოსახერხებელია სხვადასხვა სახის კონსტანტების შენახვისათვის. თუმცა ამ შემთხვევაში ოპერანდის სიგრძე ნაკლები უნდა იყოს ბრძანების კოდზე, იმდენად რამდენადაც ბრძანების თანრიგების ნაწილი დაკავებულია ოპერაციის კოდის მიერ.

პირდაპირი დამისამართება ამ დროს შემსრულებელი მისამართი ემთხვევა ბრძანების სამისამართო ნაწილს. დამისამართების ეს მეთოდი იყო ზოგადად მიღებული პირველ გამოთვლით კომპიუტერებში და გრძელდება ამჟამადაც მისი გამოყენება სხვა მეთოდებთან კომბინაციაში.

ფარდობითი დამისამართება შემსრულებელი მისამართი განისაზღვრება ბრძანების სამისამართო კოდის შეჯამებით რაღაც A_B რიცხვთან, რომელიც იწოდება საბაზო მისამართად

$A_B = A_3 + A_B$, სადაც A_B - შემსრულებელი მისამართია; A_3 - ბრძანების სამისამართო კოდი.

საბაზო მისამართების შენახვისათვის კომპიუტერში შეიძლება გათვალისწინებული იყოს რეგისტრები ან სპეციალურად ამ მიზნისთვის გამოყოფილი იყოს მეხსიერების უჯრედები (საბაზო რეგისტრები). საბაზო მისამართში თანრიგების რაოდენობას ირჩევენ ისეთნაირად, რომ ბრძანების სამისამართო ნაწილთან ერთად შესაძლებელი იყოს მეხსიერების ნებისმიერ უჯრედთან მიმართვა (ნებისმიერი უჯრედის დამისამართება). ბრძანებაში საბაზო მისამართის შენახვის ადგილის მითითებისთვის ბრძანებაში შეიძლება გამოყოფილ იქნეს შესაბამისი ველი B .

ბაზირების გამოყენებისას თანრიგების ეკონომიის მიზნით ბრძანების სამისამართო კოდი A_3 ჩვეულებრივ წარმოადგენს საბაზო მისამართზე უფრო მოკლეს და უწოდებენ მას წანაცვლებას (აღნიშნავენ ასო C-თი). წანაცვლება განსაზღვრავს ოპერანდის მდებარეობას მასივის დასაწყისთან მიმართებაში, რომელიც მოიცემა საბაზო მისამართით A_8 . ნახ.7.5 ა) საშუალებას იძლევა გავარკვიოთ თუ როგორ ხდება შემსრულებელი მისამართის ფორმირების პროცესი.

უმრავლეს ნაწილში შემსრულებელი მისამართი ბაზირების დროს გამოისახება შეჯამების სახით შემდეგი გამოსახულების შესაბამისად (ნახ.7.5 ბ)

$$A_8 = \begin{cases} <(B)+(C), \text{ თუ} \\ & (B) \neq 0; \\ \\ & (C), \text{ თუ } (B) = 0, \end{cases}$$

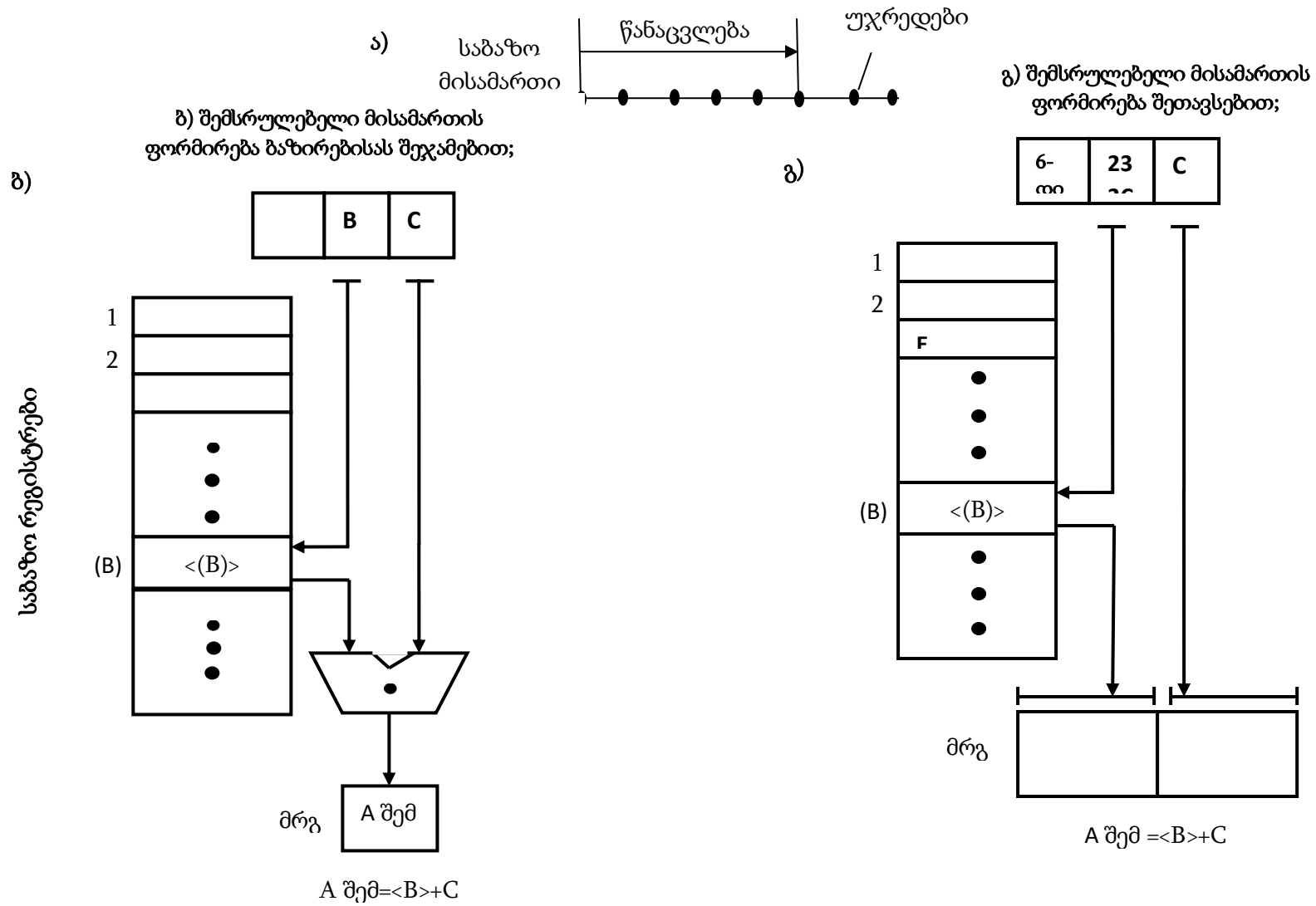
სადაც (B) და (C) - კოდებია (რიცხვებია), რომლებიც დგას ბრძანების B და C ველებში, $<(B)>$ რეგისტრის შიგთავსია ნომრით (B). როცა $(B)=0$ ფარდობითი დამისამართება იბლოკება.

შემსრულებელი მისამართის წარმოშობისთვის შეჯამება დაკავშირებულია დროის დაკარგვასთან. ამიტომ იყენებენ შემსრულებელი მისამართის ფორმირების სხვა მეთოდსაც, რომელსაც შერწყმის (შეკავშირების) მეთოდს უწოდებენ (ნახ.7.5 გ).

ასეთ შემთხვევაში შემსრულებელი მისამართის საბაზო მისამართი შეიცავს უფროს თანრიგებს, ხოლო წანაცვლება კი უმცროს თანრიგებს, რომლებიც ერთიანდებიან რეგისტრ A-ში (რეგ A) შემდეგი ოპერაციის შესაბამისად

$$\text{რეგ } A := \text{რეგ } B^* \text{ რეგ } K[C],$$

სადაც * - გაერთიანების ნიშანია; რეგ $K[C]$ - ბრძანების თანრიგებია წანაცვლების შესაბამისი.



ნახ.7.5

ფარდობითი დამისამართება უზრუნველყოფს ეგრეთწოდებულ პროგრამების “გადაადგილებადობას“, ე.ი. მეხსიერებაში პროგრამების გადაადგილების შესაძლებლობას თვით პროგრამის შიგა ცვლილების გარეშე.

ფარდობითი დამისამართება აიოლებს მეხსიერების დანაწილებას იმ შემთხვევისთვის, როცა საჭიროა რთული პროგრამების შედგენა რამდენიმე პროგრამისტის მიერ. ასეთ შემთხვევაში თითოეულ პროგრამისტს შეუძლია მეხსიერებაში ფლობდეს პროგრამის თავის ნაწილს დაწყებული ნულოვანი მისამართით. პროგრამის გაერთმთლიანებისთვის საკმარისია მხოლოდ მოცემულ იქნეს საბაზო მისამართების შესაბამისი მნიშვნელობები.

ირიბი დამისამართება. ბრძანების სამისამართო კოდი მიუთითებს მეხსიერების უჯრედის მისამართს, რომელშიც იმყოფება ოპერანდის ან ბრძანების მისამართი. ამრიგად, ირიბი დამისამართება შეიძლება სხვანაირად განსაზღვრულ იქნეს როგორც „მისამართის დამისამართება“. მეხსიერების უჯრედთან მიმართვა ოპერანდის არჩევისთვის ირიბი დამისამართების დროს სრულდება რამდენიმე ეტაპად. პირველ ეტაპზე ფორმირდება მეხსიერების უჯრედის შემსრულებელი მისამართი, სადაც ინახება ოპერანდის მისამართი. მეორე ეტაპზე ხდება ოპერანდის მისამართის ამოღება უჯრედიდან და მისამართის რეგისტრში (მრგ) მისი გადაგზავნა და ბოლოს მესამე ეტაპზე მიმდინარეობს ოპერანდის ამოღება იმ მისამართიდან, რომელიც დაფიქსირებულია მისამართის რეგისტრში.

შემოკლებული დამისამართება. ბრძანების კოდის სიგრძის შესამცირებლად ხშირად იყენებენ ეგრეთწოდებულ შემოკლებულ დამისამართებას. მისი არსი მდგომარეობს იმაში, რომ ბრძანებაში მოიცემა მისამართების მხოლოდ უმცროსი თანრიგები. ამასთან ამ შემთხვევაში მხოლოდ უფროსი თანრიგები იგულისხმებიან როგორც ნულოვანი. ასეთი დამისამართება შესაძლებლობას იძლევა გამოყენებულ იქნეს ფიქსირებული უჯრედების მხოლოდ არადიდი ჯგუფი მოკლე მისამართებით და შესაძლებელია გამოვიყენოთ ის მხოლოდ დამისამართების სხვა მეთოდებთან ერთად.

ცვლადი სიგრძის სიტყვების დამისამართება. გამოყენებული სისტემების ეფექტურობა, რომელთა დანიშნულებაა მონაცემთა დამუშავება, მაღლდება მაშინ როცა შესაძლებელია შესრულდეს ოპერაციები ცვლადი სიგრძის სიტყვებით. ასეთ შემთხვევაში კომპიუტერში გათვალისწინებული უნდა იყოს ცვლადი სიგრძის სიტყვების

დამისამართება, რომლის რეალიზება ჩვეულებრივ ხდება ბრძანებაში მეხსიერებაში სიტყვის ადგილმდებარეობის დასაწყისის და მისი სიგრძის მითითების მიხედვით.

საბორტო კომპიუტერებიდან დამისამართების სისტემა ყველაზე მეტად განვითარებულია A-15A საბორტო კომპიუტერში. A-15A საბორტო კომპიუტერში შემსრულებელი მისამართის ფორმირება შესაძლებელია განხორციელდეს 8 სხვადასხვა ხერხით (ცხრილი 7.1), რომლებიც თავის მხრივ განისაზღვრებიან დამისამართების ნიშან-თვისებით (დნთ).

ცხრილში 7.1 გამოყენებულია დამისამართების საბაზო კონსტანტების შემდეგი აღნიშვნები:

ცხრილი 7. 1

დამისამართების ნიშან-თვისება			შემსრულებელი მისამართის ფორმირების სქემა
A	B	B	
0	0	0	$A\# = C$
0	0	1	$A\# = \text{რგB1} * C$
0	1	0	$A\# = \text{შ}[C]$
0	1	1	$A\# = \text{მ}[\text{რგB2} * C]$
1	0	0	$A\# = \text{ბმ} \pm C$
1	0	1	$A\# = \text{რგU1} + C$
1	1	0	$A\# = \text{რგU2} + C$
1	1	1	$A\# = \text{რგB3} * C$

რგB1, რგB2, რგB3 - 6 თანრიგიანი საბაზო რეგისტრების შიგთავსი;

რგU1, რგU2-16 თანრიგიანი ინდექსური რეგისტრების შიგთავსი;

ბმ-16 თანრიგა ბრძანებების მრიცხველის შიგთავსი;

შ [C] -უჯრედის შიგთავსი მისამართით C;

შ [B2 * C] -უჯრედის შიგთავსი მისამართით რგB2 * C, სადაც * -არის სიმბოლო ორი კოდის ერთ 16 თანრიგიან სიტყვაში გაერთიანებისა (შერწყმისა).

A-15A საბორტო კომპიუტერის მისამართის ფორმატი

ცხრილი 7.2

სიტყვის თანრიგი	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ტრიალები	0			1			2	3			4			5		
დანიშნულება	ბლოკის ნომერი						ბლოკში მეხსიერების უჯრედის ნომერი									

A-15A საბორტო კომპიუტერის მეხსიერების ველის განაწილება

ცხრილი 7.3

ექვსი უფროსი თანრიგის კოდი	რვაობითი ჩაწერა	ბლოკის დასახელება	უჯრედების რაოდენობა
000000	00	ოღმ1	1024
000001	01	ოღმ2	1024
000010	02	თავისუფალი ზონა	1024
000011	03	თავისუფალი ზონა	1024
000100	04	თავისუფალი ზონა	1024
000101	05	თავისუფალი ზონა	1024
000110	06	ხვდმ	128
000111	07	კოტროლის პუნქტის დამგროვებელი	8
001xxx	1X	თავისუფალი ზონა	8192
010 xxx	2X	თავისუფალი ზონა	8192
011 xxx	3X	თავისუფალი ზონა	8192
100 xxx	4X	ხდმ 1	8192
101 xxx	5X	ხდმ 2	8192
110 xxx	6X	ხდმ 3	8192
111 xxx	7X	ხდმ 4	8192

ხდმ - ხანგრძლივადიანი დამხსომეხელი მოწყობილობა

7.2 ცხრილში მოცემული მისამართის ფორმატისა და 7.3 ცხრილით წარმოდგენილი მეხსიერების მისამართების ველის განაწილების შესაბამისად მისამართის ფორმირება ხორციელდება შემდეგნაირად:

როცა დნთ=000, შემსრულებელი მისამართი ტოლია წანაცვლების, რომელიც უშუალოდაა ჩაწერილი ბრძანებაში, ამასთან მისი უფროსი 6 თანრიგი ტოლია ნულის. მისამართების ველის განაწილების გათვალისწინებით ეს ნიშნავს ოპერატიულ დამხსომებელ მოწყობილობა ერთის (ოდმ1) უჯრედებთან პირდაპირი დამისამართებას.

როცა დნთ=001, შემსრულებელი მისამართი წარმოიშობა რგB1 6 თანრიგა კოდის შეერთებით ბრძანების 10 თანრიგა წანაცვლებასთან. ბაზის კოდი იკავებს 6 უფროს თანრიგს, წანაცვლება კი შემსრულებელი მისამართის უმცროს 10 თანრიგს.

როცა დნთ=010, შემსრულებელი მისამართი წარმოადგენს იმ უჯრედის შიგთავსს, რომლის მისამართი ტოლია წანაცვლების და რომელიც ჩაწერილია ბრძანებაში. მისამართების ველის განაწილების გათვალისწინებით წანაცვლება წარმოადგენს თავის მხრივ ირიბ მისამართს და მიუთითებს ოდმ1 ბლოკში მეხსიერების უჯრედის მისამართს, რომლის შიგთავსი წარმოადგენს სწორედ შემსრულებელ მისამართს.

როცა დნთ=011, მაშინ შემსრულებელი მისამართი წარმოადგენს მეხსიერების უჯრედის შიგთავსს, რომლის მისამართი ტოლია $რგ < B2 * C$, რაც იმის მაჩვენებელია, რომ შემსრულებელი მისამართი შესაძლებელია იმყოფებოდეს როგორც ხანგრძლივ, ისე ოპერატიული დამხსომებელი მოწყობილობის მეხსიერების ნებისმიერ უჯრედში.

როცა დნთ=100, შემსრულებელი მისამართი მიიღება როგორც ბრძანებების მთვლელის შიგთავსისა და წანაცვლების ჯამი. მოცემულ შემთხვევაში შეჯამების (შეკრების) ოპერაცია წარმოებს მხოლოდ მისამართის 10 უმცროს თანრიგებზე (ბრძანებების მთვლელის შიგთავსი). გადატანის წარმოშობისას მომდევნო (მეთერთმეტე) თანრიგში, ხდება მისი გადაგდება. ამიტომ ოპერაცია შესაძლებელია განხილულ იქნეს ასევე როგორც ბრძანებათა მთვლელის შიგთავსისგან წანაცვლების სიდიდის გამოკლება. ოპერაციის შედეგად მიიღება 16 თანრიგა შემსრულებელი მისამართი, რომლის უფროსი ექვსი თანრიგი (რომლებიც განსაზღვრავენ დამხსომებელი მოწყობილობის ბლოკს) რჩება უცვლელი. დამისამართების ეს სახე მოსახერხებელია გამოყენებულ იქნეს ციკლების ორგანიზაციისათვის.

როცა დნთ=101, შემსრულებელი მისამართი წარმოიშობა როგორც რგH1-ის შიგთავსის და წანაცვლების ჯამი.

როცა დნთ=110, შემსრულებელი მისამართი წარმოიშობა როგორც რგII2-ის შიგთავსის და წანაცვლების ჯამი.

როცა დნთ=111, შემსრულებელი მისამართი წარმოიშობა ექვსთანრიგა რგB3 შიგთავსის და წანაცვლების გაერთიანებით.

უნდა აღინიშნოს, რომ დამისამართების ირიბი და ფარდობითი სახეების გარეშე შესაძლებელია მივმართოთ **ოდმ1**-ის მეხსიერების მხოლოდ 1024 უჯრედს.

საბოროტო კომპიუტერში „ორბიტა“ გამოიყენება **ოდმ**-ის ბლოკებთან პირდაპირი დამისამართება და ფარდობითი დამისამართება ხანგრძლივდროიანი დამხსომებული მოწყობილობის ბლოკებთან. ფარდობითი დამისამართების რეალიზაციისთვის გამოიყენება რეგისტრები, რომლებიც იწოდებიან კვადრატების გადამრთველებად. შესაბამისად, არსებობენ ბრძანებათა კვადრატების გადამრთველები, გაერთიანებული ბრძანებათა მთვლელთან და რიცხვის კვადრატების გადამრთველი (**რკვ**).

საბოროტო კომპიუტერში **C-100** გამოიყენება პირდაპირი, ფარდობითი და ირიბი დამისამართება, როგორც ბრძანებების, ისე მიკრობრძანებების დონეზე. გარდა ამისა გამოიყენება რეგისტრული დამისამართებადა აიღება არა მეხსიერების უჯრედის მისამართი, არამედ პროცესორის რეგისტრებიდან ერთ-ერთი რეგისტრის მისამართი.

ლექცია 8

საბორტო კომპიუტერების ინტერფეისული მოწყობილობები

8.1. ინტერფეისების დანიშნულება, კლასიფიკაცია და ძირითადი მახასიათებლები

საავიაციო რადიოელექტრონულ კომპლექსების შემადგენლობაში საბორტო კომპიუტერის მუშაობის არსებით განსაკუთრებულობას წარმოადგენს კომპლექსის სისტემებსა და საბორტო კომპიუტერს შორის ინტენსიური საინფორმაციო გაცვლა. ამ სისტემების უმრავლესობისთვის ინფორმაციის განახლების დრო გაცილებით მეტია ვიდრე ბრძანებების შესრულების დრო ან უფრო მეტიც, არადიდი პროგრამების შესრულების დრო. ამიტომ ინფორმაციის გაცვლის პროცედურების მართვა საბორტო კომპიუტერის მხრიდან არაა მიზანშეწონილი, იმდენად რამდენადაც, ეს ფაქტობრივად იწვევს პროცესორის მოცდენას და მმართველი სიგნალების გამომუშავების დაყოვნებას სხვა სისტემებისთვის.

საბორტო კომპიუტერის შემადგენლობაში ინფორმაციის გაცვლის ორგანიზაციისათვის ცალკეული კონსტრუქციების სახით დამუშავებულ იქნა შეტანა-გამოტანის მოწყობილობები. პირველი გამოშვების შემტან-გამომტან მოწყობილობებში შედიოდა დიდი რაოდენობის ანალოგურ-ციფრული და ციფრულ-ანალოგური გარდამქმნელები, იმდენად რამდენადაც კომპლექსის ელემენტები მუშაობდნენ ანალოგურ სიგნალებთან. ამასთან ცალკეული სისტემების (აბონენტების) ახალი ერთობლიობა მოითხოვდა თავისი შეტანა-გამოტანის მოწყობილობების დამუშავებას, რადგან ანალოგური სიგნალების გაცვლის საშუალებების სტანდარტიზაცია და უნიფიკაცია გაძნელებული იყო.

ისეთი სისტემების დამუშავების შესაძლებლობამ, რომლებიც იყენებენ მონაცემების ციფრულ დამუშავებას, ასევე მიკროპროცესორების განვითარებამ და მათმა დანერგვამ საბორტო სისტემებში შესაძლებელი გახდა გაცვლის საბორტო საშუალებების სტანდარტიზაცია. განსაკუთრებით ეს განაპირობა მიკროპროცესორული

სისტემების ელემენტებს შორის კავშირის სალტური ორგანიზაციის განვითარებამ, ასევე ცნების გაჩენამ „ინტერფეისი“.

ტერმინი „ინტერფეისი“ აღქმული უნდა იქნეს როგორც სიტყვა „შეუღლების“ სინონიმი. სტანდარტული ინტერფეისის ცნების ქვეშ იგულისხმება უნიფიცირებული აპარატული, პროგრამული და კონსტრუქციული საშუალებების ერთობლიობა, რომელიც აუცილებელია ავტომატურ სისტემებში სხვადასხვა ფუნქციური ელემენტების ურთიერთქმედების რეალიზაციისა და ინფორმაციის შეკრებისა და დამუშავებისათვის იმ პირობებში, რომლებიც გაწერილია სტანდარტის მიხედვით და მიმართულია ინფორმაციული, ელექტრული და კონსტრუქციული შეთავსების უზრუნველყოფისათვის.

ინფორმაციული შეთავსებადობა - ეს არის სისტემის ფუნქციური ელემენტების ურთიერთქმედების შეთანხმებულობა ლოგიკური პირობების ერთობლიობის შესაბამისად, რომლებიც განსაზღვრავენ:

- სალტების უნიფიცირებული ნაკრების სტრუქტურას და შემადგენლობას;
- პროცედურების ნაკრებს მათი შესრულების თანმიმდევრობისა და ურთიერთქმედების რეალიზაციისათვის სხვადასხვა რეჟიმებში სისტემების ფუნქციონირების შემთხვევაში;
- კოდირების და მონაცემების ფორმატებს, ბრძანებების, სამისამართო და მდგომარეობის ინფორმაციის ხერხებს;
- მმართველ სიგნალებს შორის დროით დამოკიდებულებებს მათი ფორმისა და ურთიერთქმედების შემოსაზღვრულობის გათვალისწინებით.

ინფორმაციული შეთავსებადობის პირობები განსაზღვრავენ სქემოტექნიკური მოწყობილობების მოცულობას და სირთულეს, პროგრამულ უზრუნველყოფას, მათ ძირითად ტექნიკურ-ეკონომიკურ მაჩვენებლებს, გატარების უნარს და ინტერფეისის საიმედოობას, ასევე აპარატულ დანახარჯებს შეუღლების მოწყობილობაზე.

ელექტრული შეთავსებადობა - ეს არის ელექტრული სიგნალების სტატიკური და დინამიკური პარამეტრების შეთანხმება სალტების სისტემაში ინტერფეისული მოწყობილობების სივრცულ განლაგებაზე შეზღუდვების გათვალისწინებით და მიმღებ-გადამცემი ელემენტების ტექნიკური რეალიზაციის უზრუნველყოფით.

ელექტრული შეთავსებადობის პირობები განსაზღვრავს:

- მიმღებ-გადამცემი ელემენტების ტიპს;

- თანაფარდობას სიგნალების ლოგიკურ და ელექტრულ მდგომარეობებს შორის და მათი ცვლილების საზღვრებს;

- მიმღებ-გადამცემი ელემენტების დატვირთვისუნარიანობის კოეფიციენტებს, მოწყობილობებში ხაზის დასაშვები ტევადობისა და რეზისტორული დატვირთვის მნიშვნელობებს და ა.შ.

ელექტრული შეთავსებადობის პირობები გავლენას ახდენს მონაცემების გაცვლის სიჩქარეზე, მოწყობილობათა განლაგების კონფიგურაციაზე და მათ შორის მანძილზე, მისაერთებელი მოწყობილობების ზღვრულად დასაშვებ რიცხვზე და დაბრკოლება-დაცულობაზე.

კონსტრუქციული შეთავსებადობა - ეს არის ინტერფეისის კონსტრუქციული ელემენტების შეთავსებადობა ტექნიკური და ელექტრული შეერთებების უზრუნველყოფისთვის, სქემური ელემენტების, ბლოკების და მოწყობილობების შეცვლისათვის.

კონსტრუქციული შეთავსებადობის პირობები განსაზღვრავენ გარე და შიგა შესაერთებელი ელემენტების (გასართების) ტიპებს, ნაბეჭდი ფირფიტების, მოდულების (შეცვლის ტიპური ელემენტების), ბლოკების, დგარების და ა.შ. კონსტრუქციას.

ინტერფეისის კავშირების შემადგენელ ფიზიკურ ელემენტებს წარმოადგენს ელექტრული წრედები, რომლებიც ინტერფეისის ხაზებად იწოდებიან. ხაზების ნაწილი, რომლებიც დაჯგუფებულია ფუნქციური დანიშნულების მიხედვით იწოდება სალტედ, ხოლო ხაზების მთელი ერთობლიობა კი მაგისტრალად. ინტერფეისის სალტეების სისტემაში პირობითად შეიძლება გამოვყოთ ორი მაგისტრალი: ინფორმაციული მაგისტრალი და მართვის მაგისტრალი.

ინფორმაციული მაგისტრალით გადაიცემა მონაცემების, მისამართების, ბრძანებების კოდები და მოწყობილობათა მდგომარეობები. ანალოგიური სახელწოდებები ენიჭებათ ინტერფეისი შესაბამის სალტეებს.

ინფორმაციული არხის მართვის მაგისტრალი უზრუნველყოფს ინფორმაციის გაცვლის ორგანიზაციას, მმართველი სიგნალების აუცილებელი ნაკრების ფორმირებას ინტერფეისის მუშაობის პროტოკოლის შესაბამისად.

ინტერფეისების კლასიფიკაცია სტანდარტის შესაბამისად მოცემულია ცხრილში 8.1.

საკლასიფიკაციო ნიშანი	ინტერფეისების ტიპები
შეერთების ხერხების მიხედვით	მაგისტრალური; რადიალური; ჯაჭვისებრი; შერეული
გადაცემის ხერხების მიხედვით	პარალელური; მიმდევრობითი; მიმდევრობით-პარალელური
გაცვლის პრინციპის მიხედვით	ასინქრონული, სინქრონული
გადაცემის რეჟიმის მიხედვით	ორმხრივი ერთდროულად; ორმხრივი რიგითობით; ერთმხრივი

მაგისტრალური ინტერფეისის რეალიზაციისას მოწყობილობებს შორის ინფორმაციის გაცვლა ხორციელდება ან ინტერფეისის ერთი კონტროლერით, ან სისტემის ელემენტებიდან ერთით, რომელიც იწოდება ცენტრალურად.

რადიალურ ინტერფეისში აუცილებელია რომ ელემენტებიდან ერთი ელემენტი იმყოფებოდეს ცენტრში და იგი ახორციელებს ინფორმაციის განაწილებას სახვა ელემენტებს შორის ინდივიდუალურად თითოეული ელემენტისთვის კავშირების ხაზების საშუალებით.

ჯაჭვური ინტერფეისი ჩვეულებრივ გამოიყენება იქ, სადაც მოწყობილობების მიერ ფორმირებული ინფორმაცია უნდა გადადიოდეს მიმდევრობით მოწყობილობიდან მოწყობილობისაკენ.

შერეული ინტერფეისი ახორციელებს სისტემის ელემენტების ორი ან უფრო მეტი ვარიანტების შეერთებას.

ინფორმაციის გადაცემის ხერხი განსაზღვრავს ორობითი კოდის ერთდროულად გადასაცემი თანრიგების რაოდენობას. მიმდევრობითი გადაცემისას საინფორმაციო სალტე შეიცავს ერთ ხაზს, რომლითაც თანრიგები მიმდევრობით ერთმანეთის მიყოლებით გადაიცემიან სიგნალის წყაროდან მიმღებისკენ, პარალელური ინტერფეისის დროს ხორციელდება ერთდროულად მანქანური სიტყვის ყველა თანრიგის გადაცემა. თუ ხდება სიტყვის ნაწილების გადაცემა, მაშინ ინტერფეისი იწოდება მიმდევრობით-პარალელურად.

ასინქრონული ინტერფეისი შეიცავს მოწყობილობებს, რომლებიც უზრუნველყოფენ ინფორმაციის გაცვლის შესაძლებლობას რეალური სწრაფქმედებით ინფორმაციის გადამცემი და მიმღები მოწყობილობებისათვის. ასეთ შემთხვევაში სიგნალის წყარო

აფორმირებს სიგნალს (მიმართვის სიგნალს) ინფორმაციის გადაცემის ნებართვის თაობაზე და იმ მოწყობილობების მისამართს, რომელიც წარმოადგენს ინფორმაციის მიმღებს. მოწყობილობა, რომელიც იდენტიფიცირებას გაუკეთებს მისამართს, განსაზღვრავს ინფორმაციის გაცვლის შესაძლებლობას და აფორმირებს შესაბამის ნებისდამრთველ სიგნალს ინფორმაციის გადაცემაზე.

ამ სიგნალის მიღების შემდეგ ინფორმაციის წყარო გასცემს ინფორმაციულ სალტეზე გადასაცემი შეტყობინობის კოდს და გადაცემის დამთავრებამდე იჭერს მას. გადაცემის დამთავრების შემდეგ ინფორმაციის წყარო აფორმირებს შესაბამის სიგნალს „გადაცემა დასრულებულია“ და ელოდება სიგნალს „მიღება დასრულებულია“ ინფორმაციის მიმღებიდან. ამ სიგნალის მიღების შემდეგ ინფორმაციის წყარო ათავისუფლებს მონაცემების სალტეს, სინქრონული გაცვლის შესრულებას შეესაბამება სისტემის ყველაზე უფრო ნელა მომუშავე ელემენტის სწრაფქმედებას. ინფორმაციის წყარო გაცვლის ორგანიზაციის დროს გასცემს მისამართს და იმავდროულად (ან ფიქსირებული დაყოვნებით) მონაცემებს. მისამართის და მონაცემების გაცემის დრო ასევე შემოსაზღვრულია და მუდმივია ყველა მოწყობილობისთვის. სინქრონული ხერხის ღირსებაა გაცვლის ალგორითმის რეალიზაციის სიმარტივე. ნაკლოვან მხარეს კი წარმოადგენს დაბალი სწრაფქმედება სისტემის იმ ელემენტების არსებობისას, რომლებსაც გააჩნიათ არსებითად სხვადასხვა სწრაფქმედება.

ინფორმაციის გაცვლის ორგანიზაციისათვის ინტერფეისული მაგისტრალის შემადგენლობაში უნდა იყოს სამი სახის სალტე:

- მონაცემების სალტე - საინფორმაციო შეტყობინების გადაცემისათვის;
- მისამართის სალტე - მიმღებისა და წყაროს მისამართების გადაცემისათვის;
- მართვის სალტე სიგნალების გადაცემისათვის, რომლებიც უზრუნველყოფენ გაცვლის პროტოკოლს.

ისეთ სისტემებში, რომელთაც არ წაეყენებათ მკაცრი მოთხოვნები სწრაფქმედებასთან მიმართებაში, მისამართისა და მონაცემების სალტეების შეთავსება შეიძლება მოხდეს აპარატული დანახარჯების შემცირების თვალსაზრისით. ამავე მიზნით ინფორმაციის გაცვლა ერთი მოწყობილობიდან მეორისკენ და პირიქით, ხორციელდება ერთი და იგივე ხაზებით, რომლებიც ასეთ შემთხვევაში იწოდებიან ორმიმართულებიან ხაზებად.

D-15D ტიპის საბორტო კომპიუტერში კომპიუტერის ბლოკებსა და შეტანა-გამოტანის მოწყობილობას შორის ინფორმაციის გაცვლა ძირითადად ხორციელდება პარალელური კოდებით. ამიტომ ბლოკშიგა, ბლოკებსშორისი მაგისტრალები შესრულებულია ერთმიმართულებიანი პარალელური მაგისტრალების სახით. ბლოკებსშორისი შეერთებების კონსტრუქციულ საშუალებებად გამოიყენება ძირითადად გამტარების ხვეული წყვილი.

საბორტო კომპიუტერის და შეტანა-გამოტანის მოწყობილობის შეუღლებისათვის გამოიყენება ორი 16 თანრიგიანი მაგისტრალი: K_1 (შეტანა-გამოტანის მოწყობილობაში პარალელური კოდის გადაცემისათვის) და მაგისტრალი K_2 (შეტანა-გამომტან მოწყობილობიდან პარალელური კოდის მიღებისათვის). მითითებული მაგისტრალები წარმოადგენენ მისამართისა და მონაცემების ინფორმაციულ შეთავსებულ სალტებს. გარდა ერთმიმართულებიანი ინფორმაციული მაგისტრალებისა საბორტო კომპიუტერს გააჩნია მართვის ორი ერთმიმართულებიანი მაგისტრალი.

ამრიგად საბორტო კომპიუტერ A-15A ინტერფეისში გამოიყენება ერთმიმართულებიანი პარალელური მაგისტრალები - შესრულებული მრავალძარღვა კაბელების სახით შემდგარი ხვეული წყვილების გამტარებისაგან. ასეთი გადაწყვეტა უზრუნველყოფს მაგისტრალების მაღალგამტარიანობის უნარს.

საბორტო ინფორმაციული სისტემების ინტერფეისები უზრუნველყოფენ მათი შემადგენელი ნაწილების ქვესისტემების, ბლოკების, მოდულების ურთიერთქმედებას ერთმანეთს შორის და სხვა საბორტო დანადგარებთან. საბორტო ინფორმაციული სისტემების დაპროექტების ამოცანა სხვა დანარჩენებთან ერთად ითვალისწინებს ამ ინტერფეისების არჩევის ან დაპროექტების ოპერაციებს.

ვიწრო გაგებით ინტერფეისს უწოდებენ სქემური საშუალებების ერთობლიობას, რომლებიც უზრუნველყოფენ სისტემის შემადგენელი ნაწილების ურთიერთქმედებას. უფრო ფართო გაგებით ინტერფეისში იგულისხმება ტექნიკური სისტემების კომპონენტების ლოგიკური და ფიზიკური პრინციპების ურთიერთქმედების ერთობლიობა, ე.ი. იგულისხმება წესების, ალგორითმების და დროითი თანაფარდობების ერთობლიობა ამ კომპონენტებს შორის მონაცემების გაცვლის მიხედვით (ლოგიკური ინტერფეისი), ასევე მიერთების საშუალებების ფიზიკური,

მექანიკური და ფუნქციური მახასიათებლები, რომლებიც უზრუნველყოფენ ასეთ ურთიერთქმედებას (ფიზიკური ინტერფეისი).

არსებობს ინტერფეისების ბევრი სახესხვაობა, რომლებიც ერთმანეთისგან განსხვავდებიან ინფორმაციის გაცვლის თავიანთი მახასიათებლებით და პრინციპებით. მათგან ყველაზე უფრო გავრცელებული ინტერფეისები განსაზღვრულია საერთაშორისო, სახელმწიფო და დარგობრივი სტანდარტების მიხედვით.

ინტერფეისები, რომლებიც აერთებს რამდენიმე დამოუკიდებელ კომპიუტერს და რომლებიც ერთად იყენებენ რაღაც საერთო რესურსებს მიღებულია იწოდებოდეს ქსელებად.

ინტერფეისები გამოიყენებიან სხვადასხვა სტრუქტურულ დონეებზე:

- სისტემის ელექტრონული ბლოკების შიგნით ფუნქციური მოწყობილობებისა და მოდულების შეერთებისათვის;
- სისტემის შიგნით ბლოკების ერთმანეთს შორის შეერთებისათვის;
- უბრალო მარტივი გადამწოდების სისტემასთან შეერთებისათვის;
- ინტელექტუალური გადამწოდების სისტემათა შეერთებისათვის ციფრული გამოსასვლელით;
- საბორტო დანადგარების კომპლექსებში სისტემებს შორის ურთიერთქმედების უზრუნველყოფისათვის.

თითოეულ დონეზე გამოიყენება თავისი ინტერფეისი, რომელიც ოპტიმალურადაა მორგებული მოცემული დონის დამახასიათებელი ამოცანების გადაწყვეტისათვის.

ელექტრონული ბლოკების შიგნით დამუშავებული უფლებამოსილია გამოიყენოს ნებისმიერი ინტერფეისი, მათ შორის სპეციალურად დაპროექტებული მოცემული კონკრეტული გამოყენებისათვის. თუმცა პრაქტიკაში მხოლოდ ანალოგურ მოწყობილობებს აერთებენ ერთმანეთთან ნებისმიერი სახით, მაშინ როცა ციფრულ მოწყობილობების ბლოკების (პროცესორი, მეხსიერება, შეტანა-გამოტანის მოწყობილობები) შეერთებებისას, როგორც წესი იყენებენ სტანდარტულ ინტერფეისს. ეს საშუალებას იძლევა გამოყენებულ იქნეს წარმოების მიერ გამოშვებული ინტერფეისის კონტროლერების მიკროსქემები, ასევე დამუშავებული პროგრამული და აპარატული ტექნოლოგიური საშუალებები. იმდენად რამდენადაც არსებობს სტანდარტული

ინტერფეისების ფართო სპექტრი ოპტიმიზირებული სხვადასხვა ამოცანებისათვის, ამდენად მათი რიცხვიდან შესაძლებელია აირჩეს ინტერფეისი, რომელიც პრაქტიკულად ოპტიმალური იქნება მოცემული შემთხვევისათვის. ხშირად ელექტრონული ბლოკის შიგნით გამოყენებული ინტერფეისი განისაზღვრება არჩეული პროცესორის ტიპის მიხედვით. ჩვეულებრივ ბლოკშიგნითა ინტერფეისად გვევლინება რაღაც მაგისტრალური პარალელური სალტე.

რაც შეეხება საბორტო ინფორმაციულ სისტემებში გამოყენებულ ინტერფეისებს, მათი არჩევის შესაძლებლობა არ არსებობს, რამეთუ მათი გადამწოდების სისტემასთან მიერთება თხოულობს სპეციალურ მოთხოვნებს ამ გადამწოდების თავისებურებათა გათვალისწინებით და ასევე საბორტო მოწყობილობების სხვა სისტემების გათვალისწინებით.

საბორტო ინფორმაციულ სისტემის დამმუშავებელი ჩვეულებრივ მოკლებულია ინტერფეისის არჩევის შესაძლებლობას და იგი იძულებულია გამოიყენოს სისტემაში ის ინტერფეისი, რომელსაც მოითხოვს მოცემული გადამწოდი.

ინტერფეისის სახეები, რომლებიც შესაძლებელია გამოყენებულ იქნეს საჰაერო ხომალდის (საფრენი აპარატის) ბორტზე თხოულობს სპეციალური მოთხოვნების დაკმაყოფილებას, რასაც ვერ აკმაყოფილებენ მიწისზედა ელექტრონული ტექნიკის ინტერფეისები.

ბორტზე გამოყენებული ინტერფეისი უნდა აკმაყოფილებდეს შემდეგ მოთხოვნებს:

- დროის რეალურ მასშტაბში მუშაობა;
- მაღალი დაბრკოლებამდგრადობა;
- მდგრადობა მტყუნებების მიმართ - ხაზის გაწყვეტა, კავშირის ხაზში მოკლე

ჩართვა, ინტერფეისთან მიერთებული მოწყობილობის მტყუნებამ არ უნდა გამოიწვიოს ინტერფეისის მთლიანი მტყუნება;

- დეტერმინიზმი - მნიშვნელოვანი ინფორმაციის გადაცემის დაყოვნება უნდა იყოს საწინასწარმეტყველო და მცირე;
- მუშაობა გარემოს არასასურველ პირობებში - აპარატურა, რომელიც გათვლილია ინტერფეისის მუშაობის უზრუნველყოფაზე უნდა იყოს მედეგი ტემპერატურის, ვიბრაციის, დარტყმების და სხვა გარე

ზემოქმედების ფაქტორების მიმართ, რომლებიც წარმოიშობიან საფრენი აპარატის ბორტზე;

- მდგომარეობის კონტროლის შესაძლებლობა;
- მარტივი ტექნიკური მომსახურება;
- მოწყობილობის შემადგენლობისა და კონფიგურაციის ცვლილების სიადვილე, მიმღებ - გადამცემების დამტებამ ან გამოკლებამ არ უნდა გამოიწვიოს სხვა მიმღებ-გადამცემებში არსებითი ცვლილებები ამ ინტეფეისის გამოყენების პირობებში.

ამასთან დაკავშირებით საბორტო სისტემების შეუღლებისათვის გამოიყენება სპეციალური ინტერფეისები, რომლებიც დადგენილია საავიაციო სტანდარტებით. ძირითადი სტანდარტების სახით გამოიყენებიან ARINC - 429 სამოქალაქო საჰაერო ხომალდებისათვის და MIL-155B სამხედრო საჰაერო ხომალდებისათვის, თუმცა ახლა ინტენსიურად გამოიყენება ახალი სტანდარტები - სამხედრო თვითმფრინავებში სტანდარტი STANAC 3910, ხოლო სამოქალაქო ავიაციის საჰაერო ხომალდებისათვის დამუშავდა ახალი ინტერფეისები - ARINC-629 მაგისტრალური საჰაერო ხომალდებისათვის, ხოლო ABCB მსუბუქი საფრენი აპარატებისათვის.

საბორტო ინტერფეისებისათვის წაყენებული ძირითადი მოთხოვნა გამტარუნარიანობის გაზრდაა. გადასაცემი ინფორმაციის მოცულობა მკვეთრად გაიზარდა, რაც გამოწვეულია შემდეგი გარამოებებით:

- 1) მონაცემები, რომლებიც ადრე გადაიცემოდნენ ანალოგური სიგნალებით (გამოსახულება, ხმა, მონაცემები საფრენო აპარატის სენსორებიდან) ახლა გადაიცემა ციფრული სახით;
- 2) შეიცვალა საბორტო მოწყობილობის სტრუქტურა, მოხდა დაყოფა პირველადი ინფორმაციის სისტემებად და სისტემებად, რომლებიც ახორციელებენ პირველადი ინფორმაციის დამუშავებას, ინფორმაციის სხვადასხვა წყაროებიდან მონაცემების შერევას. გამომდინარე აქედან საჭირო გახდა მონაცემების დიდი ნაკადის გადაცემა პირველადი ინფორმაციის სისტემებიდან დამუშავების სისტემებისადმი, სადაც ხდება მათი დამუშავება.

3) წარმოიშვა ახალი ამოცანები, რომლებიც მოითხოვენ ინფორმაციის დიდი ნაკადის გადაცემას დროის რეალურ მასშტაბში - სიგნალების ციფრული დამუშავება, მიზნების ამოცნობა და ა.შ.

ყველაფერმა ამან მოითხოვა მონაცემების გადაცემის სიჩქარის გაზრდა 0,125-2 მეგა ბიტი/წმ-დან 100 მეგაბიტი/წმ - 1 გიგაბიტი/წმ-მდე, ამასთან უზრუნველყოფილი უნდა იქნეს ინფორმაციის დროული მიღება 1 მილიწამზე ნაკლები დაყოვნებით. უახლოეს საფრენ აპარატებზე უკვე ინერგება მაღალსიჩქარიანი ინტერფეისები/ქსელები, რომლებიც აკმაყოფილებენ ამ მოთხოვნებს. სალტე High-Speed Data Bus გამოიყენება F-22 გამანადგურებელზე და RAH-66 Comanch და შვეულმფრენზე. SAE (Society of Automotive Engineers) შექმნილია AS 4074 სტანდარტი. ავიაციაში ფართო გამოყენებას პოულობენ Fibre Channe, Etherlet/AFDX.

საფრენ აპარატზე ერთდროულად გამოიყენება რამდენიმე სხვადასხვაგვარი ინტერფეისი: შიგა ელექტრონული ბლოკების პარალელური სალტეები, მარტივი თანმიმდევრობითი ინტერფეისები გადამწოდებთან კავშირისათვის, მულტიპლექსური არხები და ქსელი საბორტო სისტემების ურთიერთქმედებისთვის. ასეთი სხვადასხვაგვარობა იწვევს საშუალებების მნიშვნელოვან დანახარჯებს, დაპროექტებისათვის ძალისხმევას და დროის დანახარჯებს. თუ მოხდებოდა ბორტზე არსებული ყველა მოწყობილობის გაერთიანება ერთი მთლიანი ინტერფეისის დახმარებით, მაშინ მივიღებდით მნიშვნელოვან ტექნიკურ - ეკონომიურ ეფექტს. ამიტომ ამ ბოლო ხანს საავიაციო მოწყობილობების დამპროექტებლების ძალისხმევა მიმართულია ასეთი უნივერსალური ინტერფეისი შექმნისაკენ. იმისათვის, რომ ამ ინტერფეისმა დააკმაყოფილოს სხვადასხვა ჯგუფების მოწყობილობების მოთხოვნები, ეს ინტერფეისი უნდა იყოს მოქნილი, მისი მახასიათებლები უნდა იცვლებოდნენ ფართო საზღვრებში გაცვლის ზოგადი პრინციპების შენარჩუნებით. ასეთი ინტერფეისები იწოდებიან მასშტაბისებურ ინტერფეისებად.

8.2. ინტერფეისის სახეები

ცალკეული ფუნქციური ბლოკების ერთმანეთს შორის შეერთება ხორციელდება ინტერფეისის ხაზებით. ინტერფეისები შეიძლება შეიცავდეს სამისამართო ხაზებს, მონაცემების ხაზებს, მართვის, სინქრონიზაციის, კონტროლის და ა.შ. ხაზებს. ხაზების შემადგენლობა და რაოდენობა განსაზღვრავს მონაცემთა გადაცემების შესაძლო ხერხებს. ინტერფეისის ხაზები შეიძლება გაერთიანდეს ჯგუფებში ოპერაციებიდან ერთ-ერთი ოპერაციის შესასრულებლად მონაცემთა გადაცემის პროცესში. ეს ჯგუფები იწოდებიან ინტერფეისის სალტეებად.

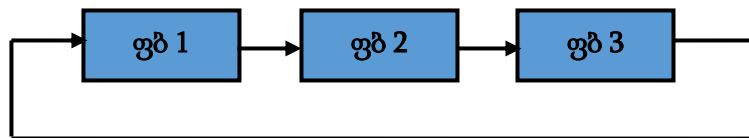
- ინტერფეისის მართვის ხერხი (ცენტრალიზებული, დეცენტრალიზებული);
- აქტიური მოწყობილობის რაოდენობა;
- შეერთების ხერხი (მაგისტრალური, ჯაჭვური, რადიალური, კომბინირებული);
- ინფორმაციის გადაცემის ხერხი (პარალელური, ნახევრადდუპლექსური, დუპლექსური, მულტიპლექსური);
- ინფორმაციის გაცვლის პრინციპი (ასინქრონული, სინქრონული);
- სალტეების და სიგნალების ნომენკლატურა (მათ შორის ხაზების საერთო რაოდენობა და ხაზების რაოდენობა მონაცემების გადაცემისათვის);
- გადაცემის მარშრუტი;
- კომუტაციის ხერხი (წრედების კომუტაცია ან მაკეტების კომუტაცია);
- გადაგზავნის ზომა;
- მისამართების რაოდენობა;
- ბრძანებების რაოდენობა;
- სწრაფქმედება/გამტარუნარიანობა;
- კავშირის ხაზის სიგრძე;
- კავშირის ხაზის ტიპი (გადაცემის გარემო);
- მისაერთებელი მოწყობილობების რაოდენობა.

ცენტრალიზებული მართვის შემთხვევაში ინფორმაციის ყველა გადაცემა ერთი ფუნქციური ბლოკიდან მეორისადმი წარმოებს მთავარი მოწყობილობის მართვის ან მისი უშუალო მონაწილეობის შედეგად. ასეთი მოწყობილობა იწოდება აქტიურ მოწყობილობად ან კონტროლერად. აქტიური შეიძლება იყოს რამდენიმე, მაშინ ისინი მართვას ახორციელებენ რიგრიგობით. გაცვლის განხორციელების წარმართვის შესაძლებლობა

შეიძლება მიენიჭოს აქტიურ მოწყობილობებს ან ცენტრალური პროცესორით ან განსაკუთრებული მოწყობილობებით, რომელსაც არბიტრი ეწოდება. დეცენტრალიზებული მართვის შემთხვევაში ყველა ფუნქციური ბლოკი თანაბარუფლებიანია და გაცვლას ახორციელებენ დამოუკიდებლად.

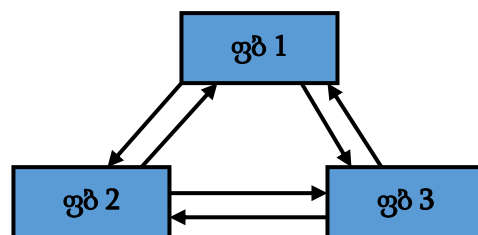
შეერთების ხერხის მიხედვით ფუნქციურ ბლოკებს განასხვავებენ ჯაჭვური, რადიალური და მაგისტრალური შეერთებების მოხედვით.

ჯაჭვური შეერთების დროს თითოეული წყვილი წყარო-მიმღები შეერთებულია წყვილურად, ხოლო მართვის ფუნქციები განაწილებულია ამ ფუნქციურ ბლოკებს შორის. ნახ.8.1-ზე ნაჩვენებია ჯაჭვური შეერთების სხვადასხვაგვარობის ერთ-ერთი სახე - რგოლური შეერთება, რომელშიც უკანასკნელი ბლოკის გამოსასვლელი ჯაჭვში შეერთებულია პირველის შესასვლელთან.



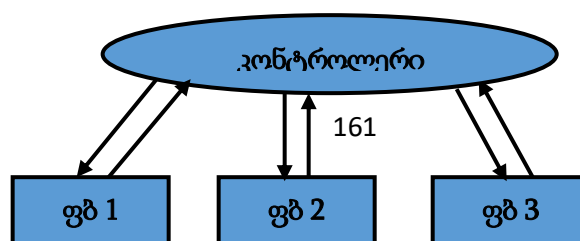
ნახ.8.1. ჯაჭვური შეერთება

რადიალური შეერთების დროს (ნახ.8.2) ერთი ფუნქციური ბლოკი ერთდროულად შეერთებულია რამდენიმე ბლოკთან, ამასთან თითოეულთან შეერთებულია ცალკეული



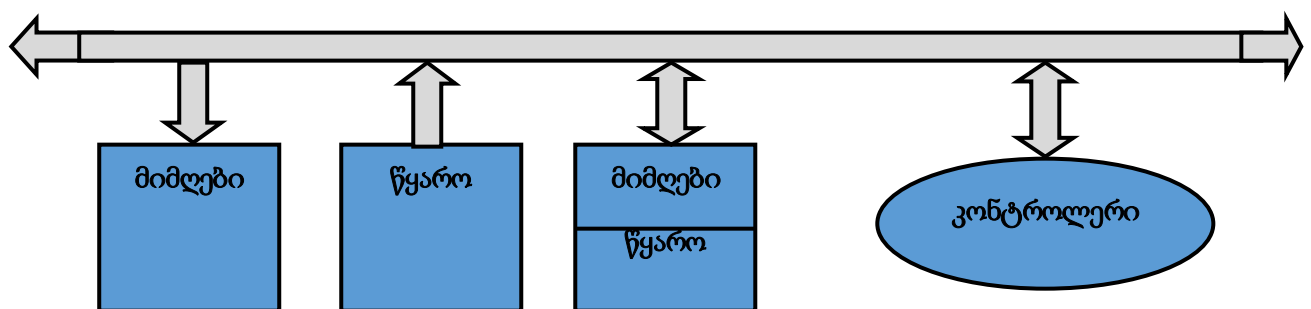
ნახ.8.2. რადიალური შეერთება

დამოუკიდებელი ხაზის საშუალებით. ასეთი სახის კავშირის ერთ-ერთ სახეს წარმოადგენს სტრუქტურა ცენტრალური მოწყობილობით - კონტროლერით (კონცეტრატორით), რომელთანაც თითოეული წყვილი წყარო წყარო-მიმღები დაკავშირებულია გაცვლის ინდივიდუალური არხით (ნახ.8.3).



კონტროლერის მართვის შედეგად მიმდინარეობს მონაცემების გაცვლა თითოეულ მოწყობილობასა და კონტროლერს შორის. ინფორმაციის გადაცემის სიჩქარე შეიძლება წარმოებდეს, როგორც კონტროლერის ინიციატივის საფუძველზე, ასევე ფუნქციური ბლოკის ინიციატივის საფუძველზეც. იმ შემთხვევაში როცა მიმართვა მოვა რამდენიმე მოწყობილობისგან ერთდროულად კონტროლერი მოემსახურება მათ ცალ-ცალკე დადგენილი პრიორიტეტის მიხედვით. რადიალური სტრუქტურის ნაკლოვანი მხარეა-შემადგენელი ხაზების დიდი სიგრძე, უპირატესობა მდგომარეობს იმაში, რომ მარტივია შეერთება დამისამართება მოთხოვნილი ფუნქციური ბლოკისა.

მაგისტრალური შეერთების დროს (ნახ.8.4) კავშირის ერთ საერთო სისტემასთან მაგისტრალთან მიერთებულია ინფორმაციის ყველა წყარო და მიმღებები, ასევე კონტროლერიც. დროის თითოეულ მომენტში მხოლოდ ერთი წყვილი წყარო - მიმღები შეუძლიათ გაცვალონ ერთმანეთს შორის ინფორმაცია საერთო მაგისტრალის საშუალებით. კავშირი ფუნქციურ ბლოკებს შორის ხორციელდება საერთო მაგისტრალის საშუალებით. კავშირი ფუნქციურ ბლოკებს შორის დგინდება თვითონ ბლოკების ინიციატივის მიხედვით ან კონტროლერის ინიციატივით.



ნახ.8.4. მაგისტრალური შეერთება

პირველ შემთხვევაში მოწყობილობა, რომელიც თხოულობს კავშირს, აგზავნის მიმართვას მომსახურებაზე. თუ მაგისტრალი თავისუფალია და მოწყობილობა

ღებულობს თანხმობას კავშირის განხორციელებისათვის, მაშინ ის კონტროლერის მართვის შედეგად ამყარებს კავშირს ინფორმაციის მიმღებთან და ახორციელებს მონაცემთა გადაცემას.

კონტროლერი არ წარმოადგენს მაგისტრალური შეერთების აუცილებელ კომპონენტს, ის შეიძლება არ არსებობდეს, ასეთ შემთხვევაში მისი ფუნქციები განაწილებულია ფუნქციურ ბლოკებს შორის.

მაგისტრალური კავშირის ნაკლოვანებას წარმოადგენს მისი დაბალი საიმედოობა რადიალურ კავშირთან შედარებით და უფრო რთული საშუალებები კავშირის დამყარებისთვის. თუმცა მოწყობილობაზე დაბალი დანახარჯები (კაბელები, გასართები) მას ხდის ურო გამოყენებად შეერთებას მოწყობილობების დიდი რიცხვის შემთხვევაში.

ნაკლოვანების აღმოფხვრისთვის, რომელიც ნიშანდობლივია რადიალური და მაგისტრალური შეერთებისათვის ხშირად მიმართავენ კომბინირებული შეერთების ხერხს, რომლის დროსაც ინტერფეისი შეიცავს როგორც ინდივიდუალურ, ისე კოლექტიურ სალტებს. ეს კი იძლევა იმის საშუალებას, რომ მიღებულ იქნეს მისაღები მახასიათებლები სწრაფქმედებისა და კაბელების სიგრძის მიხედვით.

8.3. ინფორმაციის გადაცემის ხერხები, რეჟიმები, გაცვლის

პრინციპები და გადაცემის გარემო

ინფორმაციის გადაცემის ძირითადი ხერხები. ინფორმაციის გადაცემის ძირითადი ხერხებია - პარალელური და მიმდევრობითი. გამოიყენებენ ასევე ინფორმაციის გადაცემის კომბინირებულ ხერხს - პარალელურ - მიმდევრობითს.

პარალელური გადაცემის დროს ერთდროულად ხდება ინფორმაციის **m** ბიტის გადაცემა **m** რაოდენობის ინფორმაციული ხაზების საშუალებით. ასეთ შემთხვევაში ინფორმაციის გადაცემის დრო მინიმალურია.

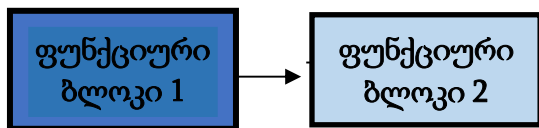
ინფორმაციის პარალელურ-მიმდევრობითი გადაცემის დროს მონაცემები გადაიცემა სიმბოლოების მიხედვით. თითოეული სიმბოლო შედგება **m** ბიტისგან, ყველა

მისი ბიტი გადაიცემა პარალელურად n ხაზების მიხედვით. სიმბოლოები გადაიცემა თანმიმდევრობით, ამიტომ m ბიტის გადაცემისათვის საჭიროა $k=(m/n)$ გადაგზავნა.

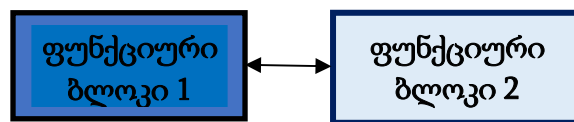
თანმიმდევრობითი გადაცემის შემთხვევაში ყველა ბიტი გადაიცემა თანმიმდევრობით ერთი ხაზის საშუალებით. კავშირების რაოდენობა ასეთი გადაცემის დროს მინიმალურია, გამომდინარე აქედან ასეთი გადაცემის კავშირების რაოდენობა მინიმალურია, შესაბამისად ასეთი ინტერფეისები იაფია და კომპაქტურია, მაგრამ გაზრდილია ინფორმაციის გადაცემის დრო. თანმიმდევრობით ინტერფეისებში მარტივია დაბრკოლებამდგრადობის უზრუნველყოფა.

ინფორმაციის გადაცემის რეჟიმები. განასხვავებენ ინფორმაციის გადაცემის სიმპლექსურ, ნახევრადდუპლექსურ, დუპლექსურ და მულტიპლექსურ რეჟიმებს.

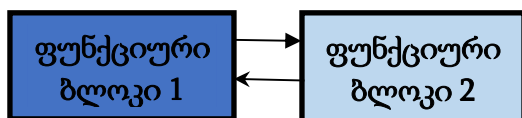
სიმპლექსურ რეჟიმში მხოლოდ ერთ ფუნქციურ ბლოკს ორიდან შეუძლია დროის ნებისმიერ მომენტში ინიცირება გაუკეთოს ინფორმაციის გადაცემა ინტერფეისის მიხედვით (ნახ.8.5).



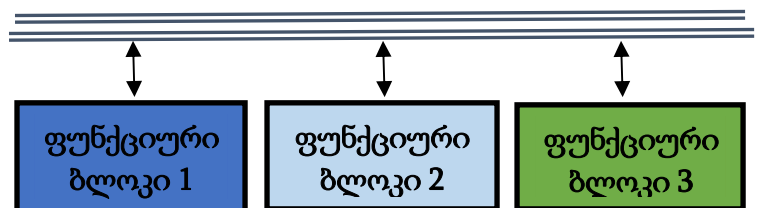
ნახ.8.5



ნახ.8.6



ნახ.8.7



ნახ.8.8

ნახევრადდუპლექსურ რეჟიმში თითოეულ ფუნქციურ ბლოკს ორიდან შეუძლია დაიწყოს ინფორმაციის გადაცემა მეორისადმი, თუ კავშირის ხაზი თავისუფალია (ნახ.8.6).

დუპლექსურ რეჟიმში თითოეულ ფუნქციურ ბლოკს შეუძლია ინფორმაციის გადაცემა მეორისადმი დროის ნებისმიერ მომენტში, თუ ამისათვის გამოიყენებს ინტერფეისის ცალკეულ ხაზებს (ნახ.8.7).

მულტიპლექსურ რეჟიმში დროის თითოეულ მომენტში შესაძლებელია განხორციელდეს კავშირი ფუნქციური ბლოკების ნებისმიერ წყვილს შორის, მხოლოდ ერთი მიმართულებით (ნახ.8.8).

ინფორმაციის გაცვლის პრინციპები. ინფორმაციის გაცვლა შეიძლება იყოს სინქრონული და ასინქრონული. გაცვლის პრინციპის არჩევა განსაზღვრავს არა მარტო გამტარუნარიანობას, არამედ ის უშუალოდ გავლენას ახდენს როგორც გაცვლის არხის ფიზიკურ სიგრძეზე, ისე მოწყობილობათა რაოდენობაზე, რომლებიც შესაძლებელია მიერთებულ იქნეს მასთან.

სინქრონული გადაცემის დროს მონაცემები გადაიცემა მუდმივი მიხედვით დროის მიხედვით. მასინქრონიზერებელი სიგნალები იძლევიან დროის განსაზღვრულ ინტერვალს, რომლის განმავლობაშიც ხდება გაცვლის არხიდან ინფორმაციის ამოკითხვა. სინქრონული სალტე შეიცავს სიგნალიზაციის სიგნალებს, რომლებიც გადაიცემა სალტის მართვის ხაზების საშუალებით, ასევე შეიცავს ფიქსირებულ პროტოკოლს, რომელიც განსაზღვრავს მისამართისა და მონაცემების განლაგებას სინქრონიზაციის სიგნალებთან მიმართებაში. იმდენად რამდენადაც, პრაქტიკულად არავითარი დამატებითი ლოგიკა არაა საჭირო, იმისათვის რომ გადაწყდეს, თუ რაა გასაკეთებელი დროის მომდევნო მომენტში, ეს სალტეები არის სწრაფი და იაფი. სინქრონული გადაცემის დროს უფრო ეფექტურად გამოიყენება გაცვლის არხი. თუმცა მათაც გააჩნიათ ნაკლოვანება. კერძოდ სინქროსიგნალების გადახრის პრობლემის გამო. სინქრონული სალტეები შეუძლებელია იყოს გრძელი. სალტე პროცესორი - მეხსიერება ჩვეულებრივ სინქრონულია.

ასინქრონული გადაცემა ხორციელდება დროის მიხედვით არამუდმივი დროითი ინტერვალის შესაბამისად და ამისათვის ჩვეულებრივ გამოიყენება სტარტ-სტოპური რეჟიმი ინფორმაციის გადაცემისა და პროტოკოლი „ხელჩამორთმევისა“ მონეცემების წყაროსა და მომხმარებელს შორის სალტეზე მმართველი სიგნალებით (მზადყოფნა გასაცვლელად, გაცვლის დაწყება, გაცვლის დასრულება, გადაცემის კონტროლი და ა.შ). გადაცემის ეს პრინციპი საშუალებას იძლევა უფრო მარტივად იქნეს სალტესთან მიერთებული სხვადასხვა მახასიათებლების მქონე მოწყობილობები, ასეთ შემთხვევაში

შესაძლებელია სალტის სიგრძის გაზრდა და არ არის საშიშროება სინქრონიზაციის სიგნალების გადახრის შესახებ. ასინქრონული გადაცემის საუკეთესო დაბრკოლება-მდგრადობა საშუალებას იძლევა ერთმანეთთან დააკავშიროს სხვადასხვა სწრაფქმედების მქონე ფუნქციური ბლოკები. შეტანა-გამოტანის სალტები ჩვეულებრივ ასინქრონულია.

გადაცემის მარშრუტი. გადამცემიდან მიმღებისკენ ინფორმაციის გადაცემის მარშრუტის განსაზღვრის ხერხის მიხედვით განსხვავებენ ინფორმაციის პირდაპირ და ირიბ გადაცემას.

ინფორმაციის პირდაპირი გადაცემის შემთხვევაში გადამცემი ირჩევს გზას, ხოლო მიმღები ახდენს მისდამი გაგზავნილი შეტყობინების ამოცნობას. ამასთან ასეთ შემთხვევაში საჭირო არაა მოხდეს ინფორმაციის გადაცემის გზების მართვა, ამიტომ საინფორმაციო არხში არაა გათვალისწინებული არავითარი ლოგიკა, გარდა ბუფერული მოწყობილობებისა და მოწყობილობების გამორიცხვით, რომლებიც თავის მხრივ ამაღლებენ ინფორმაციის გადაცემის უტყუარობას.

ინფორმაციის ირიბი გადაცემის შემთხვევაში გათვალისწინებულია გადამცემსა და მიმღებს შორის არსებული რამდენიმე ალტერნატიული მარშრუტიდან ერთის ამორჩევის ლოგიკა, ასევე ზოგიერთ შემთხვევაში დასაშვებია ინფორმაციის დამატებითი გარდაქმნა, ასეთ შემთხვევაში ინფორმაციის გადაცემის წრედში გათვალისწინებულია სპეციალური მოწყობილობები ე.წ. მარშრუტიზატორები, რომლებიც მართავენ ინფორმაციის გადაცემის მარშრუტს.

კომუტაციის ხერხი. კომუტაციის ხერხები განსხვავებულია პარალელური სალტების შემთხვევაში. ამჟამად გამოიყენება სალტების ორი სახე: 1. სალტეში ჯაჭვური კომუტაციით, 2. სალტები პაკეტების კომუტაციით, რომლებმაც თავიანთი დასახელებები მიიღეს მონაცემების გადაცემის ქსელებში კომუტაციის ხერხების ანალოგიის შესაბამისად.

სალტზე ჯაჭვური კომუტაციით შემთხვევაში მოწყობილობა თხოულობს ნებართვას სალტზე დასაშვებად, თანხმობის მიღების შემთხვევაში ის ათავსებს სალტზე მისამართს და ინარჩუნებს მას მიმართვის მომსახურების დამთავრებამდე. ასეთ შემთხვევაში მომსახურების დროის დიდი ნაწილი იხარჯება არა სალტზე ოპერაციის შესრულებაზე, არამედ მეხსიერებიდან ამორჩევის ოპერაციასა და სხვა

ოპერაციებზე. ამრიგად, შეგვიძლია დავასკვნათ, რომ სალტებზე ჯაჭვური კომუტაციით ეს დრო უბრალოდ იკარგება.

სალტე პაკეტების კომუტაციით რამდენიმე მოწყობილობის არსებობის შემთხვევაში, რომლებიც მიმართავენ სალტეს, უზრუნველყოფს ინფორმაციის გაცილებით დიდ გამტარუნარიანობას ვიდრე ეს შესაძლებელია ჯაჭვური კომუტაციით სალტის შემთხვევაში. ეს კი მიღებულია წაკითხვა/ჩაწერა ოპერაციების დანაწილებით ორ ლოგიკურ ნაწილად: მიმართვა და პასუხი. მაგალითად, მიმართვის ოპერაცია დანაწილდება ორ ელემენტალურ ოპერაციად, რომლებიც იწოდებიან ტრანზაქციებად: 1. მიმართვის, რომელიც შეიცავს მისამართს, და 2. მეხსიერების პასუხი, რომელიც შეიცავს მონაცემებს. ასეთმა ხერხმა მიიღო დასახელება „ტრანზაქციათა გახლეჩა“. ტრანზაქცია იწოდება გახლეჩილად, იმდენად რამდენადაც ასეთ შემთხვევაში სხვა პაკეტების ან ტრანზაქციების ნებისმიერ რაოდენობას შეუძლია გამოიყენოს სალტე მიმართვისა და პასუხს შორის შუალედში, მანამ სანამ მეხსიერება კითხულობს სიტყვას მითითებული მისამართის მიხედვით. თითოეული ტრანზაქცია მონიშნული უნდა იყოს შესაბამისი სახით, რათა მოწყობილობამ და მეხსიერებამ განასხვავოს რა არის რა. სალტეს ტრანზაქციის გახლეჩით გააჩნია უფრო მაღალი გამტარუნარიანობა, მაგრამ ვინაიდან თითოეული მოწყობილობა იბრძვის სალტისათვის მონაცემების გადასაგზავნად, მეხსიერებამაც უნდა იბრძოდეს სალტისთვის რათა დაიბრუნოს მონაცემები, ამდენად ასეთ სალტეს გააჩნია დიდი დაყოვნება, ვიდრე სალტეს, რომელიც დაკავებულია ტრანზაქციის შესრულებით მთელი დროის განმავლობაში.

გადაგზავნის ზომა. ინფორმაცია გადაცემიდან მიმღებისკენ შესაძლებელია გადაიცეს ერთი სიტყვით ან სიტყვების ჯგუფით (პაკეტებით). სიტყვების ჯგუფის გადაგზავნა უფრო ხანგრძლივია, ვიდრე ერთი სიტყვი გადაგზავნა, თუმცა ზედნადები დანახარჯები მართვაზე და არბიტრაჟზე ჯამში მიიღება ნაკლები.

ინფორმაციის გადაცემის გარემო. გადაცემის გარემო ეს არის ფიზიკური მონაცემების გადაცემის არხის ორგანიზაციისათვის. ელექტრონული ბლოკების შიგნით გადაცემის გარემოს სახით გამოიყენება სპილენძის გამტარები - მავთულების სახით ან ნაბეჭდი გამტარების სახით ნაბეჭდ ფირფიტებზე.

ბლოკებს გარეთ ე.ი. მათ შესაერთებლად იყენებენ სხვადასხვა ტიპის კაბელებს:

1. წყვილ-ნახვევი ეკრანში. 2. კოაქსიალური კაბელი. 3. კავშირის ბოჭკო-ოპტიკური ხაზები.

ნახვევი წყვილი (ბიფილიარი) თავის მხრივ წარმოადგენს ერთმანეთთან შეკავშირებულ ორი სპილენძის გამტარს, რომლებიც მოთავსებულია საერთო მეტალიზებული შემონაქსოვით, რომელიც გამოდის ეკრანის როლში. ეს არის ყველაზე კომპაქტური იაფი კაბელი ყველა არსებული სახის კებელებს შორის.

კოაქსიალური კაბელი წარმოადგენს სპილენძის ძარღვს რომელიც მოთავსებულია ფირფიტოვანი გარსის შიგნით და ყველა ესენი ერთად კი იმყოფება მეტალიზებული ნახვევის შიგნით, რომელიც გამოიყენება მეორე (ნულოვანი) გამტარის სახით. კოაქსიალური კაბელი მძიმე და ძვირია, თუმცა ის საშუალებას იძლევა გადაიცეს სიგნალი ძალზე შორ მანძილზე და უფრო დიდი სიჩქარით, ვიდრე ეს შესაძლებელია წყვილნახვევით.

კავშირის ბოჭკოვან-ოპტიკურ ხაზებში მონაცემები გადაიცემა კაბელის გამჭვირვალე გარემოს შიგნით სინათლის იმპულსების სახით. ამ სახის კაბელს სპილენძიანი გარემოს სახის მქონე კაბელებთან შედარებით გააჩნია მთელი რიგი უპირატესობანი, კერძოდ:

1. არ ხდება პრაქტიკულად სიგნალის მიღება, რაც საშუალებას იძლევა გადაიცეს ის შუალედური გაძლიერების გარეშე 50კმ სიგრძის მანძილამდე.
2. მისი გამოყენების შემთხვევაში გადაცემის სიჩქარე შეადგენს 40 გიგაბიტი/წამში, რაც ორი რიგით უფრო მაღალია, ვიდრე ინფორმაციის გადაცემისათვის გამოყენებული სპილენძიანი კაბელების შემთხვევაში.
3. სიგნალი არ ქმნის სხვა სისტემებისათვის დაბრკოლებებს და თავის მხრივაც არ განიცდის ელექტრომაგნიტური დაბრკოლებების ზეგავლენას.
4. მიღებულია გადაცემის დამალულობა, სხვა სახის გარემოთი ინფორმაციის გადაცემისაგან განსხვავებით ოპტიკური სიგნალის დაჭერა შეუძლებელია კაბელში უშუალოდ ჩართვის გარეშე.
5. მიღწევადია მისი საშუალებით მრავალარხულობის შესაძლებლობა, რაც გულისხმობს იმას რომ ერთი კაბელის საშუალებით შესაძლებელია გადაიცეს 1000 არხზე უფრო მეტი შემჭიდროებული ვიდეოგამოსახულებებით, აუდიოსიგნალებით და სენსორული ინფორმაციით.

6. ბოჭკოვან-ოპტიკურ კავშირის ხაზები ასევე მოქნილი და მტკიცე როგორც კოაქსიალური კაბელი, თუმცა მისგან განსხვავებით უფრო უსაფრთხოა განსაკუთრებულ პირობებში.

ბოჭკოვან-ოპტიკური კაბელის ნაკლოვან მხარეებს მიეკუთვნება ის, რომ ასეთი კაბელი საკმაოდ ძვირია, რთულია დაყენების თვალსაზრისით და ტექნომსახურების მიხედვითაც. ის კარგად გამოყენებადია მხოლოდ ჯაჭვური შეერთებისათვის (გადამცემი-მიმღები), სხვა სახის შეერთების ხერხების შემთხვევაში (გადამცემი-რამდენიმე მიმღები) წამოიჭრება პრობლემები კაბელის გაყვანასთან დაკავშირებით.

უკანასკნელ ხანს ასეთი ტიპის კაბელები სულ უფრო და უფრო ფართო გამოყენებას პოულობენ, როგორც მიწისზედა ისე საბორტო ინტერფეისებში.

ლექცია 9

საბორტო კომპიუტერის შიგა მაგისტრალი

9.1. საბორტო შიგა ინტერფეისის განხორციელებისთვის მაგისტრალის გამოყენება და მისი საშუალებით მოწყობილობებს შორის მონაცემების გაცვლის ხერხები

საბორტო კომპიუტერის შიგა მაგისტრალის სახით, როგორც წესი, გამოყენებულია მაგისტრალი, რომელთანაც მიერთებულია პროცესორი, მეხსიერების მოწყობილობები, ასევე სხვა მოწყობილობები, რომლებიც იწოდებიან გარე მოწყობილობებად (გმ). ჩვეულებრივ ასეთი მაგისტრალი წარმოადგენს პარალელურს და შეიცავს 20-40 ან უფრო მეტ ინტერფეისის ხაზებს, რომელთა საშუალებით ერთდროულად გადაიცემა 8/16/32/64 ბიტი ინფორმაცია. არსებობს მაგისტრალთა საკმაოდ ბევრი ტიპი. საოფისე და საწარმოო კომპიუტერებში ფართოდ გამოიყენება ისეთი ინტერფეისები როგორიცაა ISA, EISA, VME, Q-bus, Multibus II, PCI. ყველა ესენი ან მათი მოდიფიკაციები გამოიყენება საბორტო ციფრულ კომპიუტერებშიც. გარდა ამისა სამხედრო დანართების სახით დამუშავებულია განსაკუთრებული პარალელური ინტერფეისები, მაგალითად, PI-bus და Futurebus. მიუხედავად ინტერფეისთა დიდი სხვადასხვაგვარობისა, ისინი ორგანიზებული არიან მსგავსი პრინციპების საფუძველზე. განვიხილოთ პარალელური მაგისტრალის შემადგენლობა და ფუნქციონირება Q-bus პოპულარული მაგისტრალის მაგალითზე, რომელიც გამოიყენება მთელ რიგ საბორტო გამომთვლელელებში.

ასეთი ტიპის მაგისტრალში ინფორმაციის გაცვლაში მონაწილეობას იღებს ორი მოწყობილობა, რომლებთაგან ერთი აქტიურია, რომელიც ინიციალიზაციას უკეთებს ინფორმაციას მეორე პასიურ მოწყობილობასთან. მაგისტრალზე აქტიურ მოწყობილობას წარმოადგენს პროცესორი, თუმცა გათვალისწინებულია ასევე იმის შესაძლებლობა, რომ აქტიური მოწყობილობის ფუნქციის როლი პროცესორიდან გადაეცეს გარე მოწყობილობებს, ამასთან ამ უკანასკნელთა როლში შეიძლება გამოდიოდნენ სხვა პროცესორები, რაც საშუალებას იძლევა მაგისტრალის ბაზაზე ორგანიზაცია გაუკეთდეს მულტიპროცესორულ სისტემებს.

მეხიერებასთან ან გარე მოწყობილობებთან მიმართვისათვის გამოიყენება სამი პროცედურა:

შეტანა - ამ პროცედურის დროს მონაცემები გადაიცემიან გარე (პასიური) მოწყობილობებიდან აქტიური მოწყობილობისაკენ, ე.ი. ხორციელდება ამოკითხვის ოპერაცია;

გამოტანა - ამ პროცედურის დროს მონაცემები აქტიური მოწყობილობებიდან გადაიცემა პასიური მოწყობილობისკენ, ე.ი. ხდება მონაცემების ჩაწერა მისამართის მიხედვით, რომელიც განსაზღვრულია გაცვლის ციკლის შესრულების დასაწყისში.

შეტანა-პაუზა-გამოტანა, ამ თანმიმდევრული ოპერაციების განხორციელებისას დასწყისში შეიტანება ოპერანდი (სრულდება პროცედურა „შეტანა“), შემდეგ მასზე ხორციელდება არითმეტიკული ან ლოგიკური ოპერაცია (პროცედურა „პაუზა“), რის შემდეგ ოპერაციის შედეგი გამოიტანება იმავე მისამართზე (პროცედურა „გამოტანა“), რომლიდანაც აღებულ იქნა ოპერანდი.

გაცვლა წარმოადგენს ასინქრონულს, ხოლო კავშირი მაგისტრალის საშუალებით ორ მოწყობილობას შორის წარმოადგენს შეკრულს (ჩაკეტილს). ყველა მოწყობილობა მაგისტრალს უერთდება მიმღებებისა და გადამცემების დახმარებით, რომლებიც თავის მხრივ უზრუნველყოფენ სიგნალების გადცემას არხის საშუალებით.

მაგისტრალი უზრუნველყოფს მოწყობილობებს შორის მონაცემების გაცვლის სამ ხერხს:

- პროგრამული რეჟიმი;
- არაპროგრამული (პირდაპირი დაშვების რეჟიმი);
- გარე მოწყობილობებთან მიმართვა (პროგრამის წყვეტის რეჟიმი).

პროგრამული რეჟიმის შემთხვევაში აქტიურ მოწყობილობას წარმოადგენს პროცესორი, ხოლო გაცვლა ხორციელდება მის მიერ შესრულებადი პროგრამის მიერ.

პირდაპირი დაშვების რეჟიმში გარე მოწყობილობა მიმართავს პროცესორს და თავის განკარგულებაში ღებულობს მაგისტრალს, რის შემდეგ პროცესორი გამოირთვება მაგისტრალიდან და გადადის პასიურ მდგომარეობაში, გარე მოწყობილობა ხდება აქტიური და მართავს გაცვლის პროცესს. გარე მოწყობილობა თავისი დავალების შესრულების შემდეგ პროცესორს უბრუნებს მაგისტრალზე კონტროლის ფუნქციას. პირდაპირი დაშვება ჩვეულებრივ გამოიყენება გარე მოწყობილობის მიმართვისათვის

პროცესორის მეხსიერებასთან, ამიტომ ხშირად ამ რეჟიმს უწოდებენ მეხსიერებასთან პირდაპირი დაშვების რეჟიმს.

პროგრამის წყვეტის რეჟიმში აქტიურ მოწყობილობად რჩება პროცესორი, მაგრამ გაცვლის ოპერაციის შესრულება ხდება გარე მოწყობილობის ინიციატივით, რომელიც რაღაც მოვლენის ან რაღაც თავის დავალების დასრულების შემდეგ, პროცესორს უყენებს მოთხოვნას მომსახურებაზე.

მაგისტრალი შედგება რამდენიმე სალტისაგან. ყველა მოწყობილობა, მათ შორის პროცესორიც, სალტებთან ერთვებიან პარალელურად. სალტების უმრავლესობა ორმიმართულებიანია, რაც იმას ნიშნავს, რომ ერთი და იგივე სალტე შეიძლება გამოყენებულ იქნეს როგორც მეხსიერებაში მონაცემების ჩასაწერად, ასევე მათ ამოსაკითხავად პროცესორის ან სხვა აქტიური მოწყობილობების საშუალებით. მაგისტრალი გათვლილია 16 თანრიგიანი მონაცემთა სიტყვების და მისამართის გადაცემისთვის. ასევე გათვალისწინებულია ინფორმაციის გადაცემის შესაძლებლობა ბაიტების მიხედვით.

სალტე „მისამართი/მონაცემები“ შედგება 16 ხაზისაგან (AD00-AD15) და წარმოადგენს ორმხრივ მიმართულს. ერთი და იგივე ხაზების შეთავსებელი გამოყენება მისამართებისა და მონაცემებისათვის მიიღწევა დაყოფით დროის მიხედვით. მაგისტრალთან ყოველნაირი მიმართვა იწყება მასზე იმ მოწყობილობის მისამართის დასმით, რომელთანაც მიმართვას ახორციელებს აქტიური მოწყობილობა. აქტიური მოწყობილობა ციკლის სამისამართო ნაწილის დამთავრების შემდეგ ახორციელებს მონაცემების მიღებასა და გადაცემას.

გაცვლის მართვის სალტე (ხაზები „შეტანა“ „გამოტანა“ „ ბაიტი“) განსაზღვრავს გაცვლის მიმართულებას.

სიგნალი „შეტანა“ (DIN) ნიშნავს მონაცემების გადაცემას პასიური მოწყობილობიდან აქტიური მოწყობილობისაკენ (ოპერაცია წაკითხვა, ამოკითხვა), ხოლო სიგნალი „გამოტანა“ (DOUT)-ნიშნავს, რომ ხდება მონაცემების გადაცემა აქტიური მოწყობილობიდან პასიური მოწყობილობისკენ (ჩაწერის ოპერაცია). სიგნალი „შეტანა“ გამომუშავდება ორ შემთხვევაში: 1. როცა სიგნალი „შეტანა“ ნიშნავს, რომ აქტიური მოწყობილობა მზადაა მიიღოს მონაცემები პასიური მოწყობილობისაგან. 2.

როცა სიგნალი „შეტანა“-ს საშუალებით ხორციელდება წყვეტის ვექტორის მისამართის შეტანის (წაკითხვის) ოპერაცია.

სიგნალი „გამოტანა“ გამოიმუშავდება აქტიური მოწყობილობის მიერ და ნიშნავს, რომ სრულდება ჩაწერის ოპერაცია და სალტეზე „მისამართი-მონაცემები“ მოთავსებულია მონაცემები. ამ სიგნალზე პასიური მოწყობილობა გამოიმუშავებს საპასუხო სიგნალს.

სიგნალი „ბაიტი“ (WTBT) გამოიყენება ორ შემთხვევაში: მისამართის გადაცემისას, როცას სიგნალი „ბაიტი“ მიუთითებს, რომ შემდეგში ადგილი ექნება გამოტანის ოპერაციის და არა შეტანის ოპერაციის მიღევნას, და მონაცემების გადაცემისას, როცა ეს სიგნალი მიუთითებს, რომ გამოიტანება ბაიტი და არა სიტყვა.

სალტე სინქრონიზაცია მოიცავს ხაზებს „აქტიური მოწყობილობის სინქრონიზაცია“ (SYNC), „პასიური მოწყობილობის სინქრონიზაცია“ (RPLY) და „მისამართი მიღებულია“ (AR). SYNC სიგნალი გამოიმუშავდება აქტიური მოწყობილობით, მიიღება პასიური მოწყობილობის მიერ და ასრულებს ორ ფუნქციას:

-სიგნალის დაყენების შემთხვევაში მოითხოვება პასიური მოწყობილობისგან შესრულდეს ის ოპერაცია, რომელიც მითითებულია კოდით სალტეებზე „შეტანა“ და „გამოტანა“.

-სიგნალის მოხსნის შესახებ ინფორმაციას აკეთებს პასიური მოწყობილობა, იმის შესახებ რომ აქტიური მოწყობილობა თვლის მონაცემთა გადაცემას დამთავრებულად.

სიზ (RPLY) - სიგნალი პასიური წარმოადგენს პასიური ელემენტის პასუხს სიგნალებზე „შეტანა“, „გამოტანა“ და ნიშნავს იმას, რომ მონაცემები მიღებულია ან მოთავსებულია სალტეზე „მისამართი/მონაცემები“.

სიგნალი „მისამართი მიღებულია“ მიუთითებს პასიური მოწყობილობის მიერ მაგისტრალიდან მიღებულ მისამართზე.

სალტე წყვეტის მართვა მოიცავს ხაზებს „წყვეტის მოთხოვნა“ წმ (VIRQ), „წყვეტის წარმოდგენა“ წწ (IAKO) და „რადიალური წყვეტა“ რწ (WIR).

სიგნალი წყვეტის მოთხოვნა (წმ) ფორმირდება პასიური მოწყობილობით, თუ ამ უკანასკნელისთვის ნებადართულია წყვეტა, პროცესორს აძლევს ინფორმაციას იმის შესახებ, რომ მოწყობილობა მზადაა გადასცეს ან მიიღოს მონაცემები.

სიგნალი „წყვეტის წარმოდგენა“ ნებას რთავს მოწყობილობას, რომელიც აკეთებს მიმართვას წყვეტაზე, დააყენოს წყვეტის ვექტორი სისტემურ მაგისტრალზე. წყვეტის ვექტორი თავის მხრივ წარმოადგენს მეხსიერებაში წყვეტის დამუშავების პროგრამის დასაწყისის 8 თანრიგა მისამართს და მოვლენის თავისებურ იდენტიფიკატორს. ასეთი იდენტიფიკატორის არსებობის აუცილებლობა განპირობებულია იმით, რომ წყვეტათა წყაროები შეიძლება იყოს რამდენიმე და ამდენად პროცესორმა უნდა შეძლოს რაღაცით განასხვავოს ისინი.

სიგნალი „რადიალური წყვეტა“ მაღალპრიორიტეტული არაშენიშნული წყვეტაა წყვეტის ფიქსირებული ვექტორით (250₈). წყვეტის დამუშავება ხორციელდება პროცესორის მიერ მიმდინარე ბრძანების შესრულების დამთავრებისთანავე. ეს წყვეტა გამოიყენება დამატებით მაგისტრალებთან გაცვლის შეცდომების დამუშავებისათვის, ან სხვა შემთხვევებში, როცა მოითხოვება სწრაფი რეაგირება.

სალტე პირდაპირი დაშვების მართვა შეიცავს ხაზებს „პირდაპირი დაშვების მოთხოვნა“ (პდმ) (DMR), „პირდაპირი დაშვების წარმოდგენა (პდწ) (DMGO),
“ ამორჩევის დადასტურება“ (ად) (SACK).

სიგნალი პდმ გამომუშავდება იმ მოწყობილობით, რომელიც თხოულობს მაგისტრალს.

სიგნალი პდწ გამომუშავდება პროცესორის მიერ არხთან მიმართვის მიმდინარე ციკლის დასრულებისას და ნებას რთავს მოწყობილობას, რომელიც თხოულობს პირდაპირი დაშვების წარმოდგენას დაიკავოს მაგისტრალი.

სიგნალი ამორჩევის დადასტურება გამომუშავდება პირდაპირი დაშვების მოწყობილობის მიერ პროცესორის სიგნალის პირდაპირი დაშვების წარმოდგენის საპასუხოდ და ნიშნავს, რომ მოწყობილობამ დაიკავა მაგისტრალი და შეუძლია განახორციელოს მონაცემების გაცვლა.

დამატებით მაგისტრალებთან დაშვების მართვის სალტე შეიცავს ხაზებს „დამატებითი მაგისტრალი 1-ის დაკავებაზე მიმართვა“ (WRQ1), „დამატებითი მაგისტრალი 2-ის დაკავებაზე მიმართვა“ (WRQ2) და „დამატებითი მაგისტრალის დაკავებაზე ნებისდართვა“ (WACK). თითოეული პროცესორი შესაძლებელია დაკავშირებული იყოს სხვა ორი პროცესორის მაგისტრალებთან. ამ მაგისტრალებზე

მისამართებთან მიმართვისას გამომუშავდება სიგნალი „მიმართვა დამატებითი მაგისტრალის 1(2) -ის დაკავებაზე.“ გარე არბიტრი ტრანსლირებას უკეთებს ამ მიმართვას მაგისტრალის „უფროსთან“-პროცესორთან, ნების დართვის მიღების შემდეგ ის აყენებს სიგნალს „დამატებითი მაგისტრალის დაკავებაზე თანხმობა“.

გარდა ჩამოთვლილი სისტემებისა მაგისტრალი შეიცავს ასევე შემდეგ დამატებით ხაზებს.

„ჩამოყრა“ (INIT) - გარე მოწყობილობების დაყენება საწყის მდგომარეობაში. ეს სიგნალი ფორმირდება შემდეგ შემთხვევაში:

- როცა პროცესორის პროგრამაში ხდება ინიციალიზაციის ბრძანებას (RESET);
- როცა ელექტრული კვების დარღვევისას პროცესორი ახდენს წყვეტის შემუშავებას;
- როცა ხდება ელექტრული კვების აღდგენა.

„ტაქტური სიხშირე“ (CLCO) - პროცესორის შიგა ტაქტური სიხშირე, მასთან მომუშავე მოწყობილობების სინქრონიზაციისათვის. ამ სიგნალის სიხშირე და ფაზა ემთხვევა შესასვლელ ტაქტურ სიხშირეს, რომელიც მიეწოდება პროცესორში კვარცული გენერატორიდან (CLCI).

„სისტემურ მეხსიერებასთან მიმართვა“ (SEL). სიგნალის გამოჩენა მისამართის გაცემის ფაზაში მიუთითებს იმაზე, რომ, მიმართვა ხდება არა ძირითად, არამედ დამატებით (სისტემურ) მეხსიერებასთან. ამოკითხვისას ამ სიგნალის გამოჩენა „შეტანა“ სიგნალთან ერთად, მიუთითებს პროცესორის მიერ წაკითხვის ოპერაციის შესრულებაზე საწყისი პირობების რეგისტრიდან.

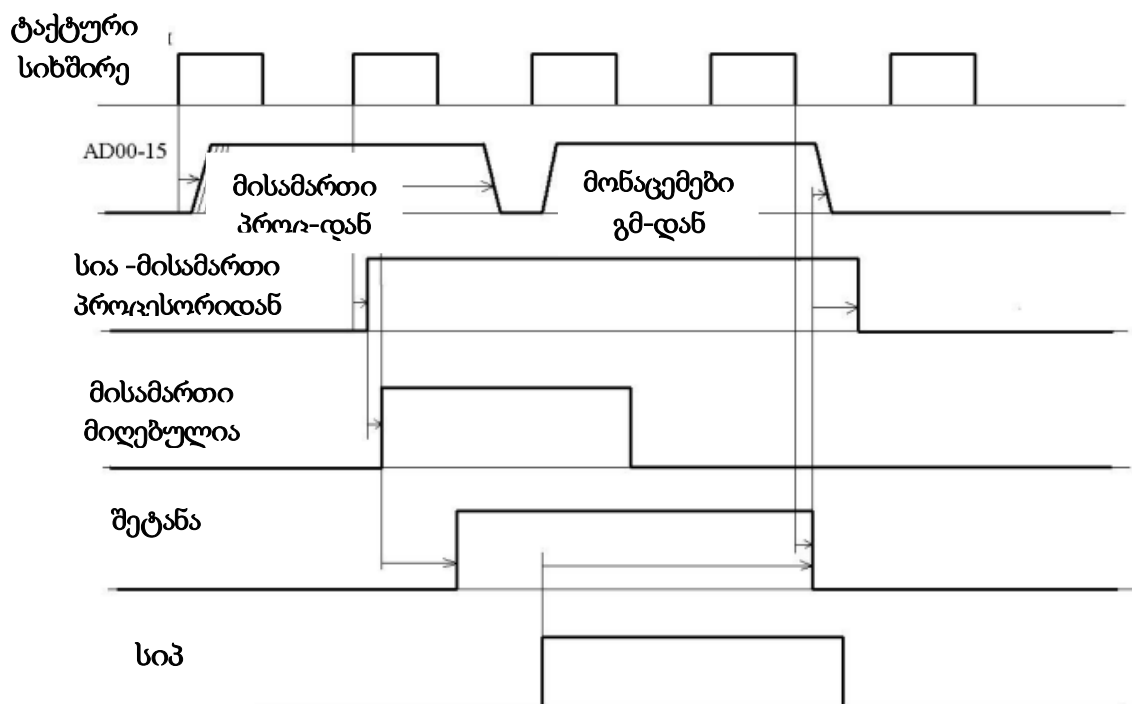
„მიმართვა წყვეტაზე ტაიმერიდან“ (EVNT) - სიგნალი გამომუშავდება სისტემური ტაიმერით. ამ სიგნალის მიხედვით პროცესორი გადადის მომსახურების პროგრამაზე მუდმივი ვექტორის საშუალებით 100₈ მისამართით (თუ პროცესორისთვის ნებადართულია წყვეტა).

„კვების წყაროს ავარია“ (კწა) (DCLO) - სიგნალი გამომუშავდება კვების წყაროს მიერ იმ შემთხვევაში, თუ მუდმივი ძაბვა მის გამოსასვლელზე გახდება დასაშვებ დონეზე ნაკლები. ეს სიგნალი იწვევს პროცესორის გაჩერებას: „აყენებს მას საწყის მდგომარეობაში და ხდება სიგნალ „ჩამოყრა“-ს გამოჩენა.

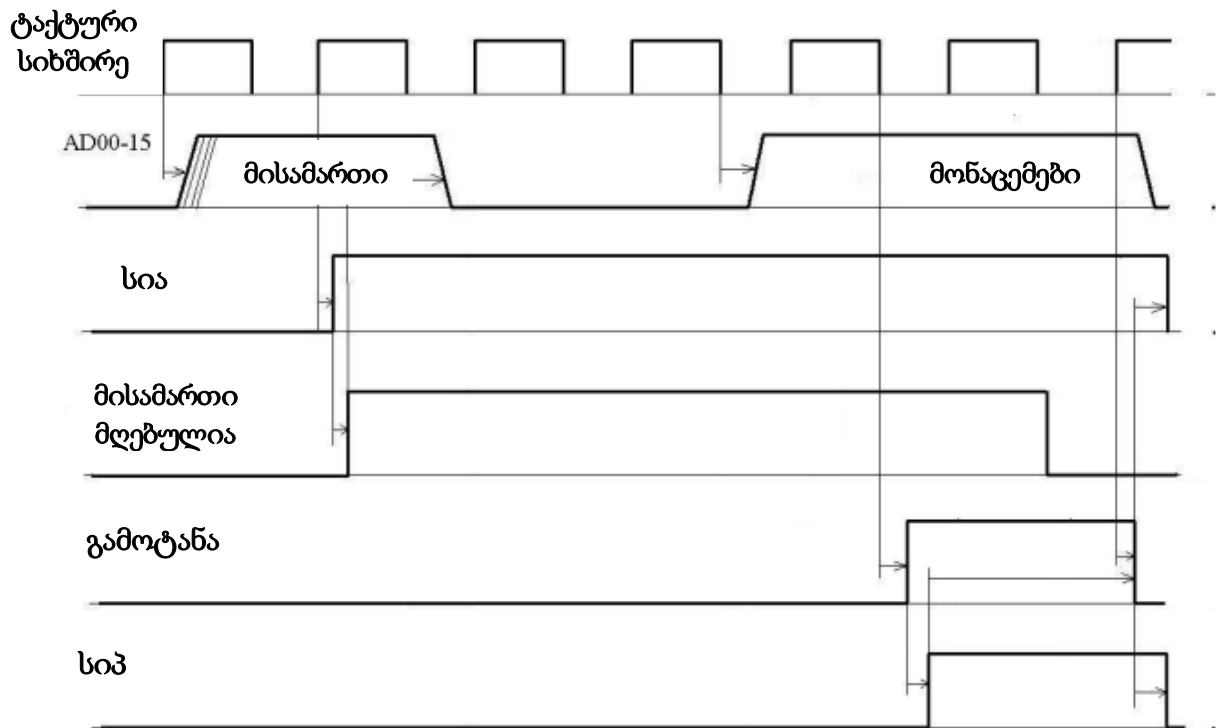
„ქსელური კვების ავარია“ (ქკა) (ACLO) - გამომუშავდება კვების წყაროს მიერ იმ შემთხვევაში, თუ ცვლადი ძაბვა გახდება დასაშვებ დონეზე ნაკლები. ეს სიგნალი იწვევს პროცესორის პროგრამის წყვეტას და გადადის წყაროს შეფერხებით გამოწვეული მდგომარეობის გამო წყვეტის დამუშავების ქვეპროგრამაზე (მუდმივი ფაქტორი 000).

„გაჩერება“ (HALT) - გადასვლა გამართვის რეჟიმში.

შეტანა. პროცედურა „შეტანა“- ს დროითი დიაგრამა გამოსახულია ნახ.9.1-ზე.



ნახ.9.1. შეტანის პროცედურის დროითი დიაგრამა



ნახ.9.2. გამოტანის პროცედურის დროითი დიაგრამა

გაცვლა წარმოებს შედეგნაირად. სიგნალის „ტექტური სიხშირის“ მაღალი დონის მიხედვით პროცესორი მაგისტრალზე აყენებს მისამართს, ხოლო ტაქტის შემდეგ - სიგნალს სია (აქტიური მოწყობილობის სინქრონიზაცია). გარე მოწყობილობა ღებულობს მისამართს და პასუხობს „მისამართი მიღებულია“. მიიღებს რა ამ სიგნალს პროცესორი მაგისტრალიდან ხსნის მისამართს და აყენებს სიგნალს „შეტანა“. მიიღებს რა სიგნალს „შეტანა“ გარე მოწყობილობა ათავსებს მაგისტრალზე მონაცემებს და იმავდროულად სიგნალსაც „სინქრონიზაცია პასიური ელემენტის (სიპ)“. პროცესორი ღებულობს მონაცემებს და ხსნის სიგნალს „შეტანა“. ეს მიუთითებს სიგნალზე შეტანის პროცედურის დამთავრების შესახებ და ყველა სიგნალი ბრუნდება საწყის მდგომარეობაში.

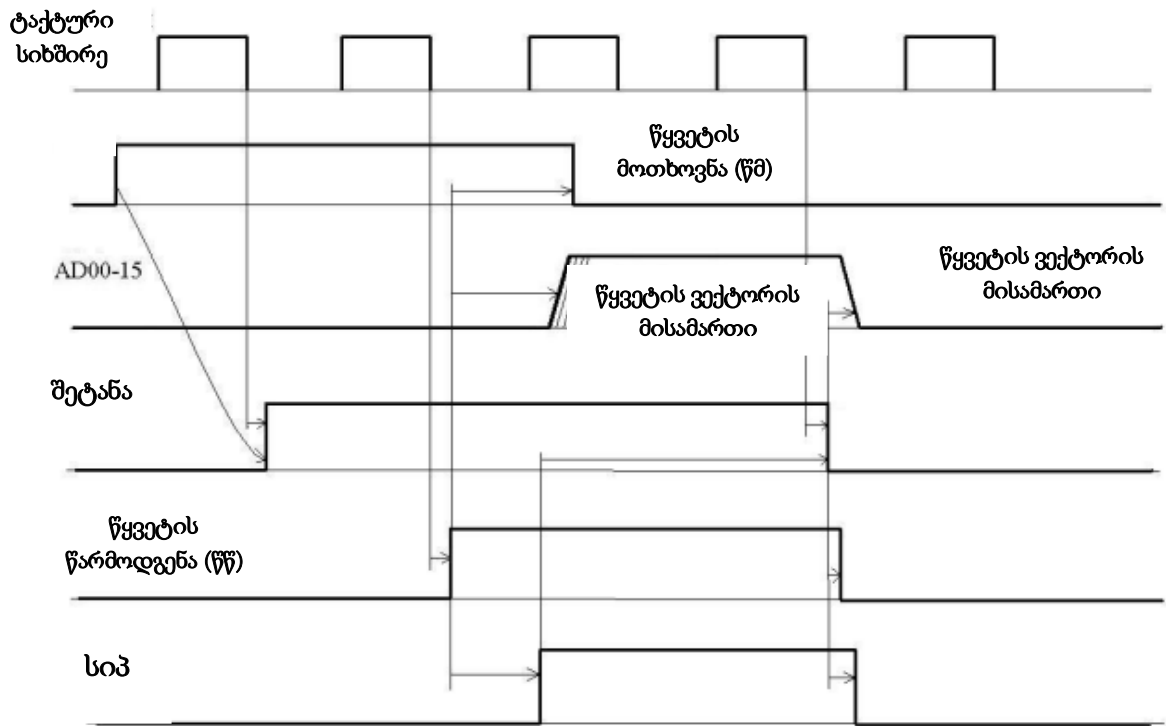
გამოტანა. პროცედურის „გამოტანა“ (ჩაწერა) დროითი დიაგრამა ნაჩვენებია ნახ.9.2-ზე. მისამართის გაცემის ფაზა სრულდება ისევე, როგორც „შეტანა“ პროცედურის დროს. გაცვლა წარმოებს შედეგნაირად. პროცესორი მიიღებს რა სიგნალს „მისამართი მიღებულია“ ის მაგისტრალიდან ხსნის მისამართს და განათავსებს მონაცემებს სიგნალის „ტექტური სიხშირე“ დაბალი დონის მიხედვით. ტაქტის შემდეგ პროცესორი განათავსებს სიგნალს „გამოტანა“. გარე მოწყობილობა მიიღებს რა სიგნალს „გამოტანა“ იგი ღებულობს

მონაცემებს და განათავსებს მაგისტრალზე სიგნალს სიპ (სინქრონიზაცია პასიური ელემენტის). მიიღებს რა პროცესორი სიპ სიგნალს, იგი ხსნის სიგნალს „გამოტანა“. ეს მიუთითებს გამოტანის დამთავრების პროცედურაზე და ყველა სიგნალი ბრუნდება საწყის მდგომარეობაში.

შეტანა-პაუზა-გამოტანა. ეს პროცედურა საშუალებას იძლევა მიღებულ იქნეს ოპერანდი, მოხდეს მასზე ოპერაციის შესრულება და შედეგი ჩაიწეროს იმავე მისამართზე. ის შედგება სამი ფაზისაგან. მისამართის გაცემის ფაზა - ეს ისეთივეა როგორც პროცედურა „შეტანის“ დროს, იმის გამოკლებით, რომ სიგნალ „შეტანა“-ს ჩამოგდების შემდეგ სიგნალი სიპ არ იხსნება. ჩაწერის ფაზა იწყება მხოლოდ მას შემდეგ როცა მოიხსნება სიგნალი სიპ, რომელიც განთავსებული იყო ამოკითხვის ფაზის დროს და მიმდინარეობს ისე, როგორც ეს აღწერილი იყო „გამოტანა“ პროცედურის შესრულების დროს.

წყვეტის ვექტორის მისამართის მიღება. წყვეტის დამუშავება იწყება წყვეტის ვექტორის მიღებით. ამ პროცესის დროითი დიაგრამა ნაჩვენებია ნახ.9.3-ზე.

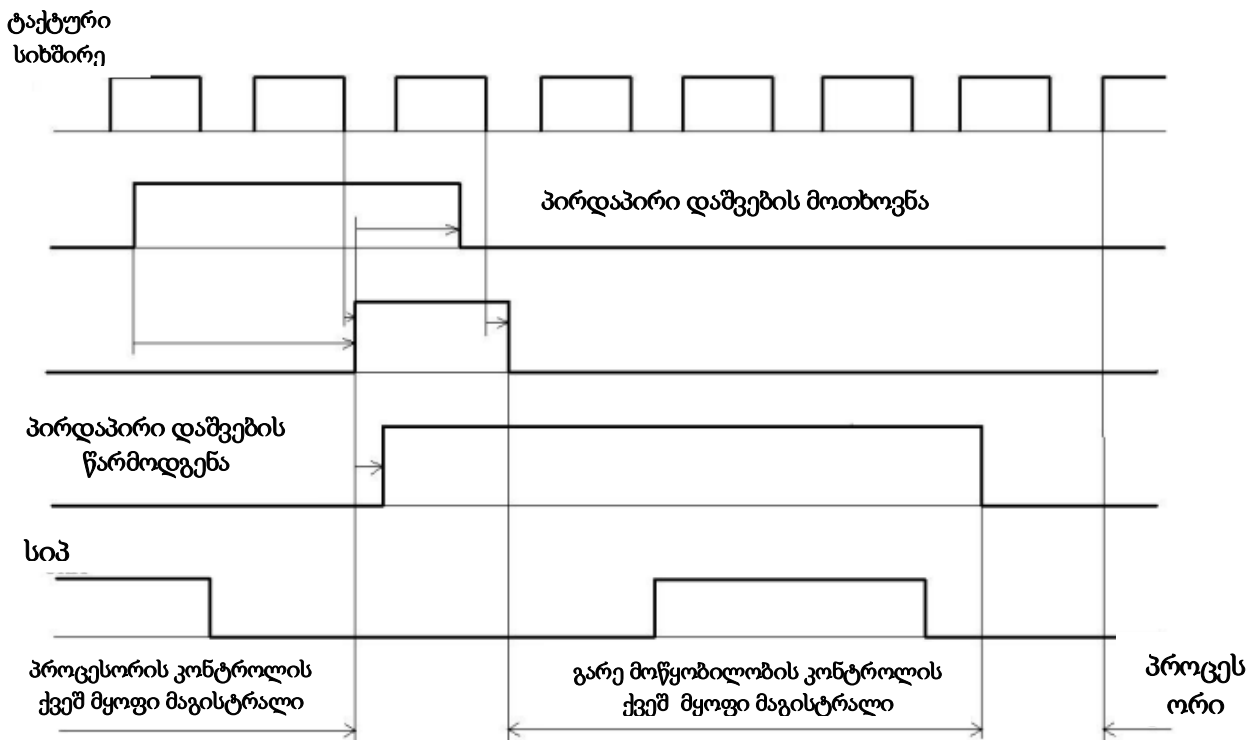
წყვეტის ვექტორის მიღება მიმდინარეობს შემდეგნაირად. პროცესორი წყვეტაზე მიმართვის მიღების შემდეგ (თუ წყვეტა არა დაბლოკილი და მაგისტრალი თავისუფალია) ის იწყებს წყვეტის დამუშავების პროცედურას. პროცესორის ბრძანებათა მთვლელის და მდგომარეობის რეგისტრის შიგთავსი გადაიგზავნება სტეკში. შემდეგ სიგნალ „ტაქტური სიხშირის“ დაბალი დონის მიხედვით პროცესორი აყენებს სიგნალს „შეტანა“. ტაქტის შემდეგ პროცესორი აყენებს წყვეტის ნებისდართვას სიგნალს (წნს) და ელოდება პასუხს - სიგნალს პასიური ელემენტიდან (სიპ). შემწყვეტი მოწყობილობა მაგისტრალზე განათავსებს წყვეტის ვექტორის მისამართს და სიგნალს სიპ. პროცესორი ამოკითხავს ვექტორის მისამართს და ხსნის სიგნალს „შეტანა“. ამით პროცედურა დასრულებულია.



ნახ.9.3. წყვეტის ვექტორის მიღება

მეხსიერებასთან პირდაპირი დაშვება. მეხსიერებასთან პირდაპირი დაშვების ორგანიზაციის დიაგრამა ნაჩვენებია ნახ.9.4-ზე.

მეხსიერებასთან პირდაპირი დაშვების ორგანიზაცია ხორციელდება ასეთნაირად. პროცესორი მიიღებს რა მიმართვას პირდაპირი დაშვების შესახებ, ის ამთავრებს მიმდინარე გაცვლას მაგისტრალის მიხედვით და სიგნალის „ტაქტური სიხშირე“ დაბალი დონის გადასვლაზე აყენებს პირდაპირი დაშვების წარმოდგენის ნებისდამრთველ სიგნალს.



ნახ.9.4 მეხსიერებასთან პირდაპირი დაშვების დროითი დიაგრამა

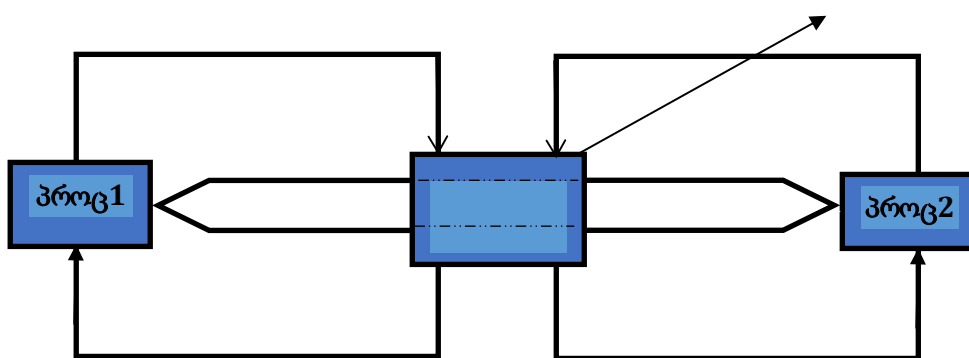
პროცესორი მიიღებს რა მიმართვას პირდაპირი დაშვების შესახებ, ის ამთავრებს მიმდინარე გაცვლას მაგისტრალის მიხედვით და სიგნალის „ტექტური სიხშირე“ დაბალი დონის გადასვლაზე აყენებს პირდაპირი დაშვების წარმოდგენის ნებისდამრთველ სიგნალს (პდნსი). მიიღებს რა ნებისდამრთველ სიგნალს მოწყობილობა, რომელმაც მოითხოვა პირდაპირი დაშვება მეხსიერებასთან, საჭიროა დასაწყისში წარმოადგინოს დადასტურება ამორჩევის დადასტურების (ად) შესახებ, ხოლო შემდეგ მოხსნას სიგნალი „პირდაპირი დაშვების მოთხოვნა“. ამის შემდეგ პროცესორი სიგნალის „ტექტური სიხშირე“-ს დაბალი დონის ფაზაში ხსნის პირდაპირი დაშვების წარმოდგენის სიგნალს და გადაჰყავს გამორთულ მდგომარეობაში მაღალი შესასვლელი წინააღმდეგობით თავისი შესასვლელში სია, „შეტანა“, „გამოტანა“, „ბაიტი“ და ADOO-AD15. მაგისტრალი გადადის გარე მოწყობილობით მართვის რეჟიმში. ინფორმაციის გაცვლის დამთავრების შემდეგ, გარე მოწყობილობა ხსნის ამორჩევის დადასტურების სიგნალს (ად).

პროცესორი იბრუნებს მართვის უფლებას მაგისტრალზე მხოლოდ მას შემდეგ როცა გარე მოწყობილობის მიერ მოხდება სიგნალების - სია, „მისამართი მიღებულია“, სიპ

მოხსნა. მაშინ პროცესორს ყველა თავისი გამოსასვლელი გადაჰყავს აქტიურ მდგომარეობაში და განაგრძობს მუშაობას.

მაგისტრალის დაკავება „ფანჯრის“ საშუალებით.

პროცესორში სამისამართო სივრცის ორი არე გამოყოფილია ფანჯრების არეებით W_0 და W_1 , რომელიც საშუალებას იძლევა მოხდეს მათი დაკავება სხვა პროცესორის მაგისტრალის ფანჯრიდან „დანახვის“ საშუალებით. ეს საშუალებას იძლევა შეიქმნას მრავალპროცესორული სისტემები ლოკალური მეხსიერებით, ისე როგორც ეს ნაჩვენებია ნახ.9.5-ზე.



ნახ.9.5. ორპროცესორული სისტემა

თუ მისამართი გაცვლის დროს მოხვდება ფანჯრების არის ინტერვალში, მაშინ პროცესორი მანამ სანამ ის დაიწყებდეს გაცვლას, აგზავნის მიმართვას დამატებითი მაგისტრალის 1 ან 2 -ის დაკავების შესახებ. ფანჯრის მართვის გარე სქემა (არბიტრი) ხელთ იგდებს მაგისტრალით მოთხოვნილ მართვას (მაგალითად, მეხსიერებასთან პირდაპირი დაშვებით) და გასცემს მოთხოვნილი პროცესორისადმი თანხმობას დამატებითი მაგისტრალის დაკავებაზე. ამის შემდეგ პროცესორი ასრულებს გაცვლის ციკლს „უცხო“ მაგისტრალზე.

9.2.პერსპექტიული შიგა ინტერფეისები

პარალელური მაგისტრალები ვერ უზრუნველყოფენ კომპიუტერების შიგა ინტერფეისებისადმი წაყენებული თანამედროვე მოთხოვნების დაკმაყოფილებას. მათი სიგრძე შემოსაზღვრულია, ამდენად ისინი არ იძლევიან შესაძლებლობას შეიქმნას

განაწილებული სისტემები, რომელთა კომპონენტები განთესილია მნიშვნელოვან მანძილებზე. პარალელურ სალტებზე რეალიზებული სისტემები ცუდად მასშტაბირებადი არიან, რის გამოც შეუძლებელია მათ ბაზაზე გაცემული იქნეს სხვადასხვა დანიშნულების და სირთულის სისტემები. პარალელური მაგისტრალები თხოულობენ ბევრ კონტაქტებს, ამიტომ იკავებენ ნაბეჭდი ფირფიტის მნიშვნელოვან (თუ არა დიდ) ნაწილს, ისინი მგრძნობიარენი არიან ელექტრომაგნიტური დაბრკოლებების მიმართ, მათი ფუნქციონირებისათვის საჭიროა დიდი ენერგია, ისინი გაბარიტულად დიდი და ძვირია. ხოლო თანამედროვე მაღალსიჩქარული თანმიმდევრობითი ინტერფეისები საშუალებას იძლევიან უზრუნველყოფილი იქნეს ინფორმაციის მაღალი სიჩქარით გადაცემა, ისინი მარტივი და იაფი არიან, ადვილია ასეთი ინტერფეისების დარეზერვება, აპარატურაში ჩაშენება, იკავებენ პატარა ადგილს, მოითხოვენ ცოტა ენერგიას, მათი საშუალებით მარტივადაა უზრუნველყოფილი ინფორმაციის გადაცემა დიდ მანძილზე. უკვე არსებობს მაგალითები საბორტო კომპიუტერებში მათი შიგა მაგისტრალების გამოყენების სახით. ასე მაგალითად, საჰაერო ხომალდზე BOENG 777 ინტეგრირებული ავიონიკის AIMS კომპლექსის შიგა ძირითად ინტერფეისად გამოყენებულია თანმიმდევრობითი სალტე SAFEbus. მისთვის დამუშავებულია და გამოშვებულია სტანდარტი ARINC-659.

არსებობს რამდენიმე სავარაუდო კანდიდატები პერსპექტიული კომპიუტერების შიგა ინტერფეისის როლის თვალსაზრისით, რა თქმა უნდა მათ შორის პერსპექტიული საბორტო კომპიუტერებისათვის. ესენი პიველ რიგში არიან Rapidio, Infiniband, Star Fabrio, ასევე PCI Express, 3GIO, Hyper Transport, Packet Switched Ethernet და სხვა.

RapidIO თავიდანვე გათვლილი იყო კომპიუტერის შიგნით მუშაობისათვის, კრისტალებშორისი და ნაბეჭდფირფიტებშორისი შეერთებათა შექმნისათვის. ამასთან არ არსებობს არავითარი პრინციპული მიზეზი, იმისა რომ RapidIo გამოყენებული იქნეს კავშირის სახით სისტემებს შორის. RapidIo-ს არქიტექტურა მუშავდებოდა სპეციალურად რეალური დროის რეჟიმში ამოცანებისათვის, ამიტომ ამ ინტერფეისისათვის თვისებადია მცირე დაყოვნებები, საიმედოობა და მასშტაბურობა.

დაყოვნებათა მცირე სიდიდე, რომელიც განსაზღვრავს მწარმოებლურობას PCI სალტესთან შედარებით, საშუალებას იძლევა RapidIo-ს საფუძველზე აიგოს მრავალპროცესორული სისტემები. განსაზღვრულია პრიორიტეტების რამდენიმე დონე, რომლებიც გარანტიას იძლევიან მოახდინონ ყველაზე უფრო საჭირო მონაცემების მოძებნა კომპუტირებადი სტრუქტურის გადატვირთვის პირობებშიც კი. შეცდომების აღმოჩენა ხდება აპარატურულად და სწორდება ასევე მონაცემთა გადაცემის თითოეულ რგოლში. შეცდომების გასწორებისთვის მონაცემები მიყოლებიან საკონტროლო CRC-კოდის გამოყენებით. გარდა ამისა, შეცდომების აღმოჩენის გარანტიისთვის ყველა პაკეტი გადაიცემა ორჯერ ნორმალური და ინვერსირებული სახით.

გამოიყენება მონაცემების გადაცემის შემდეგი სიჩქარეები 1,25/2,5/3,125 გიგა ბიტი/წამში. სისტემის შიგნით მიკროსქემები და მოდულები სისტემის შიგნით უერთდებიან ერთმანეთს 8 და 16 თანრიგა პარალელური არხების დახმარებით დაბალვოლტიანი დიფერენციალური წყვილების ბაზაზე (LVDS), ჯამური გამტარუნარიანობა თითოეული მიმართულებით შეადგენს 8-დან 32 გიგა ბიტი/წამში. სიგნალების გადაცემის LVDS სისტემა დღესდღეობით პოლპულარულია და გამოირჩევა დიდი დაბრკოლებამდგრადობით, შეცდომათა მცირე ალბათობით და ენერგიის მცირე მოხმარებით.

RapidIo-ში გამოიყენებიან ექვსპორტიანი კომპუტატორები. მათ გააჩნიათ 6-6 შესასვლელი, რომლებიც შესაძლებელია წყვილ-წყვილად შეერთდნენ ერთმანეთს შორის ნებისმიერი შეხამებით. გადაცემის გარემოს სახით გამოიყენება კაბელი სპილენძის ძარღვით (ქილით). ამასთან ხაზის სიგრძე შემოსაზღვრულია 17 მეტრით.

არსებობს RapidIo-ს სუფთა თანმიმდევრობითი ვერსია, რომლის ფიზიკური დონე თხოულობს ოთხ კონტაქტს ერთ დუბლექსურ პორტზე. თანმიმდევრობითი სხვადასხვაგვარობა ორიენტირებულია კავშირის ოპტიკურბოჭკოვან ხაზებზე.

Star Fabric. ეს არის მასშტაბური არქიტექტურა, რომელსაც შეუძლია გააერთიანოს ერთიდან დაწყებული რამდენიმე ათასეული კომპონენტი. ის ძალიან კარგად მიესადაგება ფირფიტებშორისი შეერთებების რეალიზაციას. მას გააჩნია მნიშვნელოვანი განსხვავებული განსაკუთრებულობა: ის მთლიანად გამჭვირვალეა PCI სალტისათვის. PCI-Star Fabric ხიდი, რომელიც შედის ინტერფეისულ აპარატურულ შემადგენლობაში

სრულად თავსებადია ყველა პროგრამული უზრუნველყოფისადმი, რომლებიც დაწერილია PCI-სისტემებისათვის. Star Fabric სისტემის სტრუქტურაში თითოეულ მიმართულებაში გამოიყენება ორი დუბლექსური არხი. თითოეული არხი შედგება ოთხი LVDS-გან გამატრუნარიანობით 622 მეგაბიტი/წმ-ში თითოეული მიმართულებით, შესაბამისად არხის საერთო გატარების უნარი ტოლია 2,5 გიგაბიტი/წამში (ითვალისწინებენ გამტარუნარიანობის მასშტაბირების შესაძლებლობას 10 გიგაბიტი /წმ-მდე). კომპუტატორს Star Fabric-ში გააჩნია 6 პორტი. გადაცემის გარემო შეიძლება იყოს სპილენძის მავთული ან ოპტიკური ბოჭკო.

Infiniband. ინტერფეისი Infiniband დამუშავდა სერვერების, დამხმომებელი მოწყობილობების, ქსელების მართვის საშუალებების მიერთებისათვის. ის საშუალებას იძლევა აიგოს ქსელები მრავალი ქვექსელებით და ათასობით კვანძებით თითოეულ ქვექსელში. ამასთან ერთად ის წარმატებით შესაძლებელია გამოყენებულ იქნეს შიგა ინტერფეისის სახით ციფრულ ელექტრონიკაში, ასევე კარგი საშუალებაა პროცესორული კლასტერების შესაქმნელად. მთავარ ღირსებებს შორის Infiniband-ის მთავარი ღირსებაა უნარი - უზრუნველყოს სერვერის საზღვრებს გარეთ ისეთივე მწარმოებლურობა, როგორც მის შიგნით. არქიტექტურა საშუალებას იძლევა მოინახოს მონაცემები მაკომპუტირებელი სტრუქტურის ნებისმიერ წერტილში გადაცემის ერთი და იგივე მაღალი სიჩქარით.

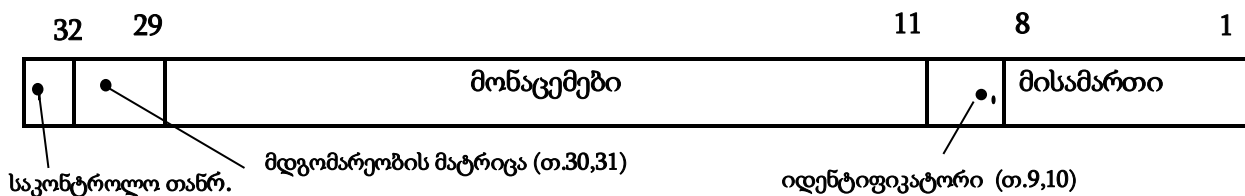
9. 3. ARINC-429-ის მიხედვით ორპოლარული კოდით ინფორმაციის გაცვლა

ამჟამად ARINC-429-ის საშუალებით ორპოლარული კოდით ინფორმაციის გაცვლა სამოქალაქო ავიაციის სააპერო ხომალდზე წარoadგენს ყველაზე უფრო გავრცელებულ ხერხს. ასეთი ხერხით ინფორმაციის გადაცემისას ძირითადი პრინციპი მდგომარეობს შემდეგში: არსებობს ინფორმაციის წყარო, რომელიც ინფორმაციას გადასცემს ყველა იმ მოწყობილობას (მომხმარებელს) რომლისთვისაც ის საჭიროა. ამასთან გამოიყენება თანმიმდევრობითი გადაცემის ხერხი და მთელი ინფორმაცია გადაიცემა წყვილი ეკრანირებული გამტარების ხვეულით - კოდური ხაზით. ინფორმაციის გადაცემა უკუმიმართულებით გამტარების ამავე წყვილის საშუალებით აკრძალულია, ამის

აუცილებლობის შემთხვევაში გამოიყენება სხვა ცალკეული ხაზი. ყველა მიმღები, რომელიც დაინტერესებულია ინფორმაციის წყაროდან ინფორმაციის მიღებაში უერთდებიან ამ გამტარების წყვილს და თითოეული გადაცემული ინფორმაციიდან აირჩევენ მხოლოდ მას, რაც მას ესაჭიროება. გადაცემა, როგორც წესი ასინქრონულია, რაც იმას ნიშნავს რომ წყარო ინფორმაციას (მონაცემებს) გადასცემს მისი მზადყოფნის მდგომარეობის მიხედვით, ხოლო მიმღები დროის ნებისმიერ მომენტში მზად უნდა იყოს მიიღოს გადაცემული მონაცემები, ან უკიდურეს შემთხვევაში მოათავსოს ის ბუფერში, რათა ისინი არ დაიკარგონ. წყაროსათვის შეტყობინება იმის შესახებ მის მიერ გადაცემული მონაცემები მიმღების მიერ მიღებული იქნა სწორად არაა გათვალისწინებული. ძირითადი ინფორმაცია გადაიცემა ციკლური პრინციპით, ამიტომ არასწორად მიღებული მონაცემები შესაძლებელია მიღებულ იქნეს მომდევნო ციკლში. მონაცემების სხვადასხვა სახისთვის გადაცემის ინტერვალი შესაძლებელია იცვლებოდეს 10 მილიწამიდან 1 წამამდე.

ინფორმაციული მახასიათებლები.

ინფორმაციის ძირითადი ელემენტია 32 თანრიგიანი ციფრული სიტყვა. სიტყვის სტრუქტურა ნაჩვენებია ნახ.9.6-ზე.



ნახ.9.6 სიტყვის სტრუქტურა ARINC-429-ის მიხედვით

სიტყვის პირველი 8 თანრიგი გამოიყენება სიტყვაში შემავალი ინფორმაციის იდენტიფიკაციისათვის ან მისი გამოყენების იდენტიფიკაციისათვის. **ARINC-429-ში** ეს 8 თანრიგი წარმოადგენს მისამართს. თითოეულ წყაროს შეუძლია გასცეს 255 სიტყვა სხვადასხვა მისამართებით (ნულოვანი მისამართი არ გამოიყენება). ყველა ძირითადი პარამეტრებისათვის, რომლებიც გამოიყენებიან საფრენ აპარატებში თავიდანვეა დადგენილი მათი მისამართები. ხოლო იმ პარამეტრებისთვის, რომლებიც სტანდარტში არაა მითითებული, მისამართს ირჩევენ სისტემის დამმუშავებელი.

თანრიგები 9,10, წყაროს იდენტიფიკატორია და გამოიყენება ინფორმაციის წყაროს ნომრის მითითებისთვის, როცა ისინი რამდენიმეა (მაგალითად, სისტემის 2

დამოუკიდებელი არხი). ეს თანრიგები შეიძლება გამოყენებულ იქნეს ასევე მიმღების ნომრის მითითებისთვის, რომლისთვისაცაა დამისამართებული ინფორმაცია.

თანრიგები 11-დან 29-ის ჩათვლით გამოყოფილია მონაცემებისთვის. მონაცემების ტიპისაგან დამოკიდებულებით გათვალისწინებულია სიტყვების 5 ძირითადი ტიპი:

- ორობითი კოდი (ოკ) ;
- ორობით-ათობითი კოდი (ოაკ);
- დისკრეტული სიგნალების სიტყვა (დსს);
- ტექნომსახურების მონაცემები;
- ციფრულ-ასოითი ინფორმაცია.

ორობითი კოდის მონაცემები შეიცავენ პარამეტრის მნიშვნელობას ორობით დამატებით კოდში. რიცხვის უფროსი ნიშნადი თანრიგი - 28, არჩეული პარამეტრისთვის მისი წონა ტოლია მაქსიმალური მნიშვნელობის ნახევრისა, უარყოფითი რიცხვი კოდირდება როგორც დადებითი სიდიდის ორობითი დამატება, ხოლო ნიშანი გადაიცემა 29-ე თანრიგში:

0 -ეს პლიუსი, ჩრდილოეთი, აღმოსავლეთი, მარჯვნივ, მაღლა, კენ;

1 -ეს მინუსი, სამხრეთი, დასავლეთი, მარცხნივ, ქვევით, დან.

კუთხური სიდიდეები კოდირდებიან საზღვრებში 0-დან 179,999⁰-მდე. კუთხური სიდიდეები (α) 180⁰-დან 359,999⁰-მდე კოდირდებიან უარყოფითი რიცხვებით - როგორც (360-α) კუთხის ორობითი დამატება. მაგალითად, კუთხე 181⁰ კოდირდება როგორც რიცხვი 179 -ის ორობითი დამატება.

ორობით-ათობითი კოდის სიტყვა შეიცავს 5 ციფრს. თითოეული ციფრი კოდირდება ტეტრადით - 4 თანრიგით (უფროსი ტეტრადა-არასრულია, შეიცავს მხოლოდ 3 თანრიგს 27-დან 29-მდე.) ეს ოთხი თანრიგი წარმოადგენს ციფრებს ორობით კოდში: „0000“-ეს 0-ია, „0001“-ეს 1-ია, „0010“-ეს 2-ია და ა.შ.

დისკრეტული სიგნალების სიტყვა (დსს) შეიცავს დისკრეტულ სიგნალებს - სიტყვის ინფორმაციული ნაწილის თითოეულ თანრიგში ერთ სიგნალს. სიგნალის არსებობა გადაიცემა თანრიგის ერთი მდგომარეობით (მაგალითად, „1“), ხოლო არარსებობა - მეორე მდგომარეობით (მაგალითად, „0“). დისკრეტული სიგნალები შესაძლებელია გადაცემულ იქნეს ასევე ორობითი კოდისა და ორობით-ათობითი კოდების სიტყვების გამოყენებული თანრიგებით.

ARINC-429 არ ითვალისწინებს გრაფიკული მონაცემების გადაცემას, მაგრამ ხაზგასასმელია, რომ აუცილებლობის შემთხვევაში ასეთი გადაცემისათვის შეიძლება გამოყენებულ იქნეს სიმბოლოების კოდირების პრინციპი, რომელიც მსგავსი იქნება ISO #5 ანბანისა, სადაც თითოეული კოდით წარმოდგენილი იქნება არა ასო ან ციფრი, არამედ გრაფიკული ელემენტი - ვექტორი, რკალი და ა.შ.

ტექნომსახურების მონაცემები გადაიცემა მასივებით შემდგარი რამდენიმე სიტყვისაგან. თითოეული სიტყვას მასივში გააჩნია ერთი და იგივე მისამართი, ხოლო მდგომარეობის მატრიცა მიუთითებს თუ რომელია ეს სიტყვა - პირველი, შუალედური თუ ბოლო. პირველ სიტყვაში თანრიგებში 9-16 გადაიცემა მასივებში არსებული სიტყვის რაოდენობა. მონაცემების გადაცემა მასივებით საშუალებას იძლევა მნიშვნელოვნად გაფართოვდეს მონაცემების თანრიგობა (ჩვეულებრივ სიტყვებში მონაცემებმა შეიძლება დაიკავოს მხოლოდ 21 თანრიგი). იგულისხმება, რომ მიმღებმა იცის, თუ როგორი ინტერპრეტაცია უნდა გაუკეთოს მოწოდებულ ინფორმაციას.

მატრიცა ნიშანი/მდგომარეობა (შემდგომში მდგომარეობის მატრიცა) ივსება ცხრილი 9.1-ის შესაბამისად.

მატრიცა ნიშანი/მდგომარეობა

ცხრილი 9.1

31	30	მონაცემთა ტიპი			
		ოაკ, დსს	ოკ	ISO#5	ფაილი
0	0	+, კენ, ჩრდილოეთი, სამხრეთი, მარჯვნივ, ზევით	მტყუნება	შუალედური სიტყვა	როგორც ოაკ ან როგორც ISO#5
0	1	არაა გამოთვლილი მონაცემები	არაა გამოთვლილი მონაცემები	საწყისი სიტყვა	საწყისი სიტყვა
1	0	ტესტი	ტესტი	ბოლო სიტყვა	ბოლო სიტყვა
1	1	_, დან, დასავლეთი, მარცხნივ, ქვევით	ნორმალური მუშაობა	მმართველი სიტყვა	როგორც ოაკ ან როგორც ISO #5

მდგომარეობის მატრიცა „არაა გამოთვლილი მონაცემები“ ნიშნავს, რომ ინფორმაცია სიტყვაში არაუტყუარია, თუმცა იგი განპირობებულია სხვა მიზეზებით და არა თვით სისტემის მტყუნებით. მატრიცა „მტყუნება“ ორობითი კოდის სიტყვებში მიუთითებს სახელდობრ თვით სისტემის მტყუნებაზე. სხვა ტიპის სიტყვებისთვის

შეტყობინება მტყუნების შესახებ არაა გათვალისწინებული, ამ შემთხვევაში სისტემა-წყარომ უბრალოდ უნდა შეწყვიტოს იმ სიტყვების გაცემა, რომელთა გადაცემა მას არ შეუძლია სწორად. მატრიცა „ტესტი“ შეიძლება გადაცემულ იქნეს:

-ინფორმაციულ სიტყვებში. ამ შემთხვევაში სიტყვაში გადაიცემა ტესტის შესრულების შედეგი, მაგალითად, მოწყობილობის გამართულობის მიწისზედა კონტროლის დროს;

-მმართველ სიტყვებში. ამ შემთხვევაში ეს არის - ბრძანება შესრულდეს ტესტი.

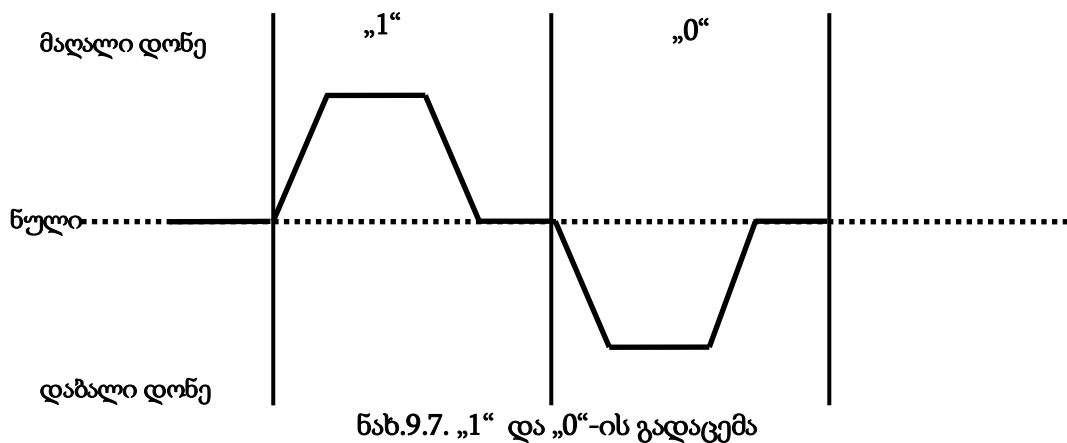
თითოეული სიტყვის შემადგენლობაში გადაიცემა საკონტროლო თანრიგი. ეს არის სიტყვის 32-ე თანრიგი, რომლის შიგთავსი ივსება გადამცემით სიტყვის გადაცემისას ისეთნაირად, რომ სიტყვაში ერთიანების საერთო რიცხვი იყოს კენტი. სიტყვის მიღებისას კონტროლი ლუწობაზე საშუალებას აძლევს მიმღებს განსაზღვროს ერთჯერადი შეფერხება და არ გამოიყენოს ასეთი, არუტყუარი სიტყვა. სიტყვების გამოუყენებელი თანრიგები ივსება ნულებით.

თითოეული სიტყვის გადაცემასთან ერთად, გათვალისწინებულია ფაილების გადაცემის შესაძლებლობაც. ფაილი შეიძლება შეიცავდეს 1-დან 127-მდე ჩანაწერს. თითოეული ჩანაწერი შეიძლება შეიცავდეს 1-დან 126-მდე მონაცემების სიტყვას ორობითი კოდის ფორმატში ან ციფრულასოით ფორმატში. ფაილის გადაცემა ხდება ოქმის (პროტოკოლის) მიხედვით „შეკითხვა-პასუხი“, ამასთან გადამცემსა და მიმღებს შორის უნდა იყოს კავშირის ორი ხაზი-თითო თითო თითოეული მიმართულებით. გადამცემი, რომელსაც გააჩნია მონაცემები გადაცემისთვის გადასცემს თავისი ხაზით სიტყვას „შეკითხვა გაგზავნაზე“. მიმღები თავისი საპასუხო ხაზით პასუხობს „თავისუფალია“. ამის შემდეგ გადამცემი აგზავნის ფაილს მდგომარეობის მატრიცის შესავსებად საწყისი, საშუალო და საბოლოო სიტყვებისთვის. მიმღები ამუშავებს ინფორმაციას ბოლო სიტყვიდან და თუ შეცდომა გადაცემის დროს არ იქნება აღმოჩენილი, პასუხობს „მიღებულია“. წინააღმდეგ შემთხვევაში გადაცემა მეორდება.

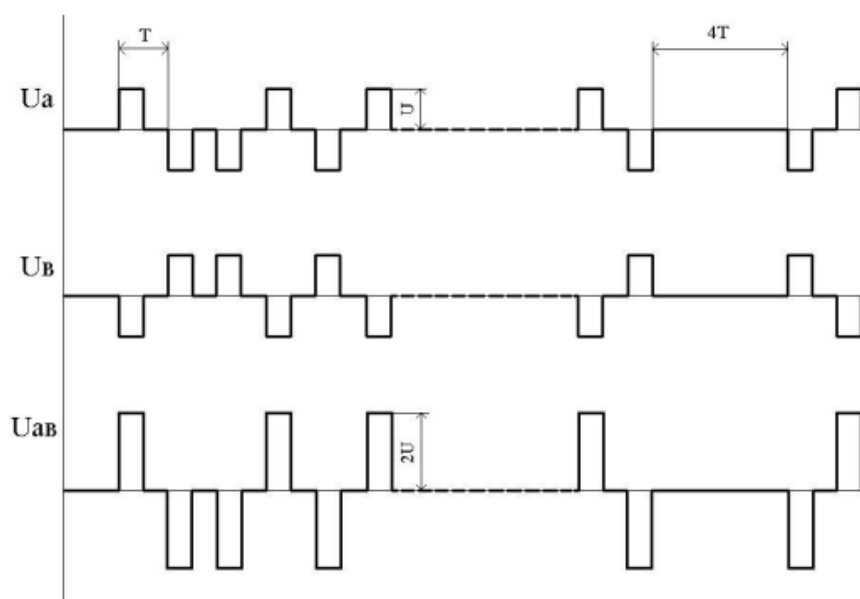
გათვალისწინებულია ინფორმაციის გადაცემის ორი სიჩქარე : 100 კილობიტი/წმ $\pm 1\%$ და 12-14,5 კილობიტი/წმ -ში. ამასთან არაა რეკომენდირებული სიჩქარე 13,6 კბ/წმ და ზუსტად 100 კილობიტი/წმ, იმდენად რამდენადაც 13,6 კილოჰერცსა და 100 კილოჰერცზე მუშაობენ საბორტო რადიოსისტემები. არჩეული სიხშირე შენარჩუნებული უნდა იქნეს 1% სტაბილურობით.

სინქრონიზაცია წარმოებს სიტყვებს შორის პაუზით, რომლის ხანგრძლივობა არ უნდა იყოს 4 ბიტზე ნაკლები. პაუზის შემდეგ პირველი თანრიგი-ეს არის მომდევნო სიტყვის პირველი თანრიგი.

ელექტრული მახასიათებლები. გამოიყენება სიგნალის სამდონიანი მოდულაცია: მაღალი დონე, ნული, დაბალი დონე (ნახ.9.7), მაღალი დონე შეესაბამება ლოგიკურ „1“, დაბალი დონე ლოგიკურ „0“-ს, ხოლო ნულოვანი დონე აღნიშნავს პაუზას გადასაცემ ბიტებს შორის.



გადამცემი გასცემს ორი „a“ და „b“ გამტარებით დიფერენციალურ გამოსავალ სიგნალს U_{ab} , რომელიც ბალანსირებულია კორპუსთან მიმართებაში (ნახ.9.8).



ნახ.9.8. კავშირის კოდურ ხაზში სიგნალების ფორმა

ამასთან სიგნალის ამპლიტუდა ხაზებში არსებული ცალკეულ სიგნალებთან შედარებით (U_a , U_b) ორმაგდება. ძაბვის დონეები, რომლებიც უზრუნველყოფილია გადამცემით უქმი სვლის რეჟიმში, მოყვანილია ცხრილში 9. 2.

სიგნალების დონე კავშირის კოდურ ხაზებში (კვხ)

ცხრილი 9.2

გაზომვის წერტილი	ძაბვის დონე, ვ		
	მაღალი	ნული	დაბალი
„ა“ და „ბ“ ხაზებს შორის U_{ab}	$+10 \pm 1$	$0 \pm 0,5$	-10 ± 1
„ა“ ხაზსა და კორპუსს შორის U_a	$+5 \pm 0,5$	$0 \pm 0,25$	$-5 \pm 0,5$
„ბ“ ხაზსა და კორპუსს შორის U_b	$-5 \pm 0,5$	$0 \pm 0,25$	$+5 \pm 0,5$

მიმღებმა უნდა უზრუნველყოს ინფორმაციის მიღება ძაბვის შემდეგი დონეების შემთხვევაში:

მაღალი $+6,5 \dots 13$ ვ;

ნული $+2,5 \dots -2,5$ ვ;

დაბალი $-6,5 \dots 13$ ვ.

როცა სიგნალი ხვდება დონეებს შორის განუსაზღვრელ ზონაში, სასურველია, რომ მიმღებმა შეწყვიტოს მონაცემების მიღება. ამასთან მიმღებმა უნდა მიიღოს დიფერენციალური შესასვლელი სიგნალი („ა“ და „ბ“ ხაზებს შორის, და არა „ა“ / „ბ“-სა და კორპუსს შორის). ეს უზრუნველყოფს დაბრკოლებამდგრადობას.

ასეთი ხერხის ნაკლოვანებას წარმოადგენს ის, რომ თუ ხდება არათანაბარი რაოდენობის ნულებისა და ერთიანების გადაცემა, ეს იწვევს ელექტრული ენერგიის გადაცემას სალტის რეაქტიულ ელემენტებში, ე.ი. სალტე შეიძლება აღმოჩნდეს დამუხტული, რის გამოც შეტყობინების დასასრული მახინჯდება და მიიღევა.

გადამცემის გამოსასვლელი წინაღობა (R_s) უნდა შეადგენდეს 75 ± 5 ომს და ტოლად უნდა იყოს განაწილებული „ა“ და „ბ“ ხაზებს შორის, იმისათვის, რომ

უზრუნველყოფილ იქნეს წინაღობის მიხედვით ბალანსირებული გამოსასვლელი. წინაღობის სიდიდე არჩეულია კაბელთან შეთანხმების თვალსაზრისით.

გადამცემმა უნდა უზრუნველყოს მუშაობა შემდეგ დატვირთვაზე:

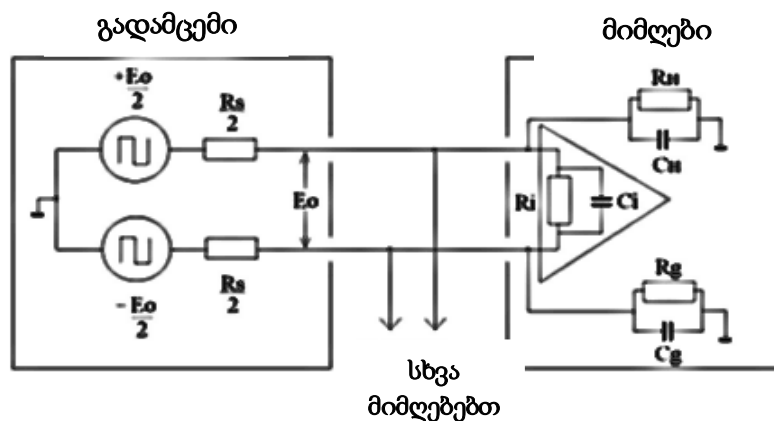
მიმღების საერთო წინაღობა (გამტარების გათვალისწინებით) შეიძლება შეადგენდეს 400-800 ომს.

მიმღების საერთო ტევადობა (გამტარების, სადენების გათვალისწინებით) 1000-30000 პიკოფარადი.

ხაზთან შესაძლებელია მიერთდეს 20 მიმღებამდე.

მიმღებს უნდა გააჩნდეს შემდეგი შესასვლელი მახასიათებლები (ნახ.9.9) :

- დიფერენციალური შესასვლელი წინაღობა R_i არა ნაკლები 12 კილომი;
- დიფერენციალური შესასვლელი ტევადობა C_i არა უმეტესი 50 პიკოფარადი;
- წინაღობა დედამიწის მიმართ არა ნაკლები 12 კილომი;
- ტევადობა დედამიწის მიმართ არაუმეტესი 50 პიკოფარადი.



ნახ.9. 9. სიგნალის გადაცემის სქემა ARINC 429-ის მიხედვით

მიმღების საერთო წინაღობა, გაზომილი მის შესასვლელ კონტაქტებზე, უნდა იყოს არა ნაკლები 8000 ომისა.

ეკრანი სადენების წყვილის ხვეულით შეერთებული უნდა იყოს ორივე ბოლოში.

ლექცია 10

ანალოგურ-ციფრული გარდამქმნელები და ციფრულ-ანალოგური გარდამქმნელები

10.1. ანალოგურ-ციფრული გარდამქმნელები

ანალოგურ-ციფრული გარდამქმნელის (აცგ) დანიშნულებაა ანალოგური სიგნალის გარდაქმნა ორობით კოდურ კომბინაციად.

ანალოგური სიგნალის ციფრულში გარდაქმნის პროცესი შესაძლებელია დაიყოს სამ ეტაპად:

- დისკრეტიზაცია დროის მიხედვით;
- დაკვანტვა დროის მიხედვით;
- კოდირება.

დროში დისკრეტიზაციის ეტაპი მდგომარეობს იმაში, რომ $f(t)$ დროში უწყვეტი სიგნალიდან აირჩევა მისი ცალკეული მნიშვნელობები, რომლებიც თავის მხრივ შეესაბამებიან დროის სატაქტო მომენტებს $t_1, t_2, \dots, t_n$, რომლებიც ფორმირდებიან ხშირ შემთხვევაში დროის ტოლ ინტერვალებად და დისკრეტიზაციის პერიოდს T დ უწოდებენ. დისკრეტიზაციის პერიოდის არჩევა უწყვეტი ფუნქციის წარმოდგენისთვის ხდება კოტელნიკოვის თეორემის პირობიდან, რომლის მიხედვით $f(t)$ ფუნქციის წარმოდგენისთვის, რომელსაც გააჩნია შემოსაზღვრული სპექტრი, აუცილებელია, რომ დისკრეტიზაციის იმპულსების გამეორების სიხშირე

$$F_n = \frac{1}{T_d}$$

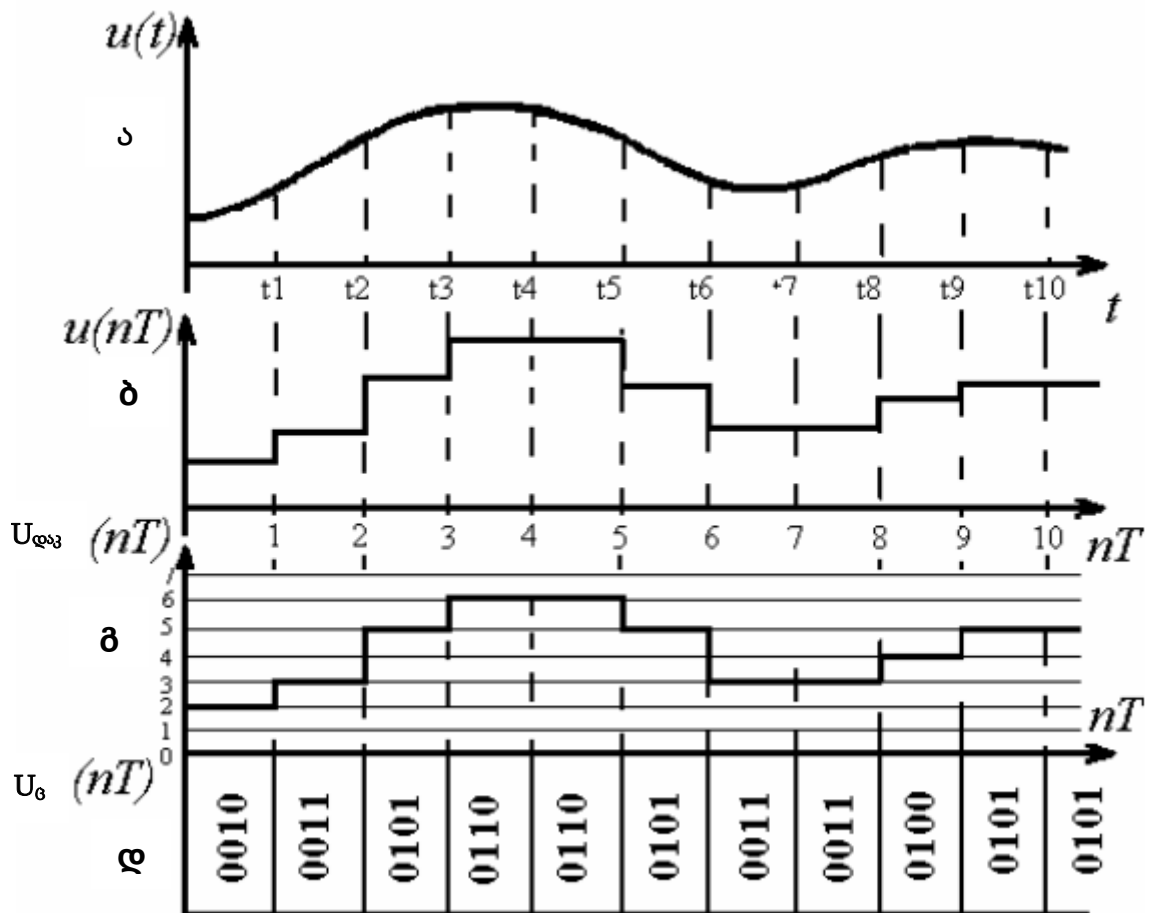
იყოს $f(t)$ ფუნქციის სპექტრის გაორმაგებულ მაქსიმალურ სიხშირეზე მეტი, ე.ი.

$$F_n \geq 2F_{max},$$

სადაც F_{max} - სპექტრის მაქსიმალური სიხშირე.

დაკვანტვის ოპერაცია მდგომარეობს იმაში, რომ იქმნება დაკვანტვის დონეების ბადე, დაძრული ერთმანეთის მიმართ კვანტის ბიჯით h (ნახ.10.1,ბ). დაკვანტვის

თითოეულ დონეს მიეწერება რიგითი ნომერი $(0,1,2,..., n-1)$. შემდეგ საწყისი ანალოგური სიგნალის მნიშვნელობები დისკრეტიზაციის შედეგად იცვლება მათთან უახლოესი დაკვანტვის დონეებით (ნახ.10.1გ). ასეთნაირად ხდება დაკვანტვის დროს საწყისი უწყვეტი ფუნქციის შეცვლა საფეხურებრივი ფუნქციით. ამასთან წარმოიშობა დაკვანტვის ცდომილებები δ_k , იმდენად რამდენადაც დაკვანტვა ახდენს ანალოგური სიგნალის არა ზუსტი მნიშვნელობების აღწარმოებას, არამედ ხდება უახლოეს დონემდე დამრგვალება. დაკვანტვის შეცდომის შემცირება შესაძლებელია დაკვანტვის ბიჯის h - ის შემცირებით.



ნახ.10.1

დაკვანტვისა და დისკრეტიზაციის შეცდომები წარმოადგენს შემთხვევით სიდიდეებს და უწოდებენ დაკვანტვისა და დისკრეტიზაციის ხმაურს.

კოდირების ეტაპზე დონეების დაკვანტვა წარმოდგება რიცხვებით - დაკვანტვის შესაბამისი დონეების ნომრებით (ნახ.10.1გ). შემდეგ ასეთნაირად მიღებული რიცხვების თანმიმდევრობა, წარმოდგება ორობითი კოდებით (ნახ.1დ).

ანალოგურ-ციფრული გარდამქმნელები გამოყენების სფეროების მიხედვით შესაძლებელია დაიყოს ოთხ ძირითად ჯგუფად:

- გადაადგილებათა გარდამქმნელები (წრფივი ან კუთხური);
- ელექტრული სიდიდეების გარდამქმნელები (ძაბვების, დენების ან წინაღობების);
- დროის ინტერვალების გარდამქმნელები;
- მოთვალთვალე ტიპის გარდამქმნელები.

ანალოგურ-ციფრული გარდამქმნელების ძირითად მახასიათებლებს წარმოადგენს თანრიგობა, შესასვლელი ძაბვის ცვლილების დიაპაზონი, ნებდართვის უნარი, გარდაქმნის ცდომილება და გარდაქმნის დრო.

ანალოგურ-ციფრული გარდამქმნელები ხასიათდებიან რეალიზაციის პრინციპებისა და სქემატოტექნიკური ვარიანტების მნიშვნელოვანი სხვადასხვაგვარობით.

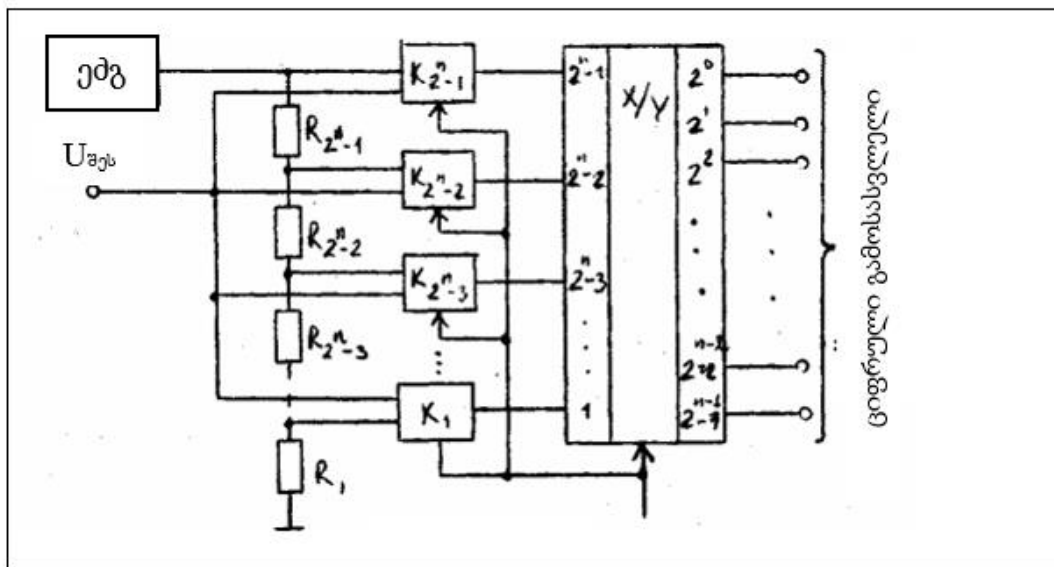
მათგან ყველაზე უფრო ფართო გავრცელება ჰპოვა შემდეგმა ანალოგურ-ციფრულმა გარდაქმნელებმა:

- ანალოგურ-ციფრული გარდამქმნელები პარალელური გარდაქმნის მეთოდებით (ნახ.10.2);
- ანალოგურ-ციფრული გარდამქმნელები მიმდევრობითი მიახლოების მეთოდებით (ნახ.10.3);
- ანალოგურ-ციფრული გარდამქმნელები ორმაგი ინტეგრირებით (ნახ.10.4).

ამათგან ყველაზე სწრაფქმედებს წარმოადგენს პარალელური გარდაქმნის ანალოგურ-ციფრული გარდამქმნელები, ყველაზე ზუსტს კი ორმაგი ინტეგრირების ანალოგურ-ციფრული გარდამქმნელები.

- პარალელური გარდაქმნის ანალოგიურ-ციფრული გარდამქმნელის სტრუქტურული სქემა მოცემულია ნახ.10.2-ზე. ის თავის შემადგენლობაში შეიცავს

ეტალონული ძაბვის წყაროს (გენერატორი) პრეციზიულ რეზისტორებზე შესრულებულ ძაბვის გამყოფით, კომპარატორების ნაკრებს (k_i) და კოდგარდამქმნელს (x/y).



ნახ.10.2

ანალოგური სიგნალი ედრება დაკვანტვის დონეებს. დაკვანტვის დონეები ტოლია

$$k_{დონ} = 2^n - 1,$$

სადაც n - ანალოგურ-ციფრული გარდამქმნელის ციფრული გამოსასვლელის თანრიგების რიცხვია.

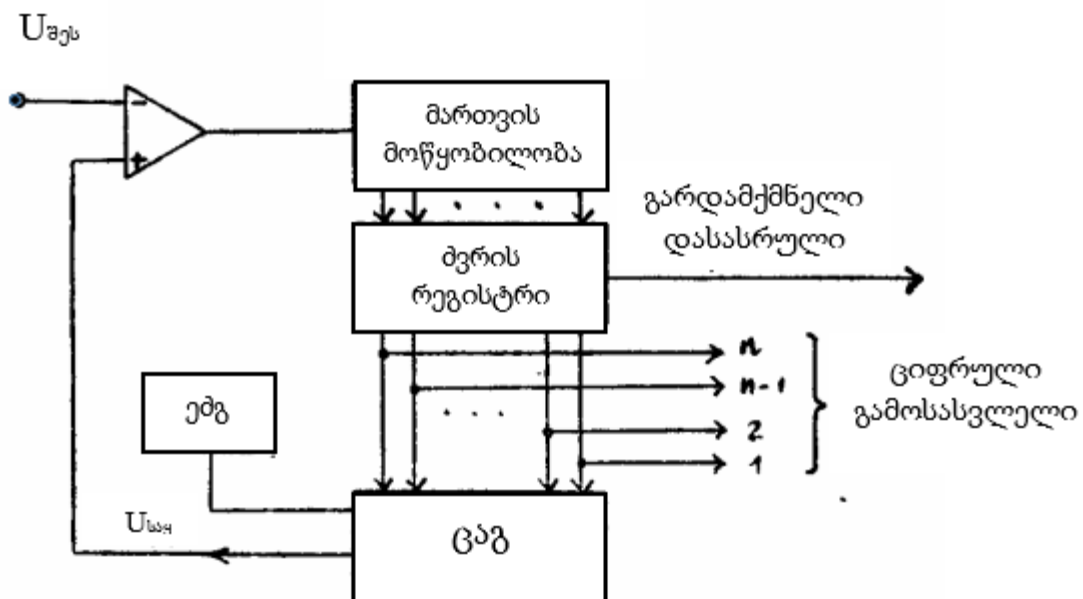
კომპარატორების გამოსასვლელზე ფორმირდება მაღალი დონის (ლოგიკური ერთიანი) სიგნალი თუ ანალოგური სიგნალი გადააჭარბებს მასთან მიმაგრებული დონეების მნიშვნელობას. სხვა დანარჩენ კომპარატორებზე გამოსასვლელი სიგნალი შეესაბამება ლოგიკურ ნულს. მიღებული ნულებისა და ერთიანების კომბინაცია კოდგარდამქმნელით გარდაიქმნება ორობით კოდში.

ანალოგურ-ციფრული გარდამქმნელის (აღკ) სწრაფქმედება განისაზღვრება კომპარატორის აღდგენის დროისა და კოდის გარდამქმნელის დაყოვნებით. პარალელური აკგ-ის შესაძლებლობა მოახდინოს ანალოგიური სიგნალების ათთანრიგა ორობით კოდში გარდაქმნის შესაბამისი სიჩქარით, საშუალებას იძლევა ეს გარდაქმნა განხორციელდეს 100 მეგაჰერცამდე სიხშირით.

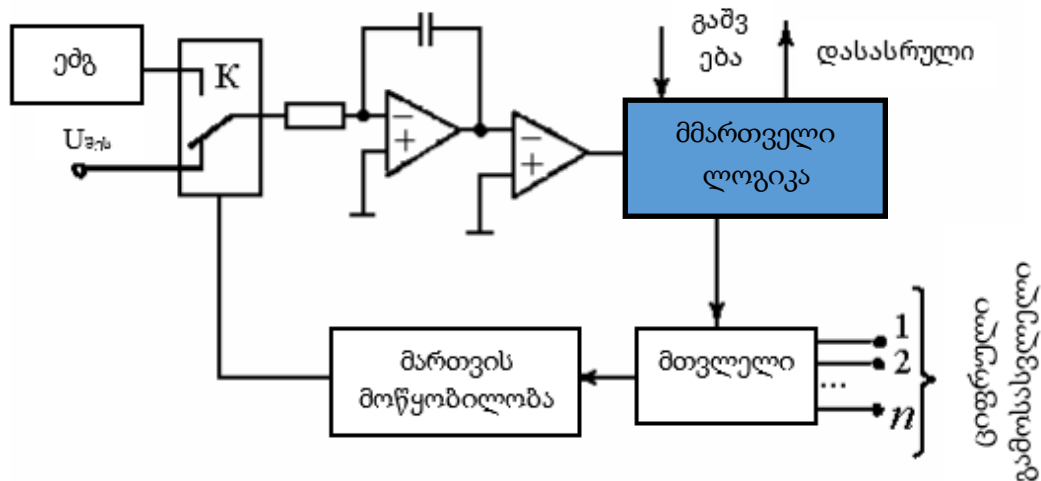
პარალელური აცგ-ების ნაკლოვან მხარეს მიეკუთვნება კომპარატორების რიცხვის მკვეთრი გაზრდა კოდის თანრიგობის უმნიშვნელოდ გაზრდის შემთხვევაში. მართლაც, თორმეტთანრიგა სწრაფქმედი აცგ-ის რეალიზაციისთვის მოითხოვება 4095 კომპარატორი. ასეთი აცგ-ების თანრიგობა ამ მიზეზის გამო არ აჭარბებს 10-ს. ცხადია, რომ ასეთი თანრიგობით შეუძლებელია მიღებულ იქნეს მაღალი ნებისდართვის უნარი და სიზუსტე.

ამ ნაკლოვანების აღმოსაფხვრელად გამოიყენებიან კომბინირებული აცგ-ები პარალელური-მიმდევრობითი გარდაქმნის ხერხით.

ანალოგიურ-ციფრული გარადამქმნელების სტრუქტურული სქემა, რომელიც აგებულია მიმდევრობითი მიახლოების პრინციპზე (ნახ.10.3), თავის თავში შეიცავს კომპარატორს - რეალიზებულს ოპერაციულ გამამდიერებელზე, ძვრის რეგისტრს, მართვის მოწყობილობას და ეტალონური ძაბვის წყაროს.



ნახ.10.3



ნახ.10.4

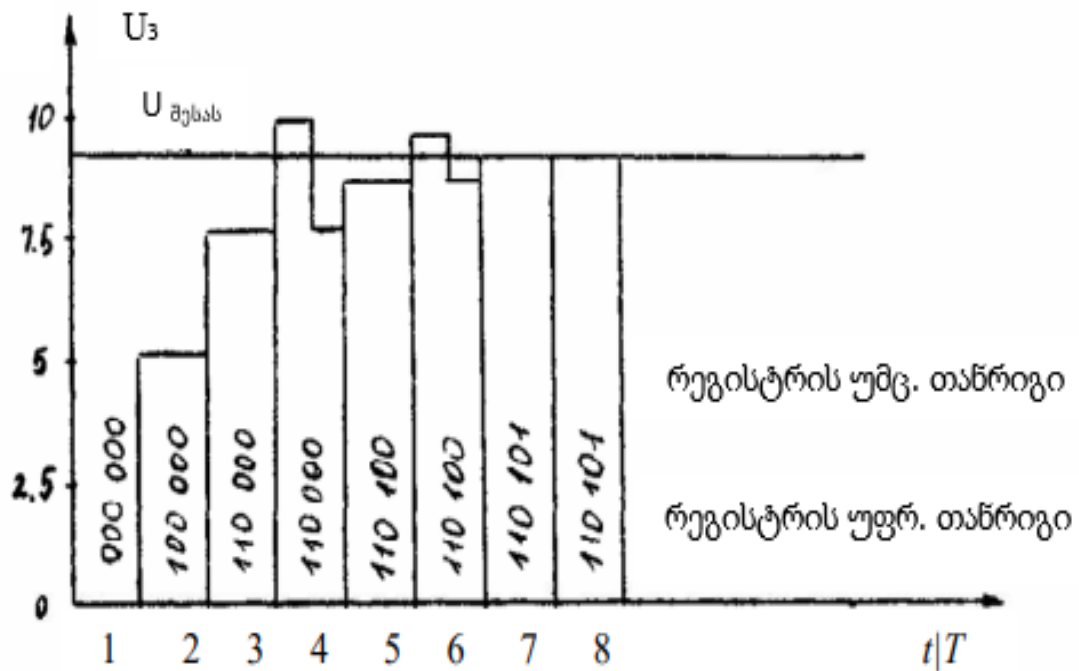
ასეთი აცგ-ის მოქმედების პრინციპი მდგომარეობს კომპარატორის საყრდენი ძაბვის $U_{სყ}$ თანმიმდევრობით ნაზრდში. ამ ძაბვას აფორმირებს ციფრულ-ანალოგური გარდამქმნელი, რომელიც ძვრის რეგისტრიდან თითოეულ ტაქტში ღებულობს ახალ ციფრულ კოდს.

ძვრის რეგისტრის ქცევა განისაზღვრება მართვის მოწყობილობით, რომელიც კომპარატორის გამოსასვლელის სიგნალით იძლევა რეგისტრის რეჟიმს შემდეგი ტაქტისთვის რეგისტრში ჩაწერილი რიცხვის ერთით გაზრდით ან შემცირებით უმცროს თანრიგში.

მაგალითის სახით განვიხილოთ 6 თანრიგა აცგ-ის მუშაობის პრინციპი, რომლის ფუნქციონირების დროითი დიაგრამა წარმოდგენილია ნახ.10.5-ზე. გამოსასვლელი რეგისტრის საწყისი მდგომარეობა ნულოვანია. გარდაქმნის ციკლი იწყება იმით, რომ რეგისტრის უფროს თანრიგში შეიტანება ერთიანი და ცაგ გამოიმუშავებს შესაბამის ძაბვას - 5 ვოლტი. რადგან შესასვლელი სიგნალის დონე არაა მიღწეული, ამიტომ ერთიანი შეიტანება მომდევნო თანრიგში. ცაგ-ის გამოსასვლელზე მეორე ტაქტში ჩნდება საყრდენი ძაბვის ნაზრდი - ტოლი 2,5 ვოლტის (ე.ი. რეგისტრის უფროსი თანრიგის ძაბვის ნახევარი). თუმცა ძაბვა აცგ-ის გამოსასვლელზე ისევ ნაკლებია შესასვლელ ძაბვაზე, ამიტომ რეგისტრის მესამე თანრიგში შევიტანთ ერთიანს. აცგ-ის გამოსასვლელზე ძაბვა ამაღლდა 1,25 ვოლტით და გადააჭარბა შესასვლელ დონეს.

კომპარატორი გადაირთო, ამიტომ ამ სიგნალით მართვის მოწყობილობამ გაანულა მესამე თანრიგი.

მეოთხე ტაქტში ერთიანი შეიტანება მეოთხე თანრიგში და ა.შ. ექვსი ტაქტის შემდეგ ანალოგური გარდამქმნელის გამოსასვლელზე დადგება ციფრული კოდი - შესასვლელი ძაბვის შესაბამისი აცგ-ის ნებისდამრთველი უნარის გათვალისწინებით.



ნახ.10.5

ასეთი აცგ-ის სწრაფქმედება განისაზღვრება დროით, რომელიც იხარჯება n სინჯზე. სინჯის ხანგრძლივობა, ძირითადად, განისაზღვრება ცაგ-ის დაყენების დროით $t_{დაყ}$

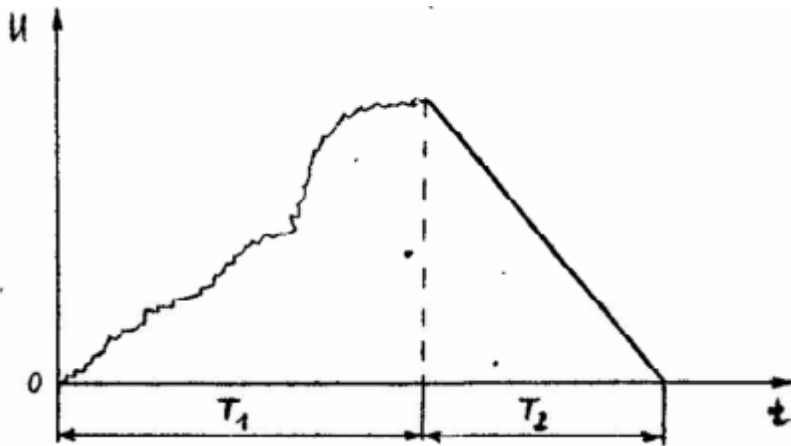
ე.ი.
$$T_{სინჯ} = n t_{დაყ}.$$

შესაბამისად, ათთანრიგიანი აცგ, აგებული ცაგ-ის გამოყენებით, რომელსაც გააჩნია $t_{დაყ} = 3 \dots 5$ მკწმ, იმუშავებს გარდაქმნის არაუმეტეს 20...30 კპც სიხშირით. აცგ-ის სწრაფქმედების ამაღლება მიმდევრობითი მიახლოებით შესაძლებელია მხოლოდ ცაგ-ის სწრაფქმედების ამაღლების გზით.

ორმაგი ინტეგრირების პრინციპით აგებული აცგ (ნახ.10.4) შედგება ინტეგრატორის, კომპარატორის, მთვლელის, მართვის მოწყობილობის და ეტალონური

ძაბვის წყაროსგან. მისი მუშაობის პრინციპი მდგომარეობს შემდეგში. დასაწყისში T_1 დროის განმავლობაში (ნახ.10.6) ხორციელდება შემავალი სიგნალის ინტეგრირება მანამ, სანამ მთვლელი არ გაგზავნის გადავსების სიგნალს მართვის მოწყობილობაში, რომელიც ინტეგრატორს გადართავს ეტალონური ძაბვის წყაროზე.

C კონდესატორი დაიწყებს განმუხტვას. მთვლელი, რომელშიც გადავსების შემდეგ დამყარდა ნულოვანი მდგომარეობა, კონდესატორის გადამუხტვის



ნახ.10.6

პროცესის დაწყებიდან მუშაობას განაგრძობს ზუსტად ისევე, როგორც T_1 დროის განმავლობაში, მხოლოდ იმ განსხვავებით, რომ ახლა ის კი არ მართავს, არამედ თან სდევს C კონდესატორის განმუხტვის პროცესს. კონდესატორის განმუხტვის დამთავრებისას ამუშავდება კომპარატორი და მართვის ლოგიკის საშუალებით გააჩერებს მთვლელს. მთვლელის გამოსასვლელზე კოდი პროპორციული იქმნება T_1 დროის ინტეგრალში ანალოგიური სიგნალის საშუალო მნიშვნელობისა. ასეთ აცგ-ის გააჩნიათ დაბალი სწრაფქმედება. მათი გამოყენების ძირითადი სფეროა პრეციზიული მოწყობილობები სიგნალების დამუშავების ძალიან დაბალი სიხშირით. ამრიგად, ძირითად ელემენტებს, მიუხედავად იმისა თუ როგორი სქემით არიან ისინი აგებული, წარმოადგენს: გასაღების სქემები, კომპარატორები, ეტალონური ძაბვის გენერატორები (წყაროები), დამგროვებელი მოწყობილობა, კოდის გარდამქმნელი და მმართველი ლოგიკა, რომელიც განსაზღვრავს ინფორმაციის ანალოგურ-ციფრული გარდამქმნელების ფუნქციონირების წესრიგს.

მაინტეგრირებელი აცგ-ის ასაგებად გამოიყენება ორმაგი ინტეგრირების პრინციპი.

ამ პრინციპის დროს თავდაპირველად რაღაც ფიქსირებული T_1 დროის ინტერვალში ინტეგრირებას განიცდის გარდასაქმნელი U_x ანალოგური სიდიდე, ხოლო შემდეგ ინტეგრირდება საპირისპირო პოლარობის მქონე ეტალონური (საყრდენი) ძაბვა $U_{საყ}$ (ნახ.10.7). მეორე ინტეგრირების დროით საყრდენი ინტერვალი T_1 გარდასაქმნელი U_x სიდიდის პროპორციულია. T_2 დროითი ინტერვალი კი დამოკიდებულია, როგორც



ნახ.10.7

$T_1/U_{საყ}$ მუდმივ სიდიდეზე, ისე U_x გარდასაქმნელ ძაბვაზე და არ არის დამოკიდებული ინტეგრატორის პარამეტრებზე.

ორმაგი ინტეგრირების მეთოდით მომუშავე ანალოგურ-ციფრული გარდამქმნელი უზრუნველყოფს გარდაქმნის მაღალ სიზუსტეს ტემპერატურის დიდ დიაპაზონში და ამიტომ იგი ფართოდ გამოიყენება საზომ სისტემებში და ფრენის მართვის და კონტროლის ავტომატიზებულ სისტემებში.

10.2. ციფრულ-ანალოგური გარდამქმნელები

ციფრულ-ანალოგური გარდამქმნელი წარმოადგენს ციფრული მონაცემების ანალოგურ სიგნალში გადამყვან მოწყობილობას.

ციფრულ-ანალოგური გარდამქმნელი ორობითი კოდის სახით მოცემულ ციფრულ სიგნალს გარდაქმნის ანალოგურ სიგნალად, რადგან ნებისმიერი n თანრიგა ორობითი N რიცხვის წარმოადგენა შეიძლება 2-ის ხარისხის ჯამის მიხედვით

$$N = 2^{n-1} \cdot a_{n-1} + 2^{n-2} \cdot a_{n-2} + \dots 2^k \cdot a_k + \dots 2^0 \cdot a_0 = \sum_{i=0}^{n-1} 2^i \cdot a_i,$$

სადაც a_i კოდის შესაბამისი თანრიგის ციფრია (0 ან 1), ამიტომ ორობითი რიცხვის ანალოგურ სიგნალად გარდაქმნა დაიყვანება თანრიგების შესაბამისი ეტალონური სიდიდეების (მაზვის, დენის) შეკრებაზე. საერთო ჯამში, რომელიმე ეტალონის შეყვანა ან არშეყვანა დამოკიდებულია შესაბამისი თანრიგის ციფრზე (1-ის შემთხვევაში შედის, ხოლო 0-ის შემთხვევაში არ შედის).

ცაგ-ის გამომავალ სიგნალს შეიძლება ჰქონდეს დენის ან მაზვის ფორმა. დენის გამოსასვლელიანი გარდამქმნელები ძირითადად გამოიყენება პრეციზიულ სქემებში. მაზვის გამოსასვლელიან გარდამქმნელებში ციფრული ორობითი კოდის დაბალი ნიშნადი თანრიგის (დნთ) შესაბამისი მაზვის მნიშვნელობა განისაზღვრება საყრდენი მაზვის მნიშვნელობის გაყოფით 2^n -ზე, სადაც n ორობითი კომბინაციის თანრიგების რაოდენობაა.

თანრიგობა (N) წარმოადგენს ციფრულ კოდში შემავალ თანრიგების რაოდენობას. თანრიგების რაოდენობის მიხედვით გამოყოფენ ამაღლებული სიზუსტის გარდამქმნელებს (დიდი თანრიგობა, $N \geq 14$) და მაღალი სწრაფქმედების გარდამქმნელებს (6-8 თანრიგი).

გარჩევითობა არის დაბალი ნიშნადი თანრიგის შესაბამისი მაზვის მნიშვნელობა. იგი დამოკიდებულია თანრიგების რაოდენობაზე და განსაზღვრავს სიგნალის გარდაქმნის სიზუსტეს.

დისკრეტიზაციის სიხშირე ის მაქსიმალური სიხშირეა, რომელზეც შეუძლია მუშაობა ცაგ-ს და გასცემს გამოსასვლელზე კორექტულ შედეგს. კოტელნიკოვის თეორემის თანახმად, ციფრული ფორმიდან ანალოგური სიგნალის აღსაწარმოებლად აუცილებელია, რომ სიგნალის სპექტრის მაქსიმალური სიხშირის გაორმაგებული მნიშვნელობა არ აღემატებოდეს დისკრეტიზაციის სიხშირეს.

სრული სკალის დიაპაზონი წარმოადგენს გამოსასვლელი სიგნალის მნიშვნელობათა დიაპაზონს.

ციფრულ-ანალოგური გარდაქმნელები გამოიყენებიან სიგნალის ფორმირებისათვის ძაბვის ან დენის სახით, რომელიც ფუნქციურად დაკავშირებულია მმართველ კოდთან. ყველაზე უფრო ხშირად ანალოგურ-ციფრული გარდაქმნელები (აცგ) გამოიყენებიან ანალოგურ-ციფრულ გარდაქმნელებში სიგნალების ციფრული დამუშავების მოწყობილობების შეუღლებისათვის ანალოგური სიგნალებით მომუშავე სისტემებთან. გარდა ამისა, ციფრული-ანალოგური გარდაქმნელები გამოიყენებიან როგორც საშუალება ანალოგურ-ციფრულ გარდაქმნელებში უკუკავშირის კვანძების განხორციელებისა და ასევე ციფრული სიდიდეების ანალოგურ სიდიდეებთან შედარების მოწყობილობებში.

ციფრული-ანალოგური გარდაქმნელების კლასიფიკაცია შეიძლება მოხდეს სხვადასხვა ნიშან-თვისებების მიხედვით: მოქმედების პრინციპის, გამოსასვლელი სიგნალის სახის, გამოსავლელი სიგნალის პოლარობის, ელემენტური ბაზის და სხვ. მოქმედების პრინციპის მიხედვით ყველაზე უფრო მეტი გამოყენება ჰპოვეს შემდეგი სახის ციფრული-ანალოგურმა გარდაქმნელებმა:

- დენების შეკრებით;
- ძაბვის გაყოფით;
- ძაბვის შეკრებით.

მიკროელექტრონულ შესრულებაში გამოიყენება მხოლოდ პირველი ორი ტიპი.

გამოსასვლელი სიგნალის სახის მიხედვით ცაგ-ები იყოფიან ორ სახედ:

- დენის მიხედვით გამოსასვლელით;
- ძაბვის მიხედვით გამოსასვლელით.

ციფრული-ანალოგური გარდაქმნელის გამოსასვლელი დენის გარდაქმნისათვის ძაბვაში, ჩვეულებრივ გამოიყენება ოპერაციული გამაძლიერებლები. გამოსასვლელი სიგნალის პოლარობის მიხედვით ციფრულ-ანალოგური გარდაქმნელების დაყოფა ხდება ერთპოლარულ და ორპოლარულ ციფრულ-ანალოგურ გარდაქმნელებად.

მმართველი კოდი, რომელიც მიეწოდება ცაგ-ის შესასვლელზე შეიძლება იყოს სხვადასხვანაირი: ორობითი, ორობით-ათობითი, გრეის, უნიტარული და სხვ. გარდა

ამისა, შეიძლება სხვადასხვანაირი იყოს ასევე ციფრული-ანალოგური გარადამქმნელის შესასვლელზე მიწოდებული ლოგიკური სიგნალების დონეები.

მმართველი კოდის ზემოქმედების შედეგად ცაგ-ის გამოსასვლელი ძაბვის ფორმირებისას ჩვეულებრივ გამოიყენება საყრდენი ძაბვის წყაროები. იმისდა მიხედვით თუ როგორი სახის საყრდენი ძაბვა იქნება გამოყენებული ცაგ-ში, მათ ყოფენ ორ ჯგუფად:

- მუდმივი საყრდენი ძაბვით;
- ცვალებადი საყრდენი ძაბვით.

გარდა ამისა, ცაგ-ს კლასიფიკაციას ახდენენ შემდეგი ძირითადი მახასიათებლების მიხედვით: თანრიგების რაოდენობის, სწრაფქმედების, გარდაქმნის სიზუსტის და მოხმარებული სიმძლავრის მიხედვით.

ცაგ-ების ყველა პარამეტრი შეიძლება დაიყოს ორ ჯგუფად: სტატიკური და დინამიკური. სტატიკურ პარამეტრებს მიეკუთვნება: ნებისდართვის უნარი, გარდაქმნის ცდომილება, გამოსასვლელი სიგნალის მნიშვნელობათა დიაპაზონი, მმართველი კოდის მახასიათებლები, ნულოვანი დონის წანაცვლება და სხვა.

ციფრულ-ანალოგური გარდამქმნელის დინამიკურ მაჩვენებლებს მიეკუთვნება: გამოსასვლელი სიგნალის დაყენების დრო, გარდაქმნის ზღვრული სიხშირე, დინამიკური ცდომილება. განვიხილოთ ზოგიერთი ამ პარამეტრებიდან.

აცგ-ის ნებისდართვის უნარი განისაზღვრება როგორც გამოსასვლელი სიგნალის გრადაციის მაქსიმალური რაოდენობის შებრუნებული სიდიდე. ასე მაგალითად, თუ ცაგ-ის ნებისდართვის უნარი შეადგენს 10^5 -ს, ეს იმას, ნიშნავს, რომ გამოსასვლელი სიგნალის გრადაციის მაქსიმალური რაოდენობა (რიცხვი) ტოლია 10^5 -ის. ხანდახან ცაგ-ის ნებისდამრთველი უნარის შეფასებას ახდენენ გამოსასვლელი ძაბვის მიხედვით, როცა ხდება შესასვლელი კოდის უმცროსი თანრიგის ცვლილება ერთით, ე.ი. დაკვანტვის ბიჯით. ბუნებრივია, რომ რაც უფრო დიდი იქნება ცაგ-ის თანრიგობა, მით უფრო მაღალი იქნება მისი ნებისდამრთველობის უნარი.

ცაგ-ის გარდაქმნის ცდომილება შეიძლება დაიყოს დიფერენციალურ და არახაზოვან ცდომილებებად. ცაგ-ის შესასვლელზე კოდის ზრდასთან ერთად იზრდება

გამოსასვლელი ძაბვაც, თუმცა ძაბვის გაზრდისას შესაძლებელია წარმოიშვას გადახრები ხაზოვანი დამოკიდებულებებისგან. არახაზოვნობის (არაწრფივობის) ცდომილებას უწოდებენ გამოსასვლელი ძაბვის მაქსიმალურ გადახრას იდეალური წრფისაგან გარდაქმნის მთელ დიაპაზონში.

დიფერენციალურ ცდომილებას უწოდებენ შესასვლელი კოდის ორ მეზობელ მნიშვნელობებისთვის წრფივობიდან მაქსიმალურ გადახრას.

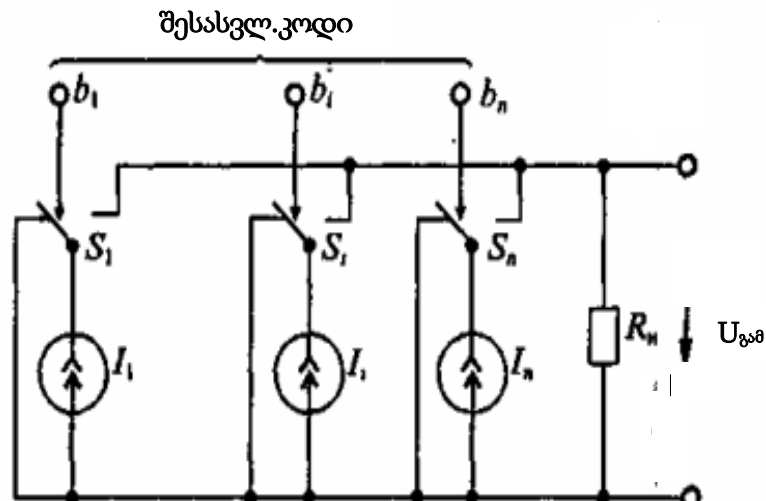
ნულის წანაცვლების ძაბვა განისაზღვრება გამოსასვლელი ძაბვით იმ შესასვლელი კოდის შემთხვევაში, რომელიც შეესაბამება ნულოვან მნიშვნელობას.

დაყენების დრო $t_{დაყ}$ - ეს არის დროის ინტერვალი შესასვლელი კოდის მიწოდებიდან გამოსასვლელი სიგნალის მოცემულ საზღვრებში შესვლამდე, რომელიც განსაზღვრულია ცდომილებით.

გარდაქმნის მაქსიმალური სიხშირე ეს არის დისკრეტიზაციის უდიდესი სიხშირე, რომლის დროსაც ცაგ-ის ყველა პარამეტრი შეესაბამება მოცემულ (დადგენილ) მნიშვნელობებს.

ცაგ-ები პარამეტრების ერთობლიობის მიხედვით მიღებულია დაიყოს სამ ჯგუფად: საერთო დანიშნულების, პრეციზიული და სწრაფქმედები. სწრაფქმედი ცაგ-ების დაყენების დრო 100 ნანოწამზე ნაკლებია. პრეციზიულ ცაგ-ს მიეკუთვნებიან ის ცაგ-ები, რომლებსაც გააჩნიათ არაწრფივობის ცდომილება 0,1%-ზე ნაკლები.

არსებობს რამდენიმე სქემა, რომელიც წარმოადგენს მრავალი სხვადასხვაგვარი ციფრულ-ანალოგური გარდამქმნელების აგების ბაზას. გამოსასვლელი ძაბვის შესაბამისი დონეების ფორმირებისათვის (ან დენის ფორმირებისათვის) ციფრულ-ანალოგური გარდამქმნელის გამოსასვლელს უერთდება საყრდენი სიგნალების $E_1, E_2, \dots, E_n$ (ან დენების $I_1, I_2, \dots, I_n$) აუცილებელი რიცხვი, ან ადგენენ დაყოფის კოეფიციენტის $K_1, K_2, \dots, K_n$ შესაბამის დისკრეტულ მნიშვნელობას.



ნახ.10.8. ცაგ-ის გამარტივებული სქემა დენების შეჯამებით

ციფრულ-ანალოგური გარდამქმნელის სქემა დენების შეჯამებით მოყვანილია ნახ.10.8-ზე. ამ სქემაში გამოიყენება n რაოდენობის საყრდენი დენის წყაროები $I_1, I_2, \dots, I_n$. შესასვლელი კოდი $b_1, b_2, \dots, b_n$ მართავს გასაღებებს $S_1, S_2, \dots, S_n$, რომლებიც დენის წყაროს მიუერთებენ დატვირთვას, ან შეკრავენ მათ მოკლედ ჩართვაზე. ამასთან, თუ $b_i=0$, მაშინ შესაბამისი წყარო დამოკლებულია და სქემის მუშაობაში მონაწილეობას არ იღებს. თუ $b_i=1$, მაშინ შესაბამისი დენის წყარო მიერთებულია დატვირთვასთან. რეზულტირებული დენი ტოლია საყრდენი წყაროების დენების ჯამისა, რომლებისთვისაც $b_i=1$. ძაბვა გამოსასვლელზე ტოლი იქნება რეზულტირებული დენი I_{Σ} გამრავლებული დატვირთვის წინააღობაზე $R_{\text{დატ}}$, ე.ი.

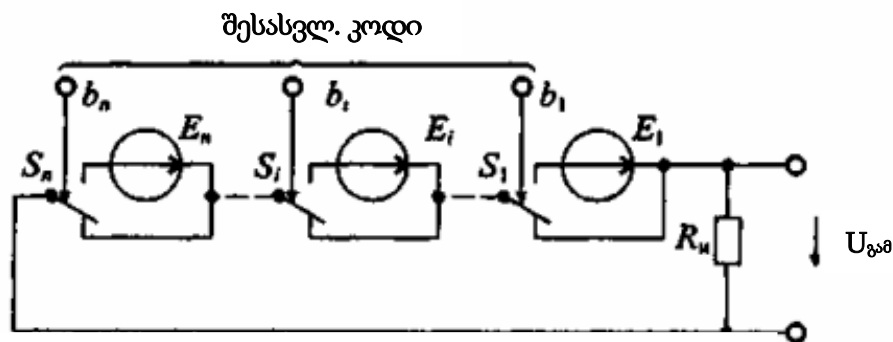
$$U_{\text{გამ}} = I_{\Sigma} R_{\text{დატ}} .$$

ასე, მაგალითად, თუ შესასვლელი კოდი წარმოადგენს ორობითს, მაშინ ჯამური დენი განისაზღვრება გამოსახულებით:

$$I_{\Sigma} = I_0(b_1 2^{n-1} + b_2 2^{n-2} + \dots + b_n 2^0) = I_0 N,$$

სადაც n - შესასვლელი კოდის ორობითი თანრიგების რიცხვია, N - კი n თანრიგა ციფრული სიტყვაა.

ციფრულ-ანალოგური გარდამქმნელის გამარტივებული სქემა ძაბვების შეჯამებით მოცემულია ნახ.10.9-ზე.



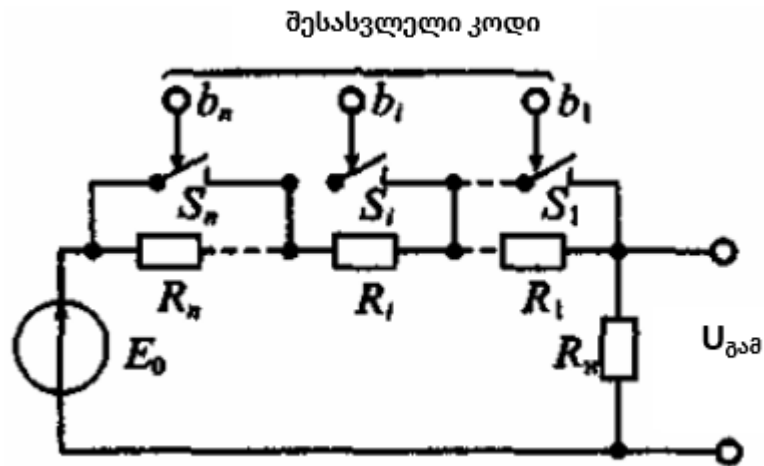
ნახ.10.9. ცაგ-ის გამარტივებული სქემა ძაბვების შეჯამებით

ამ სქემაში გამოიყენება n რაოდენობის საყრდენი წყაროები $E_1, E_2, \dots, E_n$.

შესასვლელი კოდით ხდება $S_1, S_2, \dots, S_n$ გასაღებების მართავა, რომლებიც საყრდენის ძაბვის შესაბამის წყაროებს ან მიუერთებენ დატვირთვას, ან გამორთავენ მათ. ისე როგორც დენების შეჯამების სქემის შეთხვევაში, როცა $b_i=1$ ძაბვის შესაბამისი წყარო ჩართულია, ხოლო როცა $b_i=0$ - გამორთულია. ჯამური (რეზულტირებული) ძაბვა გამოსასვლელზე ტოლია ჩართული საყრდენი წყაროების ძაბვების ჯამისა.

ასე მაგალითად, შესასვლელი ორობითი კოდისათვის გამოსასვლელი ძაბვა გამოითვლება ფორმულით

$$U_{\Sigma} = U_{\text{გაგ}} = U_0 (b_1 2^{n-1} + b_2 2^{n-2} + \dots + b_n 2^0) = U_0 N.$$



ნახ.10.10. ცაგ-ის გამარტივებული სქემა ძაბვის გამყოფით

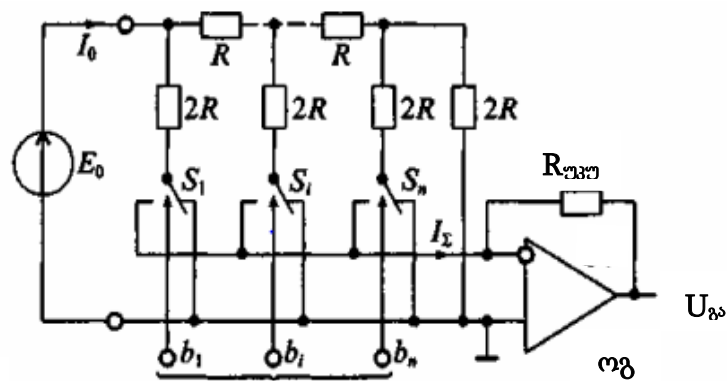
ციფრულ-ანალოგური გარდამქმნელის გამარტივებული სქემა საყრდენი E_0 ძაბვის გაყოფით მოყვანილია ნახ.10.10-ზე. ამ სქემაში არსებობს ერთი საყრდენი ძაბვის წყარო და დაკალიბრებული წინაღობების $R_1, R_2, \dots, R_n$ -ის ნაკრები, რომელთა დახმარებით საყრდენი წყაროს ძაბვა შესაძლებელია დაიყოს იმ მნიშვნელობამდე, რომელიც შესასვლელი კოდის შესაბამისი ქნება.

გამოსასვლელი ძაბვა ამ სქემისათვის გამოითვლება ფორმულით

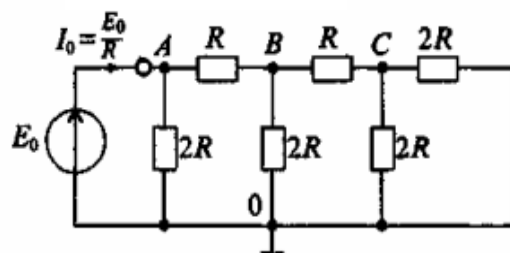
$$U_{\text{გამ}} = E_0 \frac{R_{\text{დატ}}}{R_{\text{ჯ}} + R_{\text{დატ}}},$$

სადაც $R_{\text{დატ}}$ - დატვირთვის წინაღობა, $R_{\text{ჯ}}$ -ჯამური წინაღობაა დაყენებული $S_1, S_2, \dots, S_n$ გასაღებებით, რომლებიც თავის მხრივ იმართებიან შესასვლელი კოდით.

როცა $R_{\text{დატ}}=0$, მაშინ ეს სქემა გადაიქცევა მართვად დენის წყაროდ, ე.ი. ის მუშაობს ზუსტად ისე, როგორც სქემა დენების შეჯამებით. პრაქტიკულად $R_{\text{დატ}} = 0$ -ის შესრულება შესაძლებელია პარალელური უკუკავშირის ოპერაციული გამაძლიერებლის დახმარებით.



შესასვლ. კოდი



ნახ.10.11. ცაგ-ის სქემა დენების შეჯამებით $R-2R$ ტიპის რეზისტორული მატრიცის სტრუქტურით

ნახ.10.11-ზე მოყვანილია ციფრულ-ანალოგური გარდამქმნელის პრაქტიკული სქემა დენების შეჯამებით, რომელშიც გამოიყენება საყრდენი ძაბვის E_0 -ის ერთი წყარო და რეზისტული მატრიცა $R-2R$. რეზისტული მატრიცის განსაკუთრებულობა მდგომარეობს იმაში, რომ გასაღებების $S_1, S_2, \dots, S_n$ -ის ნებისმიერ მდებარეობაში მატრიცის შესასვლელი წინაღობა ყოველთვის ტოლია R , შესაბამისად კი, დენი რომელიც გადის მატრიცაში, ტოლია $I_0 = E_0/R$. შემდეგში ის მიმდევრობით A, B, C კვანძებში იყოფა ორობითი კანონის მიხედვით. დენების განაწილების ორობითი კანონი რეზისტული მატრიცის შტოებში შენარჩუნებულია დატვირთვის წინაღობის ნულოვანი მნიშვნელობის პირობის შესრულებისას. რადგან რეზისტული მატრიცის დატვირთვას წარმოადგენს ოპერაციული გამაძლიერებელი (ოგ), რომელიც აღჭურვილია უარყოფითი უკუკავშირით $R_{\text{უკ}}$ წინაღობის საშუალებით, ამიტომ მისი (ოპერაციული გამაძლიერებლის) შესასვლელი წინაღობა საკმარისად მაღალი სიზუსტით ტოლია ნულის.

ოპერაციული გამაძლიერებლის გამოსასვლელზე ძაბვა განისაზღვრება გამოსახულებით

$$U_{\text{გამ}} = E_0 R_{\text{უკ}} (b_1 2^{n-1} + b_2 2^{n-2} + \dots + b_n 2^0) / R 2^n = E_0 R_{\text{უკ}} N / R 2^n, \quad (11.1)$$

სადაც $b_i=1$, მაშინ, როცა S_i გასაღები იმყოფება მდებარეობაში, რომლის დროსაც დენი გადის ოპერაციული გამაძლიერებლის მაინვერსირებელ შესასვლელზე, და $b_i=0$, თუ გასაღები S_i იმყოფება მდებარეობაში, რომლის დროსაც დენი გადის საერთო გამომყვანში, n -არის გარდამქმნელის თანრიგების რიცხვი.

გამოსასვლელი ძაბვის მაქსიმალური მნიშვნელობა (ე.ი. ძაბვა დიაპაზონის ბოლო წერტილში) სქემაში ფიქსირდება მაშინ როცა ყველა $b_i=1$ და განისაზღვრება შემდეგი ფორმულით:

$$U_{\text{გამ მაქს}} = E_0 R_{\text{უკ}} (1-2^{-n}) / R = E_0 R_{\text{უკ}} / R - h ,$$

სადაც h -დაკვანტვის ბიჯია, ე.ი. გამოსასვლელი ძაბვის ნაზრდი, როცა შესასვლელი კოდის უმცროსი თანრიგი ერთით იცვლება ტოლია:

$$h = E_0 R_{\text{უკ}} / 2^n R.$$

როგორც (11.1) ფორმულიდან ჩანს ციფრულ-ანალოგური გარდამქმნელის გამოსასვლელი ძაბვა დამოკიდებულია არა მარტო N შესასვლელ კოდზე, არამედ ასევე E_0 საყრდენი წყაროს ძაბვაზე. თუ დავუშვებთ რომ E_0 ძაბვა იცვლება, მაშინ ციფრულ-ანალოგური გარდამქმნელის შესასვლელი ძაბვა იქნება ორი სიდიდის ნამრავლის პროპორციული, კერძოდ შესასვლელი კოდისა და ძაბვისა, რომელიც მიეწოდება საყრდენი სიგნალის შესასვლელზე. ამასთან დაკავშირებით ასეთ ციფრულ-ანალოგურ გარდამქმნელებს გადამამრავლებელ ციფრულ-ანალოგური გარდამქმნელებს უწოდებენ. ინტეგრალურ მიკროსქემებში საყრდენი ძაბვის წყარო არ არსებობს, თუმცა არსებობს შესასვლელი მისი მიერთებისათვის.

ანალოგურ-ციფრული გარდამქმნელების ინტეგრალურ მიკროსქემებს გააჩნიათ ასოითი აღნიშვნა ПВ, მაგალითად K572ПВ2, რაც მიუთითებს იმაზე, რომ საქმე გვაქვს ორმაგი ინტეგრირების მიკროსქემასთან შვიდსეგმენტიან შუქდიოდურ ინდიკატორზე გამოსასვლელით.

ციფრულ-ანალოგურ გარდამქმნელების მიკროსქემები აღნიშნებიან ასოებით ПА, მაგალითად K572ПА2А, რაც მიუთითებს იმაზე, რომ აღნიშნული ტიპის ციფრულ-ანალოგური გარდამქმნელი არის თორმეტანრიგიანი ციფრულ - ანალოგური გარდამქმნელი.

ლექცია 11

საბორტო კომპიუტერების ექსპლუატაცია და მუშაობისუნარიანობის კონტროლი

11.1. საბორტო კომპიუტერების ექსპლუატაცია

თანამედროვე საბორტო კომპიუტერების ექსპლუატაციის ორგანიზაცია თავის თავში შეიცავს: ტექნიკური პერსონალის სწავლებას და ცოდნის მუდმივ შემოწმებას; საბორტო კომპიუტერის მუშაობის აღრიცხვას და ტექნიკური დოკუმენტაციის წარმოებას; საბორტო კომპიუტერის და მთლიანად სისტემის სწორი ფუნქციონირების რეგულარული კონტროლის წარმართვას; საბორტო კომპიუტერის რემონტის ჩატარებას. საბორტო კომპიუტერი უნდა იმყოფებოდეს მუშაობისათვის მუდმივ მზადყოფნაში, რაზეც პასუხისმგებლობა ეკისრებათ ინჟინერ – ტექნიკურ პერსონალს და საფრენოსნო ეკიპაჟს.

საბორტო კომპიუტერის ექსპლუატაცია რთული პროცესია რომელიც დაკავშირებულია მომსახურე პერსონალის მრავალმხრივ მოდვაწეობასთან. საბორტო კომპიუტერის მუშაობის ძირითადი დრო იხარჯება სხვადასხვა ტიპის ამოცანების გადაწყვეტაზე. საბორტო კომპიუტერის ექსპლუატაცია განუხრელადაა დაკავშირებული უწყისივრობების მოძებნაზე და მათ აღმოფხვრაზე.

საბორტო კომპიუტერის ექსპლუატაციის ძირითადი ეტაპებია:

- ფრენისას საბორტო კომპიუტერის შემადგენლობაში ექსპლუატაცია;
- საბორტო კომპიუტერის ექსპლუატაცია მიწაზე, როცა საჰაერო ხომალდზე სრულდება ყველა სახის სამუშაოს ჩატარება ფრენისწინა ეტაპების მომზადებისა და შესრულებისათვის.

საბორტო კომპიუტერის ექსპლუატაციის ყველაზე უფრო საპასუხისმგებლო ეტაპს წარმოადგენს მისი მუშაობა ფრენისას საბორტო მოწყობილობის შემადგენლობაში. ამ ეტაპზე საბორტო კომპიუტერი მუშაობს როგორც მმართველი კომპიუტერი, რომელიც უზრუნველყოფს ამოცანების გადაწყვეტას დროის რეალურ მასშტაბში.

საბორტო კომპიუტერებში ფართოდაა განვითარებული კონტროლის სხვადასხვა მეთოდები, რომლებიც საშუალებას იძლევა საბორტო კომპიუტერში დროულად იქნეს

აღმოჩენილი წარმოშობილი უწყესივრობები და ამით შესაძლებლობა მისცეს ეკიპაჟს შეცვალოს ფრენის რეჟიმი.

საბორტო კომპიუტერის მუშაობის კონტროლი სრულდება ექსპლუატაციის ყველა ეტაპზე. თუ საბორტო კომპიუტერში წარმოიშვა უწყესივრობა, რომელიც აღმოჩენილ იქნა, მაგალითად, ტესტური კონტროლის პროგრამით, მაშინ საპილოტაჟო-სანავიგაციო კომპლექსის შესაბამის პულტზე გაიცემა სიგნალი „საბორტო კომპიუტერის მტყუნება“. ამასთან ტესტური კონტროლის პროგრამის შესრულება ციკლურად გრძელდება.

საბორტო კომპიუტერში უწყესივრობის თვითაღმოფხვრის შემთხვევაში სიგნალი „საბორტო კომპიუტერის მტყუნება“ მოიხსნება და კომპიუტერი აგრძელებს ამოცანების პროგრამების შესრულებას ახალი საწყისი პირობების დროს.

„საბორტო კომპიუტერის მტყუნება“ სიგნალის ფორმირების სისტემა ჩვეულებრივ ისე რეალიზდება, რომ იმ შემთხვევაშიც კი, როცა ტესტური კონტროლის პროგრამის შესრულება შეუძლებელია საბორტო კომპიუტერის რაღაც კვანძის მწყობრიდან გამოსვლის გამო, კომპიუტერი სიგნალს მტყუნების შესახებ მაინც გასცემს. ამისათვის პროგრამის სწორად შესრულების თითოეული ციკლის ბოლოს სქემურად გაიცემა სიგნალი, რომელიც კრძალავს შეტყობინების „საბორტო კომპიუტერის მტყუნება“ ინდიკაციას. შესაბამისად, მთელი იმ დროის განმავლობაში, სანამ ტესტური კონტროლის პროგრამა სრულდება სწორად სიგნალი „საბორტო კომპიუტერის მტყუნება“ არ გაიცემა. საბორტო კომპიუტერში უწყესივრობების გაჩენისას ტესტური კონტროლის პროგრამა ან აღმოაჩენს უწყესივრობის ფაქტს ან რაღაც მიზეზების გამო შეწყვეტს შესრულებას. ამასთან აკრძალვის სიგნალის ფორმირება არ ხდება და ინდიკაციის სქემა საინფორმაციო ტაბლოზე გასცემს წარწერას „საბორტო კომპიუტერის მტყუნება“.

საბორტო კომპიუტერის მუშაობის უნარიანობის უზრუნველყოფის მიზნით ექსპლუატაციის ყველა ეტაპზე სრულდება ღონისძიებათა კომპლექსი ფრენების მომზადების, რეგლამენტური სამუშაოების ჩატარების და შენახვის ეტაპებისთვის.

ფრენების მომზადებისას საბორტო კომპიუტერზე შესასრულებელი სამუშაოების ძირითადი სახეები.

საბორტო კომპიუტერის მომზადება ფრენებისთვის ორგანიზდება საჰაერო ხომალდის ტექნიკური მომსახურების რეგლამენტის შესაბამისად. რეგლამენტი ადგენს საჰაერო ხომალდის ფრენის დროს მუშაობისთვის ყველა სისტემის მომზადების წესრიგს.

ყველა სახის მომზადება ხორციელდება საბორტო კომპიუტერზე, რომელიც დაყენებულია საჰაერო ხომალდზე. ამასთან საბორტო კომპიუტერის საკონტროლო-სამოწმებელი აპარატურა არ გამოიყენება. საბორტო კომპიუტერის მუშაობის კონტროლი ყველა მომზადების შემთხვევაში სრულდება ტესტური (საკონტროლო) პროგრამების დახმარებით, რომლებიც შენახულია საბორტო კომპიუტერის მუდმივ დამსხმებელ მოწყობილობაში. ტესტური პროგრამების შესრულების შედეგები, რომლებიც სრულდება ციკლურად განსაზღვრული სიხშირით, აისახება სასიგნალო ნათურების დახმარებით კომპიუტერის ბლოკებზე და კომპლექსის მართვის პულტებზე.

სამუშაოების თითოეული სახისთვის, რომელიც სრულდება ფრენების მომზადების პერიოდში, დგება ტექნოლოგიური რუკები საბორტო კომპიუტერის სისტემების ექსპლუატაციის ინსტრუქციის შესაბამისად. ტექნოლოგიურ რუკებში მოყვანილია სამუშაოების ნუსხა და თანმიმდევრობა, მითითებულია შესამოწმებელ რეჟიმებში ინდიკაციის ხასიათი, ასევე ინფორმაციის გადამწოდებისადმი წაყენებული მოთხოვნები და შემსრულებელი და საინდიკაციო მოწყობილობების რეაქცია.

საბორტო კომპიუტერის მუშაობის უნარიანობის შემოწმების შემდეგ სრულდება კომპლექსის საბორტო სისტემების შესაბამისი შემოწმებები რომლებთანაც ურთიერთქმედებს საბორტო კომპიუტერი.

რეგლამენტური სამუშაოები სრულდება იმისათვის რომ შენარჩუნებულ იქნეს საბორტო კომპიუტერის კვანძების და მოწყობილობების მახასიათებლები დასაშვებ ფარგლებში და უზრუნველყოფილ იქნეს საბორტო კომპიუტერის საიმედოობა და უმტყუნო მუშაობა ექსპლუატაციის რეგლამენტურობის პერიოდში.

რეგლამენტური სამუშაოები სრულდება ტექნოლოგიური რუკების შესაბამისად. ამასთან შესაძლებელია საბორტო კომპიუტერი მოიხსნას საჰაერო ხომალდიდან და შემოწმდეს ის სპეციალურ სტენდებზე საკონტროლო - სამოწმებელ აპარატურასთან მიერთების შედეგად.

რეგლამენტური შესრულების პროცესში საბორტო კომპიუტერის კვანძები მოწმდება როგორც პროგრამულად, ისე სქემური ხერხებით საკონტროლო - სამოწმებელი აპარატურის დახმარებით ყველა რეჟიმში. ამასთან მოწმდება ყველა ბრძანების შესრულების სისწორე, დამსხმებელი მოწყობილობების მუშაობა საკონტროლო კოდების ჩაწერის და ამოკითხვის რეჟიმებში, შემტან - გამომტან მოწყობილობებში

ინფორმაციის გარდაქმნის სიზუსტე. გარდა ამისა, ფუნქციონირებისას სრულდება საბორტო კომპიუტერის კვანძების და ელემენტების პროფილაქტიკური შეცვლა, ელექტრული გასართების გაწმენდა, მეტალიზაციის რემონტი, ლაქსაფარის აღდგენა. ყველა ამ სამუშაოების შესრულების შემდეგ საბორტო კომპიუტერი ყენდება საჰაერო ხომალდზე.

რეგლამენტური სამუშაოების შესრულების შემდეგ ეტპს წარმოადგენს საბორტო სისტემების საბორტო კომპიუტერთან და ერთმანეთთან შეპირაპირება, ხოლო ამის შემდეგ საკონტროლო ამოცანების შესრულება.

საბორტო კომპიუტერის შეპირაპირება საბორტო სისტემებთან სრულდება თანმიმდევრობით თითოეულ ბლოკთან. შესასრულებელი ოპერაციების თანმიმდევრობა მითითებულია შესაბამის ტექნოლოგიურ რუკებში.

საბორტო კომპიუტერის შეპირაპირება საბორტო სისტემებთან ხორციელდება სპეციალური პროგრამის დახმარებით, რომელიც შეპირაპირების ტესტად იწოდება. საბორტო კომპიუტერის გადაყვანა შეპირაპირების ტესტის პროგრამის შესრულებაზე სრულდება კომპიუტერის რეჟიმების პულტიდან. ამავე პულტზე მითითებულია სისტემის მისამართი (ტიპი), რომელთანაც წარმოებს შეპირაპირება.

თუ შეპირაპირება ხდება გადამწოდთან, მაშინ მასზე (გადამწოდზე) ყენდება პარამეტრის მოთხოვნილი მნიშვნელობა, მაგალითად, საჰაერო ხომალდის ფრენის სიმაღლის მნიშვნელობა. საბორტო კომპიუტერი შეპირაპირების ტესტის შესრულების პროცესში პროგრამული ხერხით გამოიკითხავს ინფორმაციას, გაცემულს გადამწოდის მიერ და მისი გარდაქმნის შემდეგ შეტანა - გამოტანის მართვის მოწყობილობაში, ხდება მისი ჩაწერა ოპერატიულ დამხსომებელ მოწყობილობის შესაბამის უჯრედში. ოპერატიულ დამხსომებელ მოწყობილობაში მიღებული ინფორმაცია გაიცემა შემდეგ მართვის პულტების სპეციალურ საკონტროლო ინდიკატორზე, სადაც აისახება ათობითი ციფრების სახით. თუ გადამწოდზე დაყენებული საკონტროლო მნიშვნელობა ემთხვევა მოცემული დაშვების მნიშვნელობების საზღვრებს ასახულს პულტზე, მაშინ ეს იმის მაჩვენებელია რომ შეპირაპირება მოცემულ წერტილში შესრულებულია. ამის შემდეგ შეპირაპირება ხდება სხვა წერტილებში მოცემული გადამწოდისთვის და ანალოგიური სახით ხორციელდება ის თანმიმდევრობით სხვა გადამწოდების შემთხვევაშიც. ინფორმაციის გავლა გადამწოდიდან საბორტო კომპიუტერის ოპერატიულ დამხსომებელ

მოწყობილობაში და შემდეგ ინდიკაციის პულტებისკენ შესაძლებელია გაკონტროლდეს საბორტო კომპიუტერის საკონტროლო - სამოწმებელი აპარატურის დახმარებით.

თუ საბორტო კომპიუტერის შეპირაპირება ხდება საბორტო სისტემის ინდიკატორულ ან შემსრულებელ მოწყობილობებთან, მაშინ კომპიუტერში პროგრამული გზით ფორმირდება კოდების აუცილებელი საკონტროლო მნიშვნელობები, რომლებიც შემტან - გამომტანი მოწყობილობის სტანდარტული არხების დახმარებით გადაიგზავნება ბლოკში, რომლითაც ხორციელდება შეპირაპირება. შესაბამისად შესაბამის ინდიკატოსზე კონტროლდება მოცემული ინფორმაციის მნიშვნელობა.

ბიტური სიგნალების გავლა შეპირაპირებისას კონტროლდება ინდიკატორის სასიგნალო ნათურების დახმარებით ან სამოწმებელი სისტემების პულტებით.

ყველა საბორტო სისტემის ერთობლივი მუშაობის შემოწმება, რომლებიც ურთიერთქმედებენ საბორტო კომპიუტერთან, საკონტროლო ამოცანების გადაწყვეტის გზით სრულდება სარეგლამენტო სამუშაოების დამამთავრებელ ეტაპზე. საკონტროლო ამოცანის გადაწყვეტა სრულდება იმ პროგრამის დახმარებით, რომელიც შენახულია საბორტო კომპიუტერის მუდმივ დამხსომებელ მოწყობილობაში. ამასთან საკონტროლო ამოცანის ტიპი მოიცემა კომპლექსის რეჟიმების მართვის პულტზე.

გადასაწყვეტი ამოცანებისთვის საწყისი მონაცემები მოიცემა ფრენის დროს გადამწოდების მუშაობის იმიტაციის გზით. ამასთან საწყისი მონაცემების მნიშვნელობები (სიჩქარე, ფრენის სიმაღლე და ა.შ) მოიცემა შესაბამის ტექნოლოგიურ რუკებში. თითოეული საკონტროლო ამოცანა შეესაბამება სისტემის მუშაობის განსაზღვრულ რეჟიმს.

საკონტროლო ამოცანების გადაწყვეტის შედეგები მოწმდება საჰაერო ხომალდის საშტატო ინდიკატორების და შემსრულებელი მოწყობილობების მიერ.

თანამედროვე საბორტო კომპლექსების სირთულე მოითხოვს მთელი რიგი ღონისძიებების გატარებას, იმისათვის რომ მიღწეულ იქნეს მათი საიმედო მუშაობა ექსპლუატაციის ყველა ეტაპზე. ამ მხრივ კომპლექსების საექსპლუატაციო საიმედოობის ამაღლების გზაზე ერთ-ერთ ძირითადს წარმოადგენს კონტროლის ეფექტური მეთოდების გამოყენება.

კონტროლის ფუნქცია შეიძლება დაეკისროს ავტონომიურ გარე სისტემებს, რომლებიც მიერთებულია კომპლექსის კონტროლის დროს და ჩაშენებულ სისტემებს,

ანუ ისეთ სისტემებს, რომლებიც შედის უშუალოდ კომპლექსის შემადგენლობაში. კონტროლის ავტონომიური გარე სისტემის გამოყენება მოითხოვს საკონტროლო ობიექტების დამატებით დამუშავებას, ხოლო მათი ტრანსპორტირება და საბორტო მოწყობილობებთან მიერთება მოითხოვს დამატებით დროს. კონტროლის ჩაშენებული სისტემები დაფუძნებულია კომპლექსის საბორტო მოწყობილობების გამოყენებაზე, ძირითადად საბორტო კომპიუტერის და მისი შეტანა - გამომტანის მოწყობილობების, მართვის სშტატო პულტების და ინდიკატორების გამოყენებაზე. კონტროლის ჩაშენებული სისტემების გამოყენება უზრუნველყოს ექსპლუატაციის ყველა ეტაპზე შემოწმებების შესრულებას, კონტროლის შედეგების დაგროვებას ექსპლუატაციის პროცესში კომპლექსის მუშაობის პროგნოზირებისთვის.

კონტროლის ჩაშენებული სისტემა შესაძლებელია ბაზირდებოდეს კომპლექსის საბორტო კომპიუტერზე ან შეიძლება გააჩნდეს თავისი საბორტო კომპიუტერი, რომელიც ინფორმაციას ცვლის ძირითად საბორტო კომპიუტერთან. მეორე შემთხვევაში ძირითადი საბორტო კომპიუტერი თავისუფლდება კონტროლის პროგრამების შესრულებისგან, თუმცა რა თქმა უნდა ასეთ დროს კომპლექსის საერთო სტრუქტურა რთულდება.

საკონტროლო ქვესისტემებს წარმოადგენენ გადამწოდები, შემსრულებელი და ინდიკატორული მოწყობილობები. ამ ქვესისტემებს შეიძლება გააჩნდეს კონტროლის საკუთარი სქემები, რომლებიც აფორმირებენ საკონტროლო ინფორმაციას. ეს ინფორმაცია, ასევე ინფორმაცია გაცემული ქვესისტემების მიერ, გამოიყენება მთელი სისტემის კონტროლის დროს. შემთხვევითი მოწყობილობები უზრუნველყოფენ ქვესისტემების კავშირს საბორტო კომპიუტერის შემტან - გამომტან მოწყობილობასთან.

მასტიმულირებელი სიგნალების ფორმირების სქემა სიგნალებს გამოიმუშავებს მოცემული პარამეტრებით, რომლებიც საბორტო კომპიუტერის ბრძანებების შესაბამისად მიეწოდება საკონტროლებელ ქვესისტემებში. ამასთან ქვესისტემებში კონტროლის ჩაშენებული სქემები გადადის კონტროლის რეჟიმში და გასცემს სიგნალებს, რომლებიც შემტან - გამომტანი მოწყობილობის საშუალებით შეიტანება საბორტო კომპიუტერში ანალიზისთვის. საბორტო კომპიუტერის კონტროლის შედეგების ინდიკაცია ხდება მართვის მოწყობილობებზე.

საკონტროლო - ჩამწერი აპარატურა (სჩა) გამოიყენება იმ ინფორმაციის ჩაწერისათვის, რომელიც მუშავდება საბორტო კომპიუტერში ძირითადად ფრენის დროს, რაც საშუალებას იძლევა მოხდეს მისი შემდგომი ანალიზი მიწისზედა პირობებში.

კომპლექსის საბორტო სისტემების კონტროლი, რომელიც ასევე იწოდება ფუნქციურ კონტროლად, სრულდება სპეციალური პროგრამებით, რომლებიც შენახულია საბორტო კომპიუტერის მუდმივ დამხსომებელ მოწყობილობაში.

საბორტო კომპიუტერი ფუნქციური კონტროლის რეჟიმში ჩართვისას გადაირთვება შესაბამისი პროგრამის პერიოდულ შესრულებაზე. ამ პროგრამით ხდება სიგნალების ფორმირება, რომლებიც შეტანა - გამოტანის მოწყობილობის საშუალებით გაიცემა შეამოწმებელ ქვესისტემებში და გადაჰყავს ისინი კონტროლის რეჟიმში. გარკვეული დროის შემდეგ შემოწმებადი ქვესისტემები იწყებენ საკონტროლო პარამეტრების გაცემას, რომლებიც საბორტო კომპიუტერის პროგრამით შეიტანება კომპიუტერში და ედრება პარამეტრის ეტალონურ ან გამოთვლილ მნიშვნელობებს. თუ შემოწმებადი პარამეტრების გადახრა არ აჭარბებს დადგენილ დამზვებას, რომელიც ინახება კომპიუტერის მუდმივ დამხსომებელ მოწყობილობაში, მაშინ კონტროლირებადი ქვესისტემის მუშაობა მოცემული არხის მიხედვით ითვლება სწორად. ანალოგიური ხერხით მოწმდება ყველა სხვა არხებიც.

ქვესისტემების შემოწმების შედეგების მიხედვით საბორტო კომპიუტერში ფორმირდება ჯამური ინფორმაცია, რომელიც აისახება სისტემის შესაბამის ინდიკატორებზე. თუ ყველა ქვესისტემა აღმოჩნდება გამართული, მაშინ საბორტო კომპიუტერში ფორმირდება საერთო სიგნალი სისტემის გამართულობის შესახებ.

საბორტო სისტემების გამართულობის შესახებ ინფორმაციის მიღების შემდეგ საბორტო კომპიუტერის დახმარებით შესაძლებელია მოხდეს ძირითადი რეჟიმების შესაბამისი ამოცანების შესრულება. მაგალითად, დროის დაჩქარებულ მასშტაბში შესაძლებელია გადაწყდეს მარშრუტის მიხედვით ფრენის ამოცანა. ამასთან ერთად კონტროლდება სანავიგაციო ინფორმაციის გაცემის სისწორე.

ფრენის დროს სისტემის ავტომატიზებული კონტროლისთვის დამახასიათებელია თავისი განსაკუთრებულობები, რომელიც მდგომარეობს იმაში, რომ კონტროლირებადი ქვესისტემები ასრულებენ თავიანთ ძირითად ფუნქციებს, რაც ყოველთვის არ იძლევა საშუალებას გადაყვანილ იქნეს ისინი საკონტროლო რეჟიმში.

ფრენის დროს სისტემის მუშაბისუნარიანობის კონტროლი შეიძლება შესრულდეს მისი მუშაობის შეფასების გზით, რომელიც შემოდის მუშა არხებით.

ამრიგად, კონტროლმა ფრენის დროს არ უნდა გამოიწვიოს მოცემულ მომენტში ძირითადი ამოცანის შეწყვეტა. აღნიშნულმა განსაკუთრებულობებმა კონტროლის ჩატარების თვალსაზრისით მოითხოვა სპეციალური მეთოდები, რომელთა შორის ყველაზე უფრო ხშირად გამოიყენება:

- ინფორმაციული სიჭარბე;
- სისტემის დუბლირება;
- პარამეტრების შედარება ირიბი მონაცემების მიხედვით;
- მონაცემების ანალიზი, რომლებიც გაცემულია ცალკეული ქვესისტემების ჩაშენებული კონტროლით;
- მუშა სიგნალების მიმდინარე მნიშვნელობების დაშვებითი შეფასება.

ჩამოთვლილი მეთოდებიდან პირველი დაფუძნებულია პარამეტრის შესახებ ინფორმაციის ანალიზზე, რომელიც მიღებულია სხვადასხვა ქვესისტემებისგან. მაგალითად, ინფორმაცია ფრენის სიჩქარის შესახებ შესაძლებელია მიღებულ იქნეს საჰაერო სიგნალების სისტემისგან (სსს), სიჩქარის და გადაცდენის კუთხის დოპლერული გაზომვიდან, ინერციალური სისტემიდან. ასეთნაირად მიღებული ინფორმაციის დამუშავება და მის ანალიზი საშუალებას იძლევა განისაზღვროს გაუმართავი არხი.

ქვესისტემების დუბლირების მეთოდი ითვალისწინებს სისტემის ორი, სამ და უფრო მეტი რაოდენობის ერთნაირი ბლოკების გამოყენებას. ეს მეთოდი საკმარისად ეფექტურია, თუმცა ამ შემთხვევაში არსებითად იზრდება საბორტო სისტემების მასა და გაბარიტები.

პარამეტრების შედარების მეთოდი ირიბი მონაცემების მიხედვით დამოკიდებულია ინფორმაციის ურთიერთდამოკიდებულებაზე, რომელიც გაიცემა სხვადასხვა ქვესისტემებით, მაგალითად, სიჩქარესა და აჩქარებას შორის, დაგვერდებასა და საჰაერო ხომალდის კუთხურ სიჩქარეს შორის. კონტროლის ალგორითმებში რომლებიც პერიოდულად სრულდებიან საბორტო კომპიუტერებით, წარმოებს ერთი მონაცემების მეორეში გარდაქმნა და მათი შეპირისპირება.

ინფორმაციის ანალიზზე დაფუძნებული მეთოდი, რომელიც გაიცემა ქვესისტემების კონტროლის სქემების მიხედვით, ითვალისწინებს სისტემების მტყუნებების სიგნალების

არსებობის პერიოდულ შემოწმებას. მტყუნების სიგნალის გაჩენისას კონკრეტული ქვესისტემის შემთხვევაში შეძლებისდაგვარად ორგანიზდება, მტყუნებული ქვესისტემის ალგორითმული შემოვლა. მაგალითად, თუ მოხდება რადიოსიმაღლის მზომის მხრიდან მტყუნების დაფიქსირება, მაშინ ფრენის სიმაღლის შესახებ ინფორმაცია შემოტანილი იქნება გეომეტრიული სიმაღლის გადამწოდით.

პარამეტრების მიმდინარე მნიშვნელობების დაშვებითი შეფასების მეთოდი დაფუძნებულია დროის გარკვეულ მონაკვეთში შესასვლელი მონაცემების ცვლილების ანალიზზე. მაგალითად, ნავიგაციის ამოცანის გადაწყვეტის ერთი ციკლიდან მომდევნომდე ინფორმაცია საჰაერო ხომალდის ფრენის შესახებ თუ შეიცვალა სიდიდით, რომელიც ამეტებს საჰაერო ხომალდის რეალურ (დასაშვებ) მნიშვნელობას, მაშინ ეს ფაქტი მიუთითებს იმაზე, რომ არასწორად მუშაობს სიჩქარის გადამწოდი ან არხი, რომლითაც ეს ინფორმაცია მიეწოდება საბორტო კომპიუტერში.

11.2. საბორტო კომპიუტერების მუშაობის უნარიანობის კონტროლის მეთოდები

საბორტო კომპიუტერის მუშაობის უნარიანობის კონტროლის მეთოდები მჭიდროდაა დაკავშირებული მის ექსპლუატაციასთან, როგორც მომსახურების ობიექტთან. აქედან გამომდინარე, თუ რამდენად რაციონალურად იქნება არჩეული კონტროლის მეთოდები დამოკიდებულია საბორტო კომპიუტერის მომსახურების ექსპლუატაციის ეფექტურობა.

საბორტო კომპიუტერის ექსპლუატაციის ზემოთ ჩამოთვლილი ძირითადი დებულებებიდან გამომდინარეობს, რომ მისი მაღალხარისხოვანი ექსპლუატაცია შეიძლება მიღწეულ იქნეს საბორტო კომპიუტერის ცალკეული მოწყობილობების და მთლიანად საბორტო კომპიუტერის ფუნქციონირების სისწორის კარგი კონტროლის ორგანიზებისას. კონტროლის მეთოდების არჩევისას ძირითადი ყურადღება უნდა მიექცეს კონტროლის მეთოდის უნარს შეცდომის აღმოჩენის თვალსაზრისით, ასევე აპარატურის მოცულობას და დროს, რომელიც იხარჯება კონტროლზე.

საბორტო კომპიუტერში გამოყენებული კონტროლის მეთოდები შეიძლება დაყოფილ იქნეს პროგრამულ და სქემურ (აპარატულ) მეთოდებად.

11.3. საბორტო კომპიუტერის მუშაობისუნარიანობის კონტროლის

პროგრამული მეთოდები

გამოთვლების სისწორის კონტროლის პროგრამული მეთოდები განკუთვნილია საბორტო კომპიუტერის სხვადასხვა რეჟიმებში მუშაობისას გამოთვლითი პროცესების რეალიზაციის სისწორის და მისი ყველა მოწყობილობის სწორი ფუნქციონირების შემოწმებისათვის.

კონტროლის პროგრამული მეთოდები დაფუძნებულია საბორტო კომპიუტერის პროგრამულ უზრუნველყოფაში სპეციალური მაკონტროლებელი პროგრამების და დამატებითი თანაფარდობების ჩართვით საერთო ალგორითმში. პროგრამული მეთოდები იყოფა პროგრამულ – ლოგიკურ და ტესტური კონტროლის მეთოდებად.

პროგრამულ – ლოგიკური კონტროლი დაფუძნებულია საინფორმაციო სიჭარბის გამოყენებაზე და განკუთვნილია გამოთვლითი პროცესის სწორი რეალიზების შემოწმებისათვის. საინფორმაციო სიჭარბის უზრუნველყოფა ხორციელდება საერთო ალგორითმში დამატებითი თანაფარდობების შეტანით, რაც იძლევა გამოთვლების დროს შეცდომების აღმოჩენის და აღმოფხვრის საშუალებას.

ყველაზე უფრო ფართოდ გამოიყენება პროგრამულ – ლოგიკური კონტროლის შემდეგი ხერხები: ორმაგი გათვლის ხერხი შედეგების შედარებით; საკონტროლო თანაფარდობათა ხერხი; „წაკვეთილი“ ალგორითმის ხერხი; შედეგების გადაწყვეტის ლოგიკური ანალიზის ხერხი.

განვიხილოთ ისინი ჩამოთვლილი თანმიმდევრობით.

ორმაგი გათვლის ხერხი შედეგების შედარებით. ეს ხერხი უფრო გავრცელებულია. მისი ძირითადი ღირსება რეალიზაციის სიმარტივეშია. ამ ხერხის გამოყენებისას მთლიანი სამუშაო პროგრამა იყოფა ცალკეულ ნაწილებად და გამოთვლის რაღაც ეტაპის დასრულების შემდეგ ხორციელდება საკონტროლო ეტაპის ყველა ბრძანების შედეგების საკონტროლო აჯამვა. ხდება საკონტროლო ჯამის დამახსოვრება და სრულდება ამ ეტაპის განმეორებით გამოთვლა შემდგომი საკონტროლო აჯამვის შესრულებით. ხორციელდება ორივე საკონტროლო ჯამების შედარება და მათი დამთხვევის შემთხვევაში სრულდება გამოთვლის შემდეგი ეტაპი. როცა არ ხდება საკონტროლო ჯამების დამთხვევა, წარმოებს ამ ეტაპის მესამე გათვლა. მესამე საკონტროლო ჯამის წინა ორი საკონტროლო ჯამებიდან ერთ – ერთზე დამთხვევისას სრულდება გამოთვლის მომდევნო ეტაპი. წინააღმდეგ

შემთხვევაში ეკიპაჟს მიეწოდება სიგნალი საბორტო კომპიუტერის შეფერხების შესახებ. ამ მეთოდის უარყოფით მხარეს წარმოადგენს ალგორითმის შესრულების დიდი დრო და ამასთანავე ის, რომ, მას შეუძლია აღმოაჩინოს და აღმოფხვრას მხოლოდ შემთხვევითი შეცდომები, რომლებიც გამოწვეულია შეფერხებებით კომპიუტერში.

ორმაგი გათვლის კონტროლის მეთოდის გამოყენებისას, საჭიროა სწორად ჩატარდეს ალგორითმის დაყოფა საკონტროლო ეტაპებად. კვლევამ აჩვენა, რომ არსებობს ალგორითმის საკონტროლო უბნების ოპტიმალური რაოდენობა რომლის დროსაც კონტროლზე დახარჯული დრო მინიმალურია.

საბორტო კომპიუტერის ექსპლუატაციის გამოცდილებამ აჩვენა, რომ შემთხვევითი შეფერხების ნაკადი შეიძლება ჩაითვალოს მარტივად. აღვნიშნოთ შემთხვევითი შეფერხების ნაკადის ინტენსიურობა λ -თი, ამოცანის ამოხსნის დრო- $T_{\text{ა}}$, საკონტროლებელი უბნების რაოდენობა N -ით, მაშინ ერთი საკონტროლებელი უბნის ალგორითმის რეალიზების დრო ტოლია

$$t = T_{\text{ა}} / N.$$

საჭიროა გავითვალისწინოთ, რომ ორმაგი გათვლის ხერხით კონტროლის ორგანიზებისას ამოცანის თითოეულ კონტროლირებად უბანზე იხარჯება დამატებითი დრო t , საკონტროლო აჯამების, საკონტროლო ჯამების შედარების, მესამე გათვლაზე გადაწყვეტილების მიღების ან გამოთვლების გაგრძელების შესახებ. აქედან გამომდინარე ალგორითმის შესრულების დრო $T_{\text{ა}}$ ტოლია

$$T_{\text{ა}} = 2N(t + t_{\text{ფ}}) + 2N(t + t_{\text{ფ}}) \cdot \lambda \cdot (t + t_{\text{ფ}}) = 2N(t + t_{\text{ფ}}) + 2(t + t_{\text{ფ}})^2 \cdot \lambda N$$

მიღებულ გამოსახულებაში t -ს მნიშვნელობის ჩასმით მიიღება

$$T_{\text{ა}} = 2N(T_{\text{ა}}/N + t_{\text{ფ}}) + 2(T_{\text{ა}}/N + t_{\text{ფ}})^2 \cdot \lambda N.$$

N ალგორითმის რეალიზების მინიმალური დროის პირობიდან არჩევა, N -ით $T_{\text{ა}}$ -ს დიფერენცირება და წარმოებულის ნულის მნიშვნელობის მინიჭება იძლევა:

$$dT_s/dN = t_{\text{დ}} - (T_s^2 \lambda) / N^2 + t_{\text{დ}}^2 \lambda = 0,$$

აქედან

$$N_{\text{ოპტ}} = T_s \sqrt{\lambda / [t_{\text{დ}} (1 + \lambda t_{\text{დ}})]}.$$

შეფერხებების ექსპონენციალური კანონით განაწილებისას და პროგრამის უბნის ოპტიმალური სიგრძის შემთხვევაში შეუფერხებელი მუშაობის ალბათობა ერთ უბანზე იქნება:

$$P(t) = e^{-\lambda t} = e^{-\lambda (T_s / N_{\text{ოპტ}})} = e^{-\lambda \sqrt{[t_{\text{დ}} (1 + \lambda t_{\text{დ}})] / \lambda}}.$$

ცდების გამეორების შესახებ თეორემის საფუძველზე შესაძლებელია განისაზღვროს ალგორითმის სწორი რეალიზაციის ალბათობა ორი-სამმაგი გათვლის არსებობისას. ის ტოლია

$$P(T_A) = \{P(t)P(t) + [1 - P(t)]P(t)P(t) + P(t)[1 - P(t)]P(t)\}^N = [e^{-2\lambda t} + 2e^{-2\lambda t}(1 - e^{-\lambda t})]^N.$$

საკონტროლო თანაფარდობების ხერხი. ის საშუალებას იძლევა განისაზღვროს არა მარტო ის შეცდომები, რომლებიც ჩნდება შემთხვევითი შეფერხებების შედეგად, არამედ ის შეცდომებიც, რომლებიც წარმოიშობა მტყუნებების შედეგადაც. ამ ხერხის არსი მდგომარეობს იმაში, რომ საერთო ალგორითმში ჩართულია სხვადასხვა მათემატიკური თანაფარდობები, რომლებიც საშუალებას იძლევა შემოწმდეს ძირითადი ამოცანის შესრულების სისწორე მათი რეალიზაციის გზით.

ასეთ მათემატიკურ თანაფარდობებს შეიძლება წარმოადგენდეს შემდეგი სახის დამოკიდებულება $\sum_{i=1}^n P_i = 1$ ან $\sin^2 \alpha + \cos^2 \alpha = 1$, რაც იმას ნიშნავს, რომ მოვლენათა სრულ ჯგუფში ყველა შესაძლო მნიშვნელობათა ალბათობების ჯამი ტოლია 1-ის.

საკონტროლო თანაფარდობების მოცემული სიზუსტით შესრულებისას ხორციელდება ძირითადი ამოცანის შემდგომი შესრულება. საკონტროლო თანაფარდობების არ შესრულებისას ამოცანის მოცემული უბანი ან მეორდება, ან საბოლოო კომპიუტერი

ჩერდება, რის შესახებაც ატყობინებენ ეკიპაჟს. აღნიშნულმა ხერხმა შეიძლება მოგვცეს საკმაოდ მნიშვნელოვანი მოგება დროის მიხედვით ორმაგი გათვლის ხერხთან შედარებით.

„წაკვეთილი“ ალგორითმის ხერხი. აღნიშნული ხერხი საშუალებას იძლევა აღმოაჩინოს და აღმოფხვრას შეცდომები გამოწვეული შემთხვევითი შეფერხებებისა და სისტემატიური მტყუნებების შედეგად. „წაკვეთილი“ ალგორითმი მნიშვნელოვნად განსხვავდება ძირითადისგან. ეს ხერხი გულისხმობს ძირითადი ამოცანის გამარტივებული ალგორითმის არსებობას. გამარტივებული ალგორითმი უნდა იყოს ძირითადზე უფრო მცირე. „წაკვეთილი“ ალგორითმის ძირითად ალგორითმთან ერთად რეალიზების შემთხვევაში მათი შედეგების მოცემული სიზუსტით ფარგლებში დამთხვევა მიუთითებს გამოთვლითი პროცესის სწორად მიმდინარეობაზე.

ამოცანის გადაწყვეტის შედეგების ლოგიკური ანალიზის ხერხი. ეს ხერხი იძლევა, როგორც შემთხვევითი, ასევე სისტემატური შეცდომების გამოვლენის საშუალებას. მისი არსი მდგომარეობს ამოცანის გადაწყვეტის მიმდინარეობისას გამოთვლითი რაღაც პარამეტრების და მათი ნაზრდების შედარებაში თავიდანვე ცნობილ მათი ცვლილების საზღვრებთან. ეს ხერხი გამოიყენება იმ შემთხვევაში თუ წინასწარ ცნობილია ზოგიერთი პარამეტრების ცვლილების კანონები.

ჩასმის ხერხი. ეს ხერხი მდგომარეობს იმაში, რომ მთელი რიგი სამიზნო სიდიდეების მიღების შემდეგ წყდება შებრუნებული ამოცანა. საწყისი მონაცემების სახით აიღება უცნობების ნაპოვნი მნიშვნელობები და განისაზღვრება რაღაც სიდიდეები, რომლებიც პირდაპირი გამოთვლებისას გამოიყენებოდნენ როგორც საწყისი მონაცემები. შემდგომ ხდება ამ სიდიდეების შედარება პირდაპირი გამოთვლების საწყის მონაცემებთან. დამთხვევა მოცემული სიზუსტის ფარგლებში ასაბუთებს გამოთვლებში შეცდომების არ არსებობას. მაგალითად, ალგებრული განტოლებების სისტემების ამოხსნისას საწყისი მონაცემების სახით შეიძლება აღებულ იქნეს გამოთვლებისას მიღებული ფესვები $X_1, X_2, \dots, X_n$; უცნობების სახით კი – რაღაც კოეფიციენტები, რომლებიც განსაზღვრული არიან ამოცანის შებრუნებული ამოხსნისას. მიღებული კოეფიციენტების შედარება არსებულ კოეფიციენტებთან მიუთითებს იმაზე, რომ სისტემა ამოხსნილია სწორად. ანალოგიური ხერხით შეიძლება ვისარგებლოთ დიფერენციალური განტოლებების ამოხსნის შემთხვევაშიც.

საბორტო კომპიუტერის ალგორითმის დამუშავებისას საჭიროა გამოყენებულ იქნეს გამოთვლების სისწორის შემოწმების ისეთი ეფექტური ხერხები, რომელთა გამოყენება შესაძლებელს გახდის მნიშვნელოვნად ამაღლდეს ინფორმაციის დამუშავების საიმედოობა საჰაერო ხომალდის ბორტზე მანქანური დროის მინიმალური დანახარჯებით.

გამოთვლების კონტროლის ყველა განხილული ხერხი ეფექტური იქნება მხოლოდ მაშინ, თუ შესრულდება საბორტო კომპიუტერის პროგრამების რეალიზაციის მსვლელობის სისწორის პირობა.

თითოეული პროგრამა, რომელიც ასრულებს საბორტო კომპიუტერის მუშაობის ამა თუ იმ რეჟიმს და მთლიანად სისტემის რეალიზებას, აიგება ცალკეული ქვეპროგრამების სახით პროგრამა დისპეტჩერის მეშვეობით. პროგრამის შესრულების მსვლელობის სისწორე განისაზღვრება: ქვეპროგრამების მუშაობის ჩართვის თანმიმდევრობის კონტროლით; ქვეპროგრამების მუშაობის ხანგრძლივობის კონტროლით; პროგრამის გადასვლების და წყვეტების შესრულების კონტროლით.

ქვეპროგრამების ჩართვის თანმიმდევრობის კონტროლი. საბორტო კომპიუტერის და სისტემის მუშაობის რეჟიმის უზრუნველყოფა ხორციელდება სათანადო პროგრამით, შედგენილი ცალკეული ქვეპროგრამებისაგან S_i , რომლებიც სრულდებიან რიგრიგობით. კონტროლის მეთოდი მდგომარეობს შემდეგში: ქვეპროგრამა S_i t_i დროის განმავლობაში არანაკლებ ერთხელ, ირთვება მუშაობაში და მისი თითოეული შესრულების შედეგად იცვლება რაღაც პარამეტრი $R_i(t)$, რომელიც ინახება კომპიუტერის ოპერატიულ მეხსიერებაში. გამოსახულების შემოწმებით შეიძლება დადგინდეს, რომ ქვეპროგრამა S_i სრულდებოდა. საბორტო კომპიუტერის მუშაობის თითოეული შესასრულებელი რეჟიმისთვის წინასწარ ცნობილია ქვეპროგრამებს შორის კავშირები. ქვეპროგრამებს შორის კავშირების დეტერმინირება გამოიყენება მათი შესრულების თანმიმდევრობის კონტროლისათვის. ამ შემთხვევაში თითოეული ქვეპროგრამა შესრულებისას მეხსიერების განსაზღვრულ უჯრედში აფიქსირებს ამ ქვეპროგრამაზე წინასწარ მიმაგრებულ პირობით კოდს. შემდეგი ქვეპროგრამის შესრულება იწყება მხოლოდ წინა ქვეპროგრამის ჩაწერილი პირობითი კოდის ანალიზის შემდეგ.

ქვეპროგრამების მუშაობის ხანგრძლივობის კონტროლი. ის გულისხმობს პროგრამის ცალკეული უბნების მუშაობის ხანგრძლივობის კონტროლს, უზრუნველ-ყოფს პროგრამის რეალიზების დარღვევების აღმოჩენას დაციკვლისა და სხვა მიზეზების გამო, რაც იწვევს პროგრამის ცალკეული ქვეპროგრამების შესრულების დროის გაზრდას. ამ მიზნით გამოიყენება ფარდობითი დროის მთვლელები, რომელთა დანიშნულებაა დროითი ინტერვალების დათვლა. ამ შემთხვევაში გარკვეული შეზღუდვები ედება S_i ქვეპროგრამის შესრულების T_i ხანგრძლივობას. დასაწყისში ფარდობითი დროის მრიცხველში სპეციალური ბრძანებით შეიტანება კოდი, რომელიც შესაბამისია მოცემული ქვეპროგრამის შესრულების დასაშვები ფარდობითი ხანგრძლივობისა. S_i ქვეპროგრამის თითოეული რეალიზაციის შემთხვევაში ფარდობითი დროის მრიცხველის შიგთავსიდან ხდება ერთიანის გამოკლება. ფარდობითი დროის მრიცხველში 0-ის არსებობისას გამომუშავდება შეცდომის სიგნალი, რომელიც იწვევს პროგრამის წყვეტას და შეცდომების ანალიზის ქვეპროგრამაზე გადასვლას. წინააღმდეგ შემთხვევაში ხდება შემდეგი ქვეპროგრამის შესრულებაზე გადასვლა.

პროგრამის წყვეტის და გადასვლების შესრულების სისწორის კონტროლი. ამ სახის კონტროლი ხორციელდება მართვის გადაცემის პირობითი და უპირობო ბრძანებების საშუალებით და დაფუძნებულია შეცდომის სიგნალის პროგრამულ ბლოკირებაზე. ამ კონტროლის იდეა მდგომარეობს იმაში, რომ გადასვლის წინ სპეციალური ბრძანებით მისამართის მოდიფიკაციისა და გადასვლის შესრულების წრედში მიეწოდება შეცდომის სიგნალი, რომლის მიწოდება შეიძლება დაბლოკილ იქნეს ბლოკირების ბრძანებების დახმარებით, რომლებიც განლაგებულია პროგრამის ყველა იმ ადგილებში, სადაც მიმდინარეობს გადასვლა.

გადასვლის არასწორად შესრულებისას შეცდომის სიგნალი არ იქნება ბლოკირებული და გამოიწვევს წყვეტას და შეფერხებების ანალიზის ქვეპროგრამის მუშაობაში ჩართვას. აღნიშნული ხერხი უზრუნველყოფს, როგორც შემთხვევითი, ასევე სისტემატური შეცდომების აღმოჩენის დიდ ალბათობას, მაგრამ ითხოვს დამატებით ხელსაწყოებს და პროგრამაში სპეციალური ბრძანებების შეტანას.

პროგრამის წყვეტა წარმოადგენს საბორტო კომპიუტერის მუშაობის დამახასიათებელ თავისებურებას, რომელიც ასრულებს თავის ალგორითმებს დროის

რეალურ მასშტაბში. საბორტო კომპიუტერის მუშაობის პროცესში, სისტემის მუშაობის რეჟიმიდან და შემოსული და გაცემული ინფორმაციის მოცულობიდან გამომდინარე, წარმოებს პროგრამის მრავლობითი წყვეტა, რომელიც ითვლის ასობით წყვეტას წამში.

პროგრამის წყვეტის შესრულების სისწორის კონტროლისთვის ხშირად გამოიყენება შეცდომის სიგნალის პროგრამული ბლოკირების განხილული მეთოდი. განსხვავება მდგომარეობს მხოლოდ იმაში, რომ შეცდომის სიგნალის ფორმირება ხდება სქემური გზით, ხოლო წყვეტა შესაძლებელია იყოს პროგრამის ნებისმიერ ადგილზე. შეცდომის სიგნალის ბლოკირება ხორციელდება ქვეპროგრამის ერთ-ერთი პირველი ბრძანებების რიგიდან. ძირითად პროგრამაზე გადასვლის კონტროლი წყვეტის შემდეგ ხორციელდება ძირითადი შედეგების რეგისტრების მდგომარეობის აღდგენის სისწორის და საბორტო კომპიუტერების მეხსიერების იმ უჯრედების შემოწმებით, რომლებიც გამოიყენებიან ქვეპროგრამის მიერ. ამ მიზნით გამოიყენება საბორტო კომპიუტერის მეხსიერებაში ცვლადი ინფორმაციის მრავალჯერადი შენახვის მეთოდი, ინფორმაციის შემდგომი შედარების განხორციელებით.

შეცდომის სიგნალი გამომუშავდება მაშინ, თუ მოხდება თუნდაც ერთი რეგისტრის შიგთავსის ან საბორტო კომპიუტერის მეხსიერების უჯრედის არ შედარება.

11. 4. საბორტო კომპიუტერების მუშაობის უნარიანობის ტესტური კონტროლი

პროგრამულ-ლოგიკური კონტროლის ძირითადი დანიშნულებაა გამოთვლითი პროცესების მიმდინარეობის დროს წარმოშობილი შეცდომებისგან დაიცვას ალგორითმები და მიზანდ არ ისახავს დაადგინოს უწესივრობების ფაქტის არსებობა საბორტო კომპიუტერების რაღაც მოწყობილობაში, ან აღმოაჩინოს და აღმოფხვრას არსებული უწესივრობა.

ტესტური კონტროლის დახმარებით რაღაც მოწყობილობაში ხდება უწესივრობების აღმოჩენა, ლოკალიზაცია და სიგნალიზაცია. ამჟამად ის წარმოადგენს შემოწმების ძირითად საშუალებას საბორტო კომპიუტერის ფუნქციონირების სისწორის დასადგენად მისი ექსპლუატაციის ყველა ეტაპზე. ტესტური კონტროლის ძირითადი დანიშნულებაა სისტემატიური შეცდომების აღმოჩენა, მათი მტყუნებების შედეგად.

საბორტო კომპიუტერის ექსპლუატაციის პროცესში გამოიყენება საბორტო კომპიუტერის კომპლექსური შემოწმების გამოსაცდელი ტესტები და დიაგნოსტიკური ტესტები.

კომპლექსური შემოწმების გამოსაცდელი ტესტები. ისინი გამოიყენებიან საბორტო კომპიუტერში უწყისვრობების აღმოჩენისათვის და საშუალებას იძლევა მოახდინოს საბორტო კომპიუტერის ყველა მოწყობილობის ერთობლივი მუშაობის კონტროლი. გამოსაცდელი ტესტის სახით ხშირად გამოიყენება, კომპიუტერის მიერ გადასაწყვეტი ამოცანების რიცხვიდან ერთი ან რამდენიმე ამოცანა. ამასთან ერთად, საწყის მონაცემებად იღებენ ისეთ მონაცემებს, რომელთა გადაწყვეტის (ამოხსნის) შედეგები წინასწარაა ცნობილი. ასეთ შემთხვევაში გამოსაცდელი ტესტები საშუალებას იძლევა შემოწმდეს არა მარტო კომპიუტერის მუშაობისუნარიანობა, არამედ სისტემის ფუნქციონირებაც მთლიანობაში. ასეთი ტიპის ტესტური პროგრამები არ მოითხოვენ მეხსიერების დამატებით ტევადობას.

დიაგნოსტიკური ტესტები გამოიყენება მანქანაში უწყისვრობის ადგილის განსაზღვრისათვის და უწყისვრობის არსებობის ფაქტის დადგენის შემდეგ საცდელი ტესტების საშუალებით საბორტო კომპიუტერის კომპლექსური შემოწმებისათვის. აღნიშნული ტესტები წარმოადგენს სპეციალურ პროგრამებს, რომელთა რეალიზაცია ხორციელდება მანქანის ბრძანებათა სისტემაში შემავალი ყველა ოპერაციის გამოყენებით. ტესტის სწორად შესრულებისას გამომუშავდება მზადყოფნის სიგნალი „გამართულია“, სხვა შემთხვევაში სიგნალი არ გამომუშავდება.

პროგრამები ინახება მუდმივ დამხსომებელ მოწყობილობაში (მეხსიერებაში) – ROM. აღნიშნული ტესტები შეიცავს მუდმივი მეხსიერების, ოპერატიული მეხსიერების RAM, არითმეტიკულ-ლოგიკური მოწყობილობის ALU, მართვის მოწყობილობის, შეტანა/გამოტანის (INPUT/OUTPUT) მოწყობილობების შემოწმების ტესტებს.

11.5. საბორტო კომპიუტერის სქემური (აპარატული) კონტროლი

სქემური კონტროლი უზრუნველყოფს საბორტო კომპიუტერის ან მისი ცალკეული მოწყობილობების კონტროლს, რომელიც ხორციელდება საბორტო კომპიუტერის შემადგენლობაში ჩართული დამატებითი მოწყობილობების საშუალებით.

სქემური კონტროლი შეიცავს: კონტროლს მაკორექტირებელი კოდების გამოყენებით და კონტროლს მოდულით.

11.5 .1. სიგნალების ჭარბი კოდირების ზოგადი ცნობები

ციფრული მოწყობილობების მუშაობა შეუძლებელია ინფორმაციის გადაცემის გარეშე ინფორმაციის გაცვლის სხვადასხვა არხებით. საინფორმაციო შეტყობინების გადაცემა ყოველთვის შეიძლება დაყვანილ იქნეს კოდების სახით წარმოდგენილი ორობითი რიცხვების გადაცემამდე.

შეზღუდული ხანგრძლივობის კოდების თვისებების კვლევით და მათი ტექნიკური რეალიზების ოპტიმალური სტრუქტურების მოძებნით დაკავებულია კოდირების თეორია. ეს თეორია ვითარდება ორი ძირითადი მიმართულებით: კოდების ძიება, რომლებიც დაბრკოლების არ მქონე არხებში უზრუნველყოფს წყაროს კოდური სიჭარბის მაქსიმალურ აღმოფხვრას და ამით ინფორმაციის გადაცემის ეფექტურობის ამაღლებას (ეკონომიური კოდირების თეორია); კოდების ძიება, რომლებიც დაბრკოლების მქონე არხებში აამაღლებენ ინფორმაციის გადაცემის უტყურობას (კოდირების დაბრკოლებამდგრადობის თეორია).

იმდენად რამდენადაც, საინფორმაციო არხისათვის დაბრკოლებას შეიძლება წარმოადგენდეს არა მარტო ის გარე ზემოქმედებები რომლებიც ხელს უშლიან ინფორმაციის გადაცემას, არამედ ის შეფერხებებიც, რომლებიც შეიძლება არსებობდეს მოწყობილობაში იწვევს ინფორმაციის დაკარგვას. ამდენად, ზოგადად, ყველა საინფორმაციო არხი შეიძლება ჩავთვალოთ არხებად დაბრკოლებებით.

განვიხილოთ დაბრკოლებამდგრადი კოდირების ამოცანის დასმის და მისი გადაწყვეტის საშუალებების შესაძლო გზები. შეტყობინების წყაროდან გამოდის სიმბოლოების თანმიმდევრობა $b_1, b_2, \dots, b_n$, რომლებიც იწოდებიან საინფორმაციო თანმიმდევრობებად. თითოეული საინფორმაციო თანმიმდევრობის კოდირებისას თითოეულს შეესაბამება ურთიერთ-ერთმნიშვნელოვანი შეტყობინება (კოდური თანმიმდევრობა) $\beta_1, \beta_2, \dots, \beta_n$. დაბრკოლებების მოქმედების შედეგად არხის გამოსასვლელზე მიიღება შეტყობინება $\beta'_1, \beta'_2, \dots, \beta'_n$, სადაც სიმბოლოების ნაწილი წმენდილობა განსხვავდებოდეს გადასაცემი შეტყობინების სიმბოლოებისგან: $\beta'_v \neq \beta_v$. თუ

შეტყობინების მიღებისას თითოეული სიმბოლო აუცილებლად იქნება იდენტიფიცირებული, არხის ანბანის სიმბოლოებიდან ერთ-ერთით (არხი წაშლის გარეშე), ე.ი. $\beta'_y \in B$ მაშინ შეცდომის ერთადერთი ფორმა შეიძლება იყოს სიმბოლოების ტრანსფორმაცია. თუ ზოგიერთი მიღებული სიგნალებიდან გადაწყვეტილება მათი იდენტიფიკაციის შესახებ B ანბანის განსაზღვრული სიმბოლოთი საერთოდ არ აღიქმება, მაშინ შესაძლებელია ადგილი ჰქონდეს შეცდომის ორ სახეს: სიმბოლოების ტრანსფორმაცია და სიმბოლოს არაღქმა (წაშლა). არაღქმადი (წაშლადი) სიმბოლოები ავლნიშნოთ ϕ -ით, სადაც $\Phi \notin B$.

შეტყობინებათა დაბრკოლებამდგრადი კოდირების ამოცანაა უზრუნველყოს საკმარისად მაღალი ალბათობით შეცდომების ან აღმოჩენა, ან აღმოჩენა და გასწორება. ამის შესაბამისად, განასხვავებენ კოდებს შეცდომების აღმოჩენით და კოდებს შეცდომების გასწორებით. შეცდომების აღმომჩენი კოდი უზრუნველყოფს შეტყობინების იმ ნაწილის წაშლას ან განსაკუთრებულ აღნიშვნას, რომელშიც აღმოჩენილია შეცდომები. ასეთი კოდი ეფექტურია უკუსაინფორმაციო კავშირიან სისტემებში, როცა არსებობს შესაძლებლობა განმეორებით მოხდეს შეტყობინების იმ ნაწილის მოთხოვნა, რომელშიც აღმოჩენილი იყო შეცდომები. უკუსაინფორმაციო კავშირების არ არსებობის შემთხვევაში კოდი შეცდომების აღმოჩენით ეფექტურია მხოლოდ ისეთ სისტემებში, რომლებიც ნაკლებად კრიტიკული არიან შეტყობინების წაშლადი ნაწილისა, ვიდრე მისი არასწორი მიღებისა. კოდი შეცდომების გასწორებით საშუალებას იძლევა მიღებულ იქნეს სანდო შეტყობინებები, მიუხედავად ცალკეული სიმბოლოების დამახინჯების არსებობისა (კოდის მაკორექტირებელი უნარის საზღვრებში). არ არსებობს სასრული სიგრძის კოდები, რომლებიც შეძლებდნენ ყველა შესაძლო შეცდომების აღმოჩენას და მით უფრო გასწორებას. ფაქტობრივად შესაძლებელია აიგოს მხოლოდ ისეთი კოდები, რომლებიც შეძლებენ აღმოაჩინონ ან გაასწორონ განსაზღვრული სახის შეცდომების გარკვეული რიცხვი. (განსაკუთრებულად უფრო ალბათური ან განსაკუთრებულად უფრო საშიში).

კოდები, რომლებიც ახორციელებენ შეცდომების აღმოჩენას და მათ აღმოფხვრას იწოდებიან დაბრკოლებამდგრად ან მაკორექტირებელ კოდებად. კოდის მაკორექტირებელი უნარი დაფუძნებულია მის სიჭარბეზე.

სიჭარბის არარსებობის შემთხვევაში არხის ანბანის სიმბოლოების ნებისმიერ კომბინაციის გადაცემისას შეუძლებელია შეცდომების უტყუარი აღმოჩენა.

კოდების მაკორექტირებელი უნარის განხილვისას გაითვალისწინება მხოლოდ ის სიჭარბე, რომელიც შეტანილია შეტყობინებაში კოდირების დროს. შეტყობინებები, რომლებიც მოდიან წყაროდან უნდა ჩაითვალოს სიჭარბის გარეშე. კერძო შემთხვევაში, როცა წყაროს და არხს გააჩნიათ ანბანის ერთნაირი მოცულობა m , სიჭარბის შემოტანა ნიშნავს გადასაცემ ინფორმაციის თანმიმდევრობაში სიმბოლოების რაოდენობის გაზრდას. საინფორმაციო თანმიმდევრობის დამოუკიდებლობასთან მიმართებაში. მაგალითად, n სიგრძის საინფორმაციო შეტყობინების ორობითი კოდისთვის ($m=2$) და საკონტროლო თანრიგების l რაოდენობის შემთხვევაში გადასაცემ შეტყობინებას გააჩნია სახე:

$$X = \{x_0, x_1, \dots, x_{n-1}, x_n, x_{n+1}, \dots, x_{n+l-1}\},$$

სადაც $x_i \in \{0,1\}$.

სიჭარბის შემოტანის ფორმის მიხედვით განასხვავებენ სისტემატურ და არასისტემატურ კოდებს.

სისტემატურ კოდებში ინფორმაციული თანმიმდევრობის სიმბოლოები მათში შედიან ცვლილების გარეშე გადასაცემ შეტყობინებაში, რომელშიც მათ უკავიათ სპეციალურად მათთვის გამოყოფილი პოზიციები, ხოლო კოდირების ოპერაცია დაყვანილია გადასაცემ შეტყობინებაში დამატებითი (ჭარბი) სიმბოლოების შეტანაზე, რომლებიც თავის მხრივ ინფორმაციული თანმიმდევრობითობის სიმბოლოებთან დაკავშირებული არიან განსაზღვრული დამოკიდებულებით. ისინი გადასაცემ შეტყობინებაში ცხადი სახით არ შედიან და შესაძლებელია დადგენილ იქნეს ცნობილი დამოკიდებულებებით, რომლებიც აკავშირებენ მათ გადასაცემი შეტყობინების სიმბოლოებთან.

კოდირების ხერხის მიხედვით განასხვავებენ ბლოკურ და უწყვეტ კოდებს.

ბლოკური კოდირებისას საინფორმაციო თანმიმდევრობა იყოფა ფიქსირებული k სიგრძის სიმბოლოების $(b_1, b_2, \dots, b_k)$ ბლოკებად, რომლებთაგან თითოეულ მათგანს შეესაბამება არხის ანბანის სიმბოლოების განსაზღვრული კომბინაცია $\beta_1, \beta_2, \dots, \beta_k$,

რომელსაც ეწოდება სიტყვის კოდი. კოდი ბლოკური კოდირებისას განსაზღვრავს კოდური სიტყვის ფორმირების კანონს.

სისტემატური ბლოკური კოდის შემთხვევაში კოდური სიტყვა შეიძლება წარმოდგენილ იქნეს შემდეგნაირად: $b_1, b_2, \dots, b_k, \beta_k, \beta_{k+1}, \dots, \beta_n$, სადაც $\beta_k, \beta_{k+1}, \dots, \beta_n$ -ჰარბი სიმბოლოებია.

ბლოკური კოდი, რომელსაც შესაბამისობაში მოჰყავს k სიგრძის ინფორმაციული სიმბოლოების ბლოკთან n სიგრძის კოდური სიტყვა აღინიშნება ასეთნაირად (k, n) . სიმბოლოები არ ფლობენ დამატებით ინფორმაციას შეტყობინების წყაროს შესახებ (ისინი ცალსახად განისაზღვრებიან ინფორმაციული სიმბოლოებით $(b_1, b_2, \dots, b_k)$, ამიტომ კოდური სიტყვა ფლობს სასარგებლო ინფორმაციის იმ რაოდენობას, რასაც ფლობს საინფორმაციო სიმბოლოების შესაბამისი ბლოკი. k სიგრძის ინფორმაციული სიმბოლოების ბლოკი შემავალი შეტყობინების არასიქარბის შემთხვევაში (ე.ი კოდური სიტყვაც) იქნება $H = K \log_2 m$ ინფორმაციის რაოდენობის მატარებელი, სადაც m - არის წყაროს ანბანის მოცულობა.

ინფორმაციის მაქსიმალური რაოდენობა, რომელსაც შეიძლება შეიცავდეს n სიმბოლოებისგან შემდგარი სიტყვა m ანბანის იგივე მოცულობისას ტოლია:

$H_{\max} = n \log_2 m$, ამიტომ (k, n) კოდის სიქარბე:

$$P_k = 1 - \frac{H}{H_{\max}} = 1 - \frac{k \log_2 m}{n \log_2 m} = \frac{n-k}{n} = \frac{r}{n},$$

სადაც $r = n - k$ - კოდირებისას ჰარბი სიმბოლოების რაოდენობაა.

უწყვეტი კოდირების დროს გადასაცემი შეტყობინების თითოეული სიმბოლო განისაზღვრება რეკურენტული თანაფარდობით, რომელიც აკავშირებს მას ინფორმაციული თანმიდევრობის შესაბამის სიმბოლოებთან:

$$\beta_{i,j} = \left[\sum_{v=-g}^{s-1} C_{vj} b_{i+v} \right] (\text{mod } m), \quad j = 0, 1, \dots, l-1. \quad (11.1)$$

ამ გამოსახულების მარჯვენა ნაწილის მნიშვნელობა განისაზღვრება m მოდულის მიხედვით იმისათვის, რომ β_{ij} სიმბოლოების მიღებული მნიშვნელობები მიეკუთვნებოდნენ არხის ანბანს, ე.ი მიიღონ ერთ მნიშვნელობა $0, 1, \dots, m-1$ მნიშვნელობათა სიმრავლიდან.

უწყვეტი კოდები შეიძლება იყოს, როგორც სისტემატური ასევე არასისტემატური. სისტემატური კოდები მიიღება კერძო შემთხვევაში, როცა j - ინდექსის ერთ-ერთი მნიშვნელობისათვის (მაგალითად, $j=0$) (11.1)) გამოსახულებაში კოეფიციენტები ღებულობენ მნიშვნელობას

$$C_{v_0} = \begin{cases} 0, & \text{როცა } v \neq 0, \\ 1, & \text{როცა } v = 0 \end{cases},$$

და შესაბამისად $\beta_{i0} = b_i$. ამ შემთხვევაში გადასაცემ თანამიმდევრობას გააჩნია შემდეგი სახე:

$$\dots b_i, \beta_{i,1}, \dots, \beta_{i,l-1}, b_{i+1}, \beta_{i+1,1}, \dots, \beta_{i+1,l} l+1, \dots$$

11.5 .2. შეცდომების მაკონტროლებელი კოდები

მოდულით კონტროლი წარმოადგენს საბორტო კომპიუტერებში ციფრული ინფორმაციის გადაცემის, შენახვის და დამუშავების კონტროლის ფართოდ გავრცელებულ მეთოდს. ამ მეთოდის დროს გამოიყენება ჭარბი ინფორმაცია, რომელიც წარმოდგენილია დამატებითი თანრიგებით საინფორმაციო სიტყვაში.

აღნიშნული მეთოდის არსი მდგომარეობს განსაზღვრული წესებით საკონტროლო კოდების განსაზღვრასა და ანალიზში, რომლებიც თავის მხრივ წარმოადგენენ უმცირეს ნაშთს თვით რიცხვების მოდულზე გაყოფისას (რიცხვითი კონტროლი) ან მათი ციფრების ჯამს (ციფრული კონტროლი).

პირველ შემთხვევაში საკონტროლო თანრიგები განისაზღვრება შემდეგი დამოკიდებულებით:

$$\alpha_m = \text{ნაშთი } x/m,$$

სადაც X - საკონტროლებელი რიცხვია;

α_m - უმცირესი ნაშთი, მიღებული X რიცხვის გაყოფილ m მოდულზე.

მაგალითად, ორის მოდულით კონტროლის დროს, $X=1001$ რიცხვისათვის ვღებულობთ

$$\alpha_2 = \text{ნაშთი } \frac{1001}{10} = 1.$$

მოცემულ მაგალითში ჭარბ კოდს აქვს მნიშვნელობა

$$x_2 = 1001 \underset{\substack{\uparrow \\ \text{საკონტროლო თანრიგი.}}}{1}$$

ცხადია, რომ ყველა კენტი რიცხვების საკონტროლო თანრიგი ტოლი იქნება ერთის, ხოლო ლუწი რიცხვების შემთხვევაში კი – ნულის. ასეთ შემთხვევაში კონტროლი დაიყვანება შემდეგ მოქმედებებზე: გადამცემ პოზიციაზე საკონტროლო თანრიგის ფორმირება; ჭარბი კოდის გადაცემა; მიმღებ ნაწილში იმავე წესებით ფორმირებული საკონტროლო თანრიგის შემოწმება, რომელიც მიღებულია საკონტროლო თანრიგის არხით. რიცხვითი კონტროლის შემთხვევაში ორის მოდულით შეცდომა არ იქნება აღმოჩენილი მაშინ, როცა დამახინჯების შემთხვევაში ლუწი რიცხვი დარჩება ლუწად, ხოლო კენტი რიცხვი კენტად. ასეთი მოვლენის ალბათობა ტოლია 0,5 - ის. რაც უფრო მაღალია მოდულის მნიშვნელობა, მით უფრო მეტი საკონტროლო თანრიგები ემატება გადასაცემ შეტყობინებაში. ამავე დროს უფრო დიდია შეცდომის აღმოჩენის ალბათობა.

ციფრული კონტროლის მეთოდი მოდულით გულისხმობს რიცხვის თანრიგების მნიშვნელობების შეჯამებას რაღაც მოდულით შემდეგი გამოსახულების შესაბამისად

$$\alpha_m = \left[\sum_{i=0}^{k-1} b_i \right] \text{mod } m,$$

სადაც b_i - რიცხვის i -ური თანრიგის მნიშვნელობა;

k -თანრიგების რიცხვი.

მაგალითად, 1001 რიცხვისთვის საკონტროლო თანრიგს, როცა $m=2$ ექნება მნიშვნელობა

$$\alpha_2 = [1+0+0+1] \text{mod } 2 = 0,$$

და ამიტომ ჭარბი კოდი იქნება ასეთი სახის

$x_2 = 10010$

საკონტროლო თანრიგი

ცხადია, რომ ასეთი კოდირებისას როცა $m=2$ ყველა შეცდომა, რომელიც იწვევს ერთი თანრიგის დამახინჯებას (სამი, ხუთი და ა. შ.) აღმოჩენილი იქნება. თუ დამახინჯდება ორი თანრიგი (ოთხი, ექვსი და ა.შ.), მაშინ შეცდომა არ იქნება აღმოჩენილი. თუმცა ალბათობა იმისა შეცდომა აღმოჩნდეს ერთდროულად 2 თანრიგში, არსებითად ნაკლებ სავარაუდოა ვიდრე ერთჯერადი შეცდომების წარმოშობა.

კონტროლის სქემურ მეთოდებს 2-ის მოდულით ასევე უწოდებენ კონტროლს ლუწობის მიხედვით. თუ საკონტროლო თანრიგებს ენიჭებათ ინვერსიული მნიშვნელობები (μ_2), მაშინ კონტროლი იწოდება კენტობის მიხედვით შემოწმებად.

ლუწობაზე (კენტობაზე) შემოწმების კონტროლი ხასიათდება დაბალი სიჭარბით და მისი რეალიზაცია არ ითხოვს ხელსაწყოების დიდ დანახარჯს. ეს მეთოდი ფართოდ გამოიყენება ციფრულ მოწყობილობებში ინფორმაციის გადაცემის კონტროლისათვის, როგორც მოწყობილობის შიგნით, ისე მის გარეთ, ასევე დამხსომებელი მოწყობილობიდან ამოკითხული ინფორმაციის კონტროლისათვის.

11.5.3. შეცდომების გამსწორებელი კოდები. ხემინგის კოდი

გამსწორებელ კოდებში ორობითი რიცხვი, რომელიც შედგენილია საკონტროლო თანრიგებისგან (კოდის სინდრომი), უნდა განსაზღვროს კოდური სიტყვის დამახინჯებული თანრიგის რიგითი ნომერი. შემდეგ ხორციელდება ამ თანრიგის ინვერტირება, რითაც მიიღწევა მისი კორექცია.

აღნიშნული კოდების მაგალითს წარმოადგენს ხემინგის კოდი. ამ კოდში საკონტროლო ჯამი დგება ისეთნაირად, რომ ჯამში S_j უნდა შევიდეს კოდური სიტყვის სიმბოლოები, რომლებსაც მათი რიგითი ნომრების ორობით ჩაწერაში j -ურ თანრიგში გააჩნიათ ერთიანი.

მნელი არ არის დავინახოთ, რომ სიმბოლოები, რომელთა რიგითი ნომერი ორობით ჩაწერაში გამოისახება j -ური თანრიგის ერთიანით, ე.ი. ტოლია

$$2^{j-1}, j = 1, 2, 3, \dots,$$

შევლენ მხოლოდ ერთ საკონტროლო ჯამში S_j . სახელდობრ ამ პოზიციებში $(1,2,4,8,...,2^i)$ ხემინგის კოდში განლაგდებიან შესამოწმებელი სიმბოლოები $(\beta_1, \beta_2, \beta_4, \beta_8, \dots, \beta^{2^i})$. მათი არჩევის შესაბამისად ყველა საკონტროლო ჯამი ხდება ნულის ტოლი. დამახინჯების არ არსებობის შემთხვევაში ყველა საკონტროლო ჯამების მნიშვნელობა ნულის ტოლია. ერთჯერადი შეცდომა, მაგალითად, კოდური სიტყვის I -ური თანრიგის დამახინჯება გამოიწვევს საკონტროლო ჯამების ერთიანში გადასვლას, რომელთა ნომრები (საკონტროლო ჯამების ნომრები) შეესაბამებიან I რიცხვის ორობითი ჩაწერის ერთიანობის შემცველ თანრიგების ნომრებს. ამიტომ თუ მოვახდენთ საკონტროლო ჯამების მნიშვნელობის ჩაწერას მათი ნომრების შემცირების მიმართულებით, ე.ი. სინდრომის ჩაწერისას S_4, S_3, S_2, S_1 , ვღებულობთ ერთი დამახინჯებული თანრიგის ორობით ჩაწერას.

თვალსაჩინოებისთვის კოდური სიტყვის თანრიგების ნომრების ორობით სისტემაში ჩაწერის მაგალითი წარმოდგენილია ცხრილში 11.1 (ცხრილი მოყვანლია იმ შემთხვევისათვის, როცა ინფორმაციული შეტყობინებები, წარმოდგენილია თხუთმეტათანრიგის კოდების სახით).

ცხრილის შესაბამისად თხუთმეტათანრიგ კოდის საკონტროლო ჯამებს გააჩნიათ შემდეგი სახე:

$$\begin{aligned} s_1 &= \beta_1 \oplus b_3 \oplus b_5 \oplus b_7 \oplus b_9 \oplus b_{11} \oplus b_{13} \oplus b_{15}; \\ s_2 &= \beta_2 \oplus b_3 \oplus b_6 \oplus b_7 \oplus b_{10} \oplus b_{11} \oplus b_{14} \oplus b_{15}; \\ s_3 &= \beta_4 \oplus b_5 \oplus b_6 \oplus b_7 \oplus b_{12} \oplus b_{13} \oplus b_{14} \oplus b_{15}; \\ s_4 &= \beta_8 \oplus b_9 \oplus b_{10} \oplus b_{11} \oplus b_{12} \oplus b_{13} \oplus b_{14} \oplus b_{15}, \end{aligned} \quad (11.2)$$

სადაც b_i – კოდის საინფორმაციო სიმბოლოებს მნიშვნელობა;

$\beta_1, \beta_2, \beta_4, \beta_8$ – შესამოწმებელი სიმბოლოების მნიშვნელობა;

$\oplus$ – ორის მოდულით აჯამვის ნიშანი.

შესამოწმებელი სიმბოლოების მნიშვნელობები, რომლებიც საკონტროლო ჯამებს ასახავენ ნულში ასეთია:

ცხრილი 11.1

თანრიგის რიგითი ნომერი	კოდის თანრიგების რიგითი ნომრების ორობითი ჩაწერის თანრიგების მნიშვნელობები			
	4	3	2	1 (უმცროსი)
1 (უმცროსი)	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

$$\begin{aligned}
 \beta_1 &= b_3 \oplus b_5 \oplus b_7 \oplus b_9 \oplus b_{11} \oplus b_{13} \oplus b_{15}; \\
 \beta_2 &= b_3 \oplus b_6 \oplus b_7 \oplus b_{10} \oplus b_{11} \oplus b_{14} \oplus b_{15}; \\
 \beta_4 &= b_5 \oplus b_6 \oplus b_7 \oplus b_{12} \oplus b_{13} \oplus b_{14} \oplus b_{15}; \\
 \beta_8 &= b_9 \oplus b_{10} \oplus b_{11} \oplus b_{12} \oplus b_{13} \oplus b_{14} \oplus b_{15},
 \end{aligned}
 \tag{11.3}$$

ხემინგის გამსწორებელი კოდის ფორმირების პროცედურა შეიცავს შემდეგ ოპერაციებს:

1. საინფორმაციო სიმბოლოების განლაგებას თანრიგების ზრდის მიმართულებით (მარჯვნიდან მარცხნივ) 2^{j-1} ნომრების მქონე დაუკავებელი თანრიგების შენახვით, (თანრიგების ნომრები 1, 2, 4, 8 და ა.შ.)

2. შესამოწმებელი სიმბოლოების გამოთვლა (11.3) –ის ფორმულის მიხედვით რომლებიც იკავებენ 2^{j-1} ნომრის მქონე თანრიგებს.

ხემინგის გამსწორებელი კოდის გამოყენებისას გადასაცემი საინფორმაციო კოდის განსაზღვრის პროცედურა შეიცავს:

1. (11.2) ფორმულებით საკონტროლო ჯამების გამოთვლა;
2. დამახინჯებული სიმბოლოს რიგითი ნომრის გასაზღვრა, თუ ყველა საკონტროლო ჯამი უდრის ნულს და მისი გასწორება სხვა ორობით სიმბოლოზე;
3. ინფორმაციული კოდის გამოყოფა 2^{j-1} , $j=1,2,\dots$ ნომრების მქონე თანრიგების გამორიცხვით.

მაგალითად, თუ საინფორმაციო კოდს გააჩნია სახე 101001, მაშინ მისი სიმბოლოები ხემინგის კოდში განლაგდება შემდეგი თანმიმდევრობით:

$$\begin{array}{cccccccc} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 0 & \beta_8 & 1 & 0 & 0 & \beta_4 & 1 & \beta_2 & \beta_1 \end{array}$$

განსაზღვროთ შესამოწმებელი სიმბოლოების მნიშვნელობა (11.3) ფორმულებით:

$$\begin{aligned} \beta_1 &= b_3 \oplus b_5 \oplus b_7 \oplus b_9 = 1 \oplus 0 \oplus 1 \oplus 0 = 0; \\ \beta_2 &= b_3 \oplus b_6 \oplus b_7 \oplus b_{10} = 1 \oplus 0 \oplus 1 \oplus 1 = 1; \\ \beta_4 &= b_5 \oplus b_6 \oplus b_7 = 0 \oplus 0 \oplus 1 = 1; \\ \beta_8 &= 0 \end{aligned}$$

საკონტროლო თანრიგების გათვალისწინებით მივიღებთ შემდეგ კოდს:

$$1001001110$$

თუ, მაგალითად, აღნიშნული კოდის მეხუთე თანრიგი მიღებულ იქნება შეცდომით როგორც ერთიანი მაშინ სინდრომებს (საკონტროლო ჯამებს) ექნებათ მნიშვნელობა:

$$\begin{aligned}
s_1 &= \beta_1 \oplus b_3 \oplus b_5 \oplus b_7 \oplus b_9 = 0 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 1; \\
s_2 &= \beta_2 \oplus b_3 \oplus b_6 \oplus b_7 \oplus b_{10} = 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0; \\
s_3 &= \beta_4 \oplus b_5 \oplus b_6 = 1 \oplus 1 \oplus 0 \oplus 1 = 1; \\
s_4 &= \beta_8 = 0
\end{aligned}$$

სინდრომის მნიშვნელობა, რომელიც განსაზღვრავს დამახინჯებული თანრიგის ნომერს, უდრის $S_4 S_3 S_2 S_1 = 0101$, რაც შეესაბამება ათობითი ათვლის სისტემის ციფრი 5-ის წარმოდგენას ორობით კოდში. მაშასადამე, შეცდომის გასწორებისათვის საჭიროა მეხუთე თანრიგში სიმბოლოს გასწორება.

11.5.4. კასკადური კოდები

შეცდომების აღმოჩენისა და გასწორების კოდის შესაძლებლობა განისაზღვრება კოდის სიჭარბის სიდიდით. ამასთან ინფორმაციის გადაცემის უშეცდომობა მიიღწევა მხოლოდ რაც შეიძლება მნიშვნელოვანი სიგრძის კოდური სიტყვების შემთხვევაში. კოდირების და დეკოდირების ეფექტური სქემების ძიებამ, რომელთა რეალიზაციის სირთულე კოდის სიგრძის ზრდასთან ერთად გაზრდილიყო რაც შეიძლება უფრო ნელა, გამოიწვია კასკადური კოდების გამოჩენა.

განზოგადებული კასკადური კოდი განისაზღვრება ასე. α ორობითი სიტყვა სიგრძით $n=n_1 \cdot n_2$ წარმოადგენს m რიგის განზოგადებული კასკადური კოდის კოდურ სიტყვას მხოლოდ მაშინ, როცა α სიტყვასთან დაკავშირებული ვექტორები Y_i , $i=1, m+1$, წარმოადგენენ თავის მხრივ მეორე საფეხურის შესაბამის $(i-x)$ კოდების კოდურ სიტყვას.

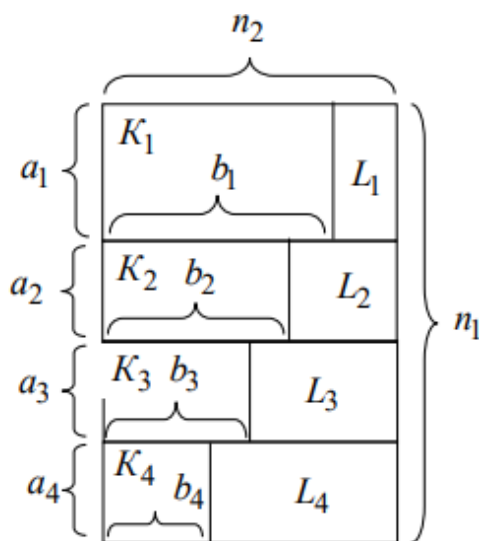
განზოგადებულ კასკადურ კოდს შეგვიძლია შევუსაბამოთ საკმარისად თვალსაჩინო და მოხერხებული გეომეტრიული ტრაქტოვკა მრავალი განსჯის დროს.

α ორობითი სიმბოლოების თანმიმდევრობა რომელიც ქმნის განზოგადებული კასკადურ კოდის კოდურ სიტყვას განვალაგებთ $n_1 \cdot n_2$ ზომის მართკუთხა ცხრილში ,

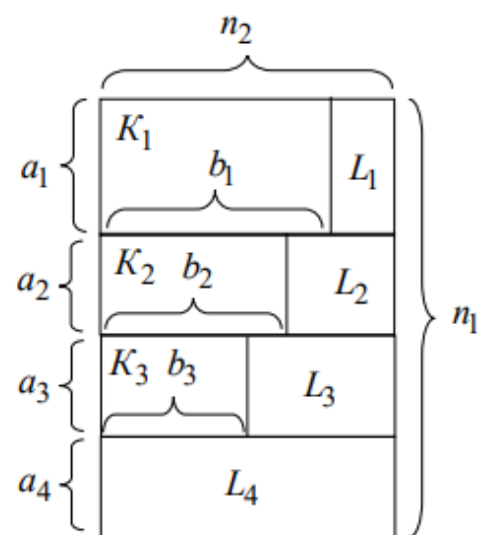
ისეთნაირად, რომ α_j ვექტორები იქნებიან მისი სვეტები. მაშინ ვექტორები $\alpha_{j1}, \alpha_{j2}, \dots, \alpha_{jn2}$ შექმნიან ამ ცხრილის i -ურ ჰორიზონტალურ ბლოკს B_i , რომელთა საერთო რიცხვი ტოლია $m+1$ -ის, სადაც m კოდის რიგია.

თითოეული ჰორიზონტალური ბლოკი B_i , შემდგარი α_j სტრიქონისაგან, თავის მხრივ შეგვიძლია დავყოთ ორ ბლოკად - K_i და L_i , რომლებიც შესაბამისად შეიცავენ b_i და $n_2 - b_i$ სვეტებს.

K_i ბლოკებში განლაგებულია კასკადური კოდის ინფორმაციული სიმბოლოები, ხოლო L_i ბლოკებში კი მისი შემმოწმებელი სიმბოლოები (ნახ.11.1). თუ $b_{m+1}=0$, მაშინ B_{m+1} ბლოკი ემთხვევა L_{m+1} -ს და შეიცავს მხოლოდ შემმოწმებელ სიმბოლოებს (ნახ.11.2).



ნახ.11.1



ნახ.11.2

ავღნიშნოთ, რომ სიდიდეები $a_i > 0$ და $b_i \geq 0$, რომლებიც განსაზღვრავენ განზოგადებული კასკადური კოდის შიგა სტრუქტურას, შეიძლება ამოირჩეს ნებისმიერად, ამასთან ინფორმაციული სიმბოლოების საერთო რიცხვი K და შემმოწმებელი სიმბოლოების რიცხვი ტოლია

$$k = \sum_{i=1}^{m+1} a_i b_i; \quad r = \sum_{i=1}^{m+1} a_i (n_2 - b_i).$$

მაღალსაფეხურიანი კოდირების გამოყენება, სადაც მის თითოეულ საფეხურზე გამოიყენება თავისი ჭარბი კოდი, უზრუნველყოფს კოდირების მაღალ ეფექტურობას სიჭარბის შედარებით მცირე მნიშვნელობების დროს.

შეგვიძლია მოვახდინოთ სქემური კონტროლის ეფექტურობის ალბათური შეფასება საბორტო კომპიუტერებში. საბორტო კომპიუტერში დიდი უტყუარობით შეგვიძლია ჩავთვალოთ შეცდომები კოდის სხვადასხვა თანრიგებში X დამოუკიდებელ შემთხვევით სიდიდეებად, რომლებსაც ექნებათ წარმოშობის ერთნაირი ალბათობა. მაშინ i -ჯერადი შეცდომებისთვის n - თანრიგიან კოდში სამართლიანია განაწილების ბინომალური კანონი, ხოლო n - თანრიგა მოწყობილობაში i -ს ტოლი შეცდომების წარმოშობის ალბათობას გააჩნია ასეთი სახე:

$$P_i = C_n^i q^i (1-q)^{n-i}.$$

აქ q განისაზღვრება შემდეგი გამოსახულებით

$$q = 1 - e^{-\sum_{j=1}^N \lambda_j t},$$

და ის არის ერთი შეცდომის მაინც გამოვლენის ალბათობა ცალკეულ თანრიგში ერთი ოპერაციის განმავლობაში; λ_j არის j -ური ელემენტის მტყუნებათა ინტენსიურობა; N - თანრიგში ელემენტების რაოდენობა.

თუ მივიღებთ, რომ ცალკეულ თანრიგში შეცდომის გაჩენის ალბათობა მაკონტროლებელ და საკონტროლო ნაწილებში ერთნაირია და ტოლია, მაშინ i -ზე არანაკლები შეცდომების წარმოშობის ალბათობა მაკონტროლებელ და საკონტროლო მოწყობილობებში იქნება

$$P = \sum_{i=1}^{n+n'} C_{n+n'}^i q^i (1-q)^{n+n'-i},$$

სადაც n - საკონტროლო თანრიგების რაოდენობაა.

სქემური კონტროლის ეფექტურობის შეფასებისათვის შემოიტანება მოწყობილობის ხარისხის კოეფიციენტი კონტროლით

$$k(t) = 1 - \sum_{i=2}^{n+n'} \bar{P}_i C_{n+n'}^i q^i (1-q)^{n+n'-i},$$

სადაც $\bar{P}_i$ - i ჯერადი შეცდომის არ აღმოჩენის ალბათობა (თუ კონტროლი აღმოაჩენს i -ს ჯერად შეცდომას, მაშინ $\bar{P}_i = 0$).

პირობების შემთხვევაში როცა $n' \ll n$, $i \ll n$ და $m \leq 7$ შეგვიძლია ვაჩვენოთ, რომ არ აღმოჩენის ალბათობა i - ჯერადი შეცდომისა m მოდულის მიხედვით კონტროლის დროს ტოლია

$$\bar{P}_i = 1/m + (m-1)/m(-1/(m-1))^i.$$

სქემური კონტროლი ყოველთვის არ უზრუნველყოფს საბორტო კომპიუტერის ფუნქციონირების სისწორის შემოწმებას აუცილებელი ხარისხით. ამიტომ კონტროლის სისტემის ყველაზე რაციონალურ ორგანიზაციას წარმოადგენს კონტროლის კომბინირებული მეთოდი, რომელიც აკმაყოფილებს საბორტო კომპიუტერების კონტროლის სისტემების მიმართ წაყენებულ ყველა მოთხოვნას. კონტროლის კომბინირებული მეთოდის გამოყენებისას ძირითად პრობლემას წარმოადგენს ოპტიმალური თანაფარდობის განსაზღვრა კონტროლის პროგრამულ და სქემურ მეთოდებს შორის, რომელიც დამოკიდებულია საბორტო კომპიუტერის მუშაობის პირობებსა და გამოყენებაზე.

ლექცია 12

საბორტო გამოთვლითი სისტემები

12.1. საბორტო გამოთვლითი სისტემების კლასიფიკაცია

დღეს საჰაერო ხომალდებისადმი წაყენებული მოთხოვნებიდან ერთ-ერთი უმთავრესია საბორტო რადიოელექტრონული კომპლექსის შემადგენლობაში შემავალი გამოთვლითი საშუალებების სრულყოფა. ეს კი პირველ რიგში გულისხმობს სრულყოფილი გახდეს გამოთვლითი ტექნიკის საბორტო საშუალებები. ასეთი სრულყოფის მიზანია გამოთვლითი ოპერაციების სიჩქარის ამაღლება, საიმედოობისა და სიცოცხლისუნარიანობის გაზრდა, ტექნიკურ-ეკონომიკური მაჩვენებლების გაუმჯობესება. გამოთვლითი ტექნიკის პროგრესი დაფუძნებულია ახალ არქიტექტურულ და სტრუქტურულ გადაწყვეტათა მიღებაზე, ინფორმაციის დამუშავების და გამოთვლითი საშუალებების აგების ახალი პრინციპების რეალიზაციაზე.

კომპიუტერების და მათ შორის საბორტო კომპიუტერების განვითარების ხანგრძლივი პერიოდი მდგომარეობდა ჯონ ფონ ნეიმანის მიერ შემოთავაზებული სტრუქტურის სრულყოფაში (შემოთავაზება მოხდა გასული საუკუნის 40-იანი წლების ბოლოს). გამოთვლების ავტომატიზაციის კონცეპტუალურ საფუძველს წარმოადგენდა ადამიანი - გამომთვლელის მოდელი. კომპიუტერების სწრაფქმედების ამაღლება და მუშაობის სიჩქარე თეორიულად მიღწევად მნიშვნელობასთან მიმართებაში ახლოსაა მათი წარმოებისათვის გამოყენებული ტექნოლოგიის შემთხვევაში.

ერთეული გამომთვლელის მოდელის შეცვლა მოხდა გასული საუკუნის 60-იანი წლების დასწყისში, როცა აკადემიკოსმა ევრეინოვმა ერთეული გამომთვლელის მოდელის მაგივრად შემოიტანა გამომთვლელის კოლექტივის მოდელი. ეს მოდელი არ აწესებს შეზღუდვებს გამოთვლების სიჩქარეზე, ასევე საიმედოობისა და სიცოცხლისუნარიანობის მაჩვენებლების მნიშვნელობებზე.

გამოთვლების კოლექტიური მოდელი ძირითადი პრინციპი მდგომარეობს ოპერაციათა შესრულების პარალელურობაში, სტრუქტურის პროგრამირებადობაში და ელემენტების კონსტრუქციულ ერთგვაროვნობაში. აპარატურ - პროგრამული საშუალებები, რომლებიც აკმაყოფილებენ მოცემულ პრინციპებს (შეიძლება ნაწილობრივადაც კი), უწოდებენ გამოთვლით სისტემებს.

საბორტო ციფრული გამოთვლით სისტემების (სცგს) გამოჩენა განპირობებული იყო არსებული საბორტო კომპიუტერების მახასიათებლებისა და პარამეტრების მნიშვნელოვანი შეუსაბამობით საბორტო კომპლექსების მოთხოვნებთან მწარმოებლურობისა და სიზუსტის თვალსაზრისით.

დღესდღეობით თანამედროვე ინტეგრალური ტექნოლოგიის წარმატებები განაპირობებს კომპიუტერების მასის, მოცულობის, მოთხოვნილი სიმძლავრის შემცირებას, ხოლო მათ მიერ გადასაწყვეტი ამოცანების გადაჭრის პარალელიზმის შესაძლებლობა უფრო გაადვილებულს ხდის ბორტზე განლაგდეს რამდენიმე ურთიერთდაკავშირებული მოწყობილობა ინფორმაციის დამუშავებისა და შენახვისათვის.

გამოთვლითი სტრუქტურების დიდი რიცხვის არსებობამ განაპირობა მათი კლასიფიკაციის სხვადასხვა ვარიანტების გაჩენა. კლასიფიკაციის სქემის არჩევისას აუცილებელია ყურადღება გამახვილდეს შემდეგ აუცილებელ საკითხებზე:

- გამოთვლითი სტრუქტურების ყველა კლასიფიკაციის განხილვა, როგორც არსებული ისე წინასწარგანჭვრეტითი თვალსაზრისით;
- არსებითად სხვადასხვა გამოთვლითი სტრუქტურების დიფერენციაცია;
- ნებისმიერი გამოთვლითი სტრუქტურის კლასიფიკაციის ერთმნიშვნელოვნობა;
- საკლასიფიკაციო სქემის თვალსაჩინოება, სიმარტივე და პრაქტიკული მიზანშეწონილობა.

ამჟამად პრაქტიკული გამოყენება ჰპოვეს შემდეგმა საკლასიფიკაციო სქემებმა:

- ფლინის კლასიფიკაციის გაფართოებული სქემა;
- შორის საკლასიფიკაციო სისტემა;
- კლასიფიკაციის ერლანგენური სქემა.

ცხრილში 12.1 მოყვანილია ჩამოთვლილი საკლასიფიკაციო სქემების საკლასიფიკაციო ნიშან-თვისებები.

არსებული საბორტო ციფრული გამოთვლითი სისტემები ყველაზე უფრო სრულად აღიწერება ფლინის საკლასიფიკაციო სქემით, ამიტომ მიზანშეწონილია ის განვიხილოთ დაწვრილებით.

ფლინის კლასიფიკაციის გაფართოებული სქემის ორი პირველი ნიშან-თვისების შესაბამისად, რომელიც ეხება ბრძანებებისა და მონაცემების ნაკადების კლასიფიკაციას, შემოიტანება გამოთვლითი სისტემების ოთხი ძირითადი კლასის ცნება:

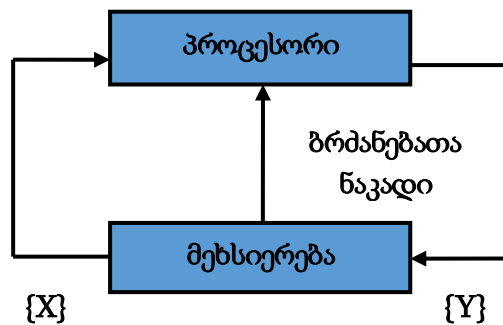
1. სისტემები ბრძანებათა ერთჯერადი ნაკადით და მონაცემთა ერთჯერადი ნაკადით;
2. სისტემები ბრძანებათა ერთჯერადი ნაკადით და მონაცემთა მრავლობითი ნაკადით;
3. სისტემები ბრძანებათა მრავლობითი ნაკადით და მონაცემთა ერთჯერადი ნაკადით;
4. სისტემები ბრძანებათა და მონაცემთა მრავლობითი ნაკადებით.

ბრძანებათა ნაკადის ცნების ქვეშ იგულისხმება ბრძანებათა თანმიმდევრობითი ნაკადი, რომელსაც ასრულებს სისტემა, ხოლო მონაცემთა ნაკადი - ეს არის მონაცემთა თანმიმდევრობითი რიგი, რომელიც გამოწვეულია ბრძანებათა ნაკადით და გულისხმობს საშუალოდ შედეგებსაც. ბრძანებათა ან მონაცემთა მრავლობითობა განისაზღვრება, როგორც ოპერაციების (ბრძანების) ან ოპერანდების (მონაცემების) მაქსიმალური შესაძლო რიცხვი, რომლებიც იმყოფებიან დამუშავების ერთნაირ სტადიაში.

გამოთვლით სისტემების გამარტივებული სტრუქტურული სქემები მოცემულია ნახაზებზე 12.1 – 12.3 (ნახ.12.1 - სისტემა ბრძანებათა ერთჯერადი ნაკადით და მონაცემთა ერთჯერადი ნაკადით, ნახ.12.2 - სისტემა ბრძანებათა ერთჯერადი ნაკადით და მონაცემთა მრავლობითი ნაკადით, ნახ.12.3 - სისტემა ბრძანებათა მრავლობითი ნაკადით და მონაცემთა ერთჯერადი ნაკადით, ნახ.12.4 - სისტემა ბრძანებათა და მონაცემთა მრავლობითი ნაკადებით). ნახაზებზე მონაცემთა ნაკადები დამუშავებისათვის აღნიშნულია {X}-ით, ხოლო მონაცემთა ნაკადი დამუშავების შემდეგ (შედეგები) - {Y}-ით.

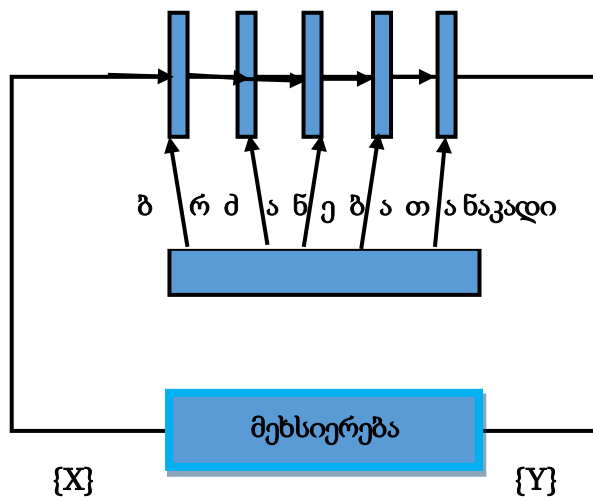
ცხრილი 12.1

საკლასიფიკაციო სქემა	კლასიფიკაციის ნიშან-თვისებები
ფლინის	ბრძანებათა ნაკადის ტიპი; მონაცემთა ნაკადის ტიპი; მონაცემთა დამუშავების ხერხი; ფუნქციური ელემენტების ბმულობის (კავშირობის) ხარისხი; საბოლოო ციფრული გამოთვლით სისტემების ელემენტებს შორის კავშირის ტიპი (სახე)
შორის	გამოთვლით სისტემის შემადგენელი ნაწილების შეერთების ხერხი. მონაცემების დამუშავების ხერხი
ერლანგენის	მართვის მოწყობილობათა რიცხვი, დამუშავების მოწყობილობათა რიცხვი; მანქანური სიტყვის სიგრძე

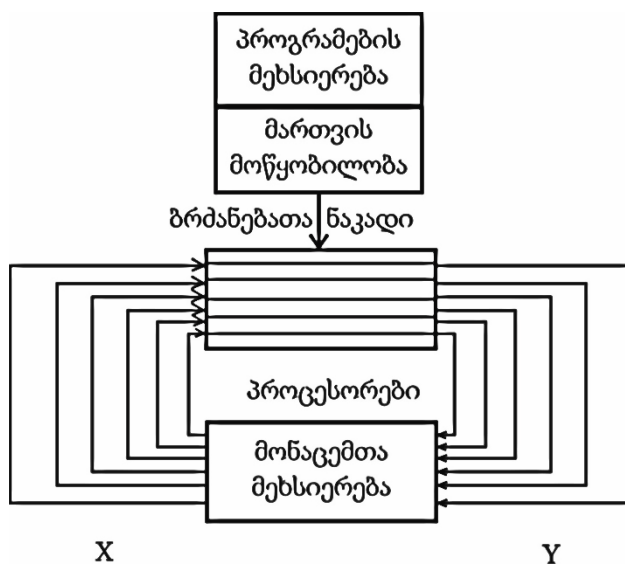


ნახ. 12.1

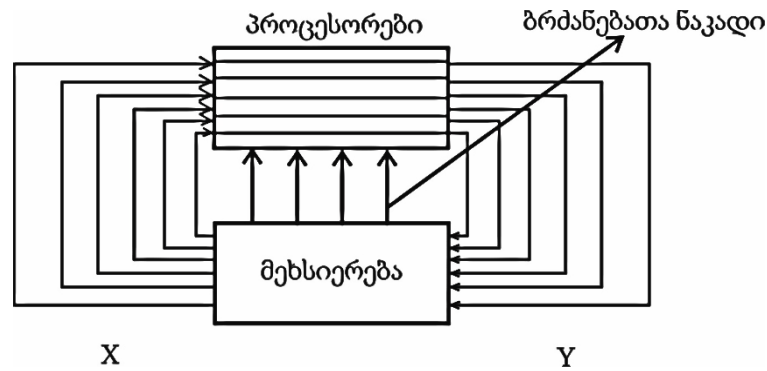
დამუშავების მაგისტრალი
(პროცესორების კონვეიერი)



ნახ.12.2

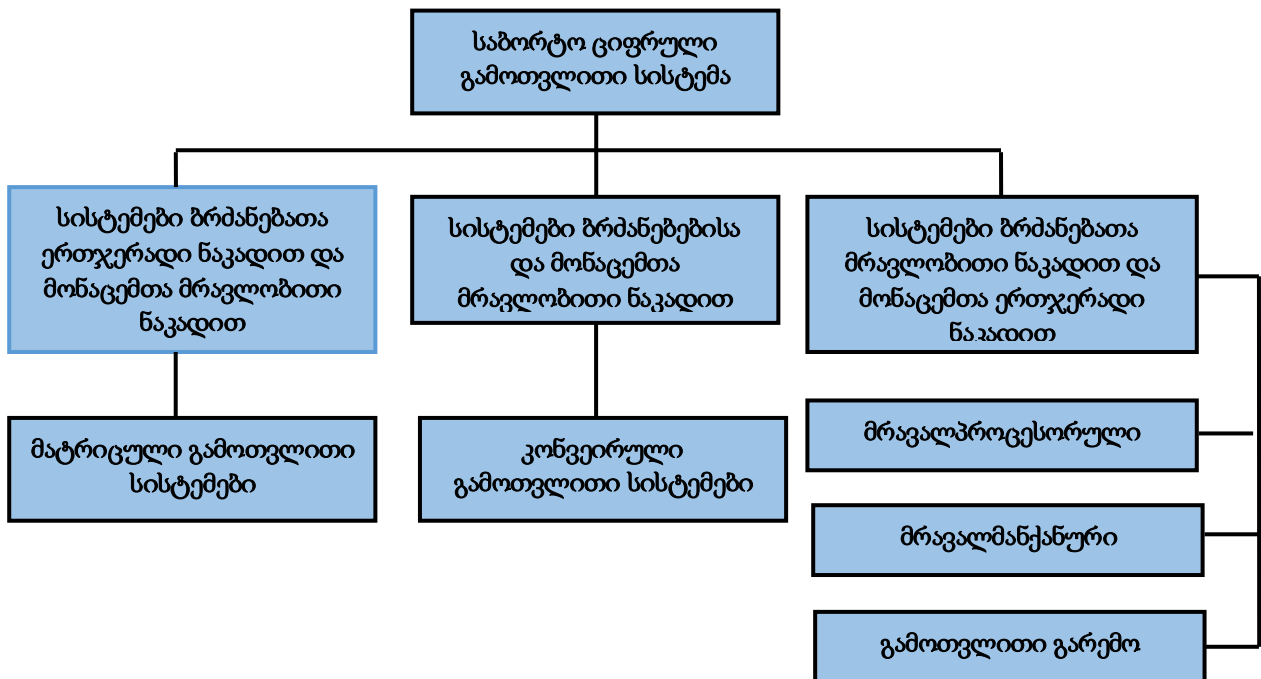


ნახ.12.3



ნახ.12.4

ამჟამად ყველაზე უფრო გამოყენებული საბორტო ციფრული გამოთვლითი სისტემების ძირითადი ტიპები ფლინის საკლასიფიკაციო სქემის შესაბამისად მოცემულია ნახ.12.5-ზე.



ნახ.12.5. საბორტო ციფრული გამოთვლითი სისტემების ძირითადი ტიპები

მატრიცული გამოთვლითი სისტემები დაფუძნებულია იმაზე, რომ საერთო მართვის მოწყობილობა დამოუკიდებელი ერთგვაროვანი პროცესორული ელემენტებისათვის განსაზღვრავს ერთნაირ ბრძანებებს პროგრამის გამოთვლის თითოეული ეტაპისთვის (ნახ.12.3). ყველა პროცესორი ერთდროულად ასრულებს ერთსა და იმავე ოპერაციას -

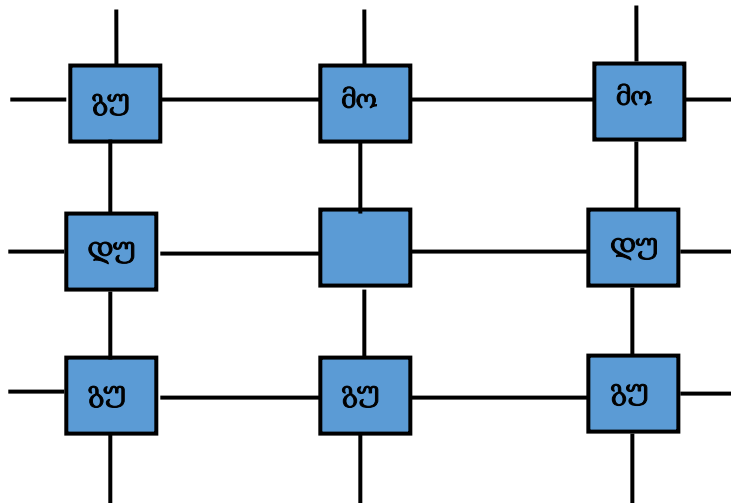
თითოეული მათგანი რა თქმა უნდა თავიანთ მონაცემებზე, რომლებიც თავის მხრივ შენახულია პროცესორების ლოკალურ დამხსომებელ მოწყობილობებში. მონაცემები უნდა იყოს ერთტიპიური (მაგალითად, მატრიცის ელემენტებით).

კონვეირული გამოთვლითი სისტემის მუშაობა დაფუძნებულია ერთდროულად ორი ან უფრო მეტი ბრძანების შესრულებაზე, თუმცა ეს შეიძლება ხდებოდეს სხვადასხვა სტადიებზე (ნახ.12.2). მაგალითად, ერთი ბრძანება შეიძლება იმყოფებოდეს დასრულების მდგომარეობაში, მეორე კი მოლოდინის მდგომარეობაში, როცა ის ელოდება ოპერანდს ოპერატიული მეხსიერებიდან. მართვის საერთო მოწყობილობა ემსახურება რამდენიმე სხვადასხვა დანიშნულებისა და სტრუქტურის ფუნქციურ პროცესორებს. თითოეული ფუნქციური პროცესორი ასრულებს ოპერაციას მონაცემთა ბლოკებზე, რის დასრულების შემდეგ გადასცემს მას მომდევნო პროცესორს, ხოლო თვითონ წინა პროცესორიდან ღებულობს დამუშავებისათვის მონაცემთა ახალ ბლოკს.

მრავალმანქანური ციფრული გამოთვითი სისტემა თავის მხრივ წარმოადგენს კომპლექსს, რომელიც აერთიანებს რამდენიმე კომპიუტერს, რომლებთან თითოეული თავის შემადგენლობაში შეიცავს ცენტრალურ პროცესორს, მეხსიერებას და შეტანა-გამოტანის არხებს. ასეთი ციფრული გამოთვლითი სისტემების ცალკეული ნაწილი (ერთ-ერთი კომპიუტერი), შეიძლება წარმოადგენდეს ასევე გამოთვლით სისტემას.

მრავალპროცესორული ციფრული გამოთვლითი სისტემა - ეს არის გამოთვლითი საშუალებების ერთობლიობა, რომელიც შეიცავს ორ ან უფრო მეტ პროცესორს, რომლებიც ფუნქციონირებენ ერთიანი ოპერაციული სისტემის მართვის მიხედვით საერთო მეხსიერებისა და ინფორმაციული გაცვლის არხების ერთდროული გამოყენების საფუძველზე.

ერთგვაროვანი გამოთვლითი გარემოს მქონე გამოთვლითი სისტემები, წარმოადგენენ ერთ-ერთ სახეობას მრავალპროცესორული ციფრული გამოთვლითი სისტემების ნუსხიდან, რომელსაც გააჩნია ორგანზომილებიანი გისოსური სტრუქტურა, რომლის კვანძებში განლაგებულია ინფორმაციის დამუშავების ფუნქციური მოწყობილობები (ნახ.12.6).

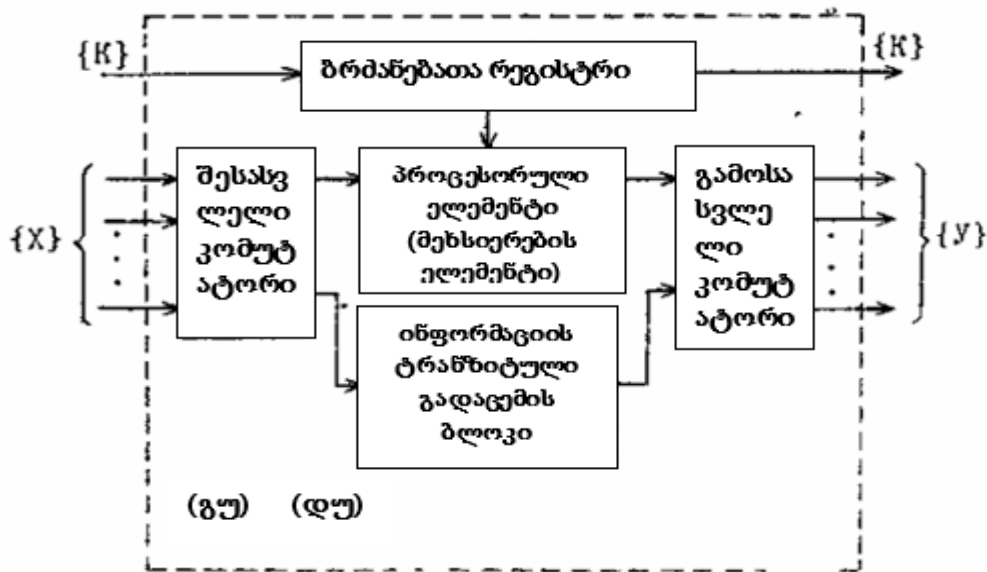


ნახ.12. 6. ერთგვაროვანი გამოთვლითი გარემოს ორგანოზომილებიანი
გისოსის სტრუქტურა

ეს მოწყობილობები თავის მხრივ წარმოადგენენ ლოგიკურ ავტომატებს, რომლებიც იწოდებიან გამოთვლით უჯრედებად (გუ). გამოთვლითი უჯრედები უზრუნველყოფენ ფუნქციურად ელემენტარულ ოპერაციების სრული ნაკრების შესრულებას.

ერთგვაროვანი გამოთვლითი გარემოს ფუნქციონირების ორგანიზაცია იგება გამოთვლითი კოლექტივის მოდელების საფუძველზე ოპერაციათა ნებისმიერი რიცხვის პარალელური შესრულების პრინციპების გამოყენებით. გამომთვლელის სტრუქტურა აიწყობა პროგრამულად კონკრეტულ გადასაწყვეტ ამოცანაზე. ამასთან თითოეული გამოთვლითი უჯრედისათვის განისაზღვრება მის მიერ შესასრულებელი ფუნქცია (ფუნქციები) და კავშირები მეზობელ უჯრედებთან.

ერთგვაროვანი გამოთვლითი გარემოს სტრუქტურაში პარალელურად მიმდინარე გამოთვლითი პროცესების დროში შესათანხმებლად შესაძლებელია შემოტანილ იქნეს ელემენტარული დამხსომეხელი უჯრედები (დუ), რომლებიც ახორციელებენ მონაცემების დამახსოვრებას ტაქტების განსაზღვრული რაოდენობის შემთხვევაში. გამოთვლითი დამხსომეხელი ელემენტების შიგა ორგანიზაცია ერთნაირია და შეიცავს დამხსომეხელ (გამოთვლით) მოწყობილობას, შესასრულებელი ფუნქციების პროგრამული აწყობის საშუალებებს და ინფორმაციის ტრანზიტული გადაცემის ლოგიკურ მოწყობილობას (ნახ.12.7).



ნახ.12.7

თანამედროვე გამანადგურებლების საბორტო გამოთვლითი სისტემა წარმოადგენს მრავალმანქანურ სისტემას, ამასთან ზოგიერთი გამოთვლითი მანქანები წარმოადგენენ მრავალპროცესორიანს (მაგალითად, II-100 შეიცავს ძირითად პროცესორს და შეტანა-გამოტანის პროცესორს).

მრავალმანქანური საბორტო ციფრული გამოთვლითი სისტემების სტრუქტურა და მახასიათებლები განისაზღვრება თითოეული კონკრეტული გამოყენების მოთხოვნებით და ამიტომ ისინი ერთმანეთისგან განსხვავდებიან ვარიანტების დიდი სხვადასხვაობით.

მრავალმანქანური ციფრული გამოთვლითი სისტემების აგების ზოგადი კონცეფცია საბორტო გამოყენების შემთხვევაში შესაძლებელია ჩამოვაყალიბოთ შემდეგნაირად:

- სტრუქტურის მრავალდონიანი იერარქიული ორგანიზაცია, რომელიც შეესაბამება გადასაწყვეტი ფუნქციური ამოცანების იერარქიულ ხასიათს;
- სტრუქტურის პროგრამირებადობა გამოთვლითი ქსელების ზედა დონეებზე ორგანიზაციის შესაძლებლობის უზრუნველყოფით და სისტემის რესურსების მიხედვით ამოცანათა დინამიკური გადანაწილებით;
- ინფორმაციული გაცვლის იერარქიული და პროგრამულ-მართვადი საშუალებების გამოყენება უნიფიცირებული არხების საფუძველზე;

- საბორტო ციფრული გამოთვლითი სისტემის როგორც ელემენტების, ისე მთელი სისტემის სტრუქტურის და კონსტრუქციის მოდულობის უზრუნველყოფა უნიფიცირებული კონსტრუქციულ-ფუნქციური მოდულების გამოყენების ხარჯზე;

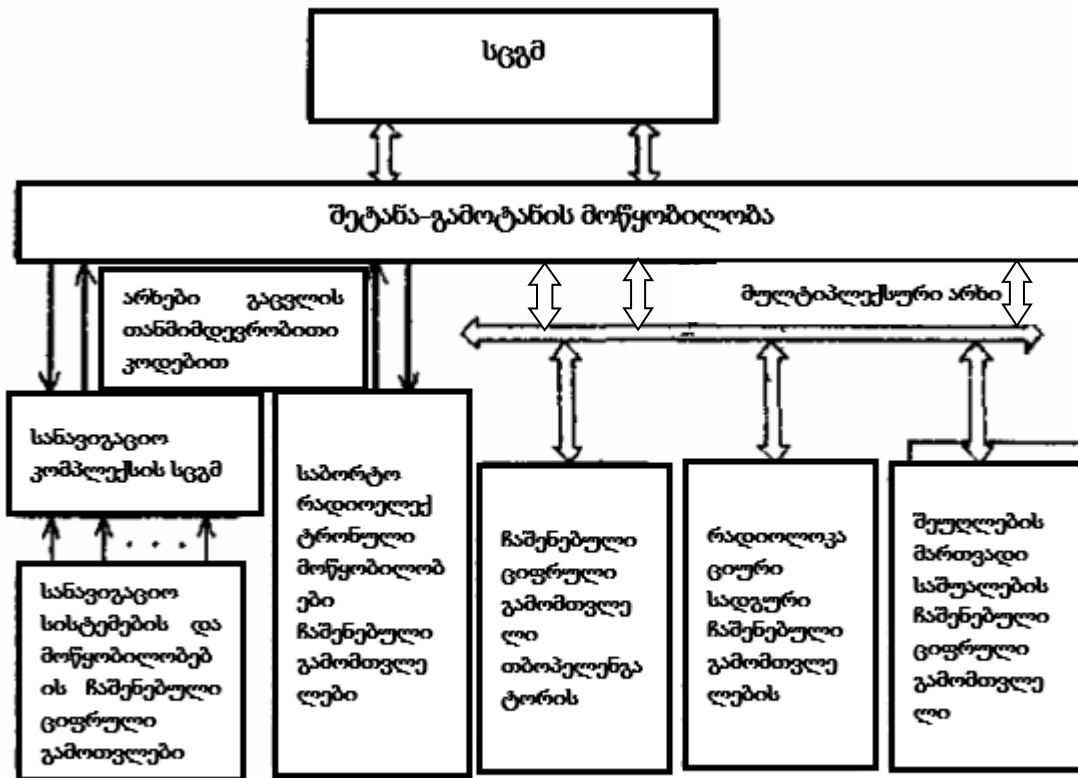
- ფუნქციური კონტროლისა და დიაგნოსტიკის განვითარებული საშუალებების არსებობა.

მრავალმანქანური საბორტო ციფრული გამოთვლითი სისტემის მრავალდონიანი იერარქიული ორგანიზაცია განისაზღვრება ინფორმაციის დამუშავების ამოცანების დონეებად დაყოფის აუცილებლობით, რომლებთაგან თითოეული ხასიათდება ინფორმაციის განზოგადების სხვადასხვა ხარისხით, რომელსაც იყენებს მოცემული დონის ამოცანის გადასაწყვეტად.

მაგალითად, საავიაციო რადიოელექტრონული კომპლექსის ამოცანები შეიძლება განაწილებულ იქნეს ოთხი იერარქიული დონის მიხედვით:

- ამოცანები, რომლებიც გადაიჭრებიან მხოლოდ ერთი გადამწოდის ინფორმაციის საფუძველზე;
- ინფორმაციის დამუშავების ამოცანები მთელი რიგი ფუნქციურად ახლო გადამწოდებისთვის;
- საერთო კომპლექსური ხასიათის ამოცანები - კომპლექსის ძირითადი ამოცანები, რომლებიც გადაწყვეტის პროცესში განაზოგადებენ ინფორმაციას, უკიდურეს შემთხვევაში ორი სხვადასხვა ფუნქციური ქვესისტემიდან;
- ფუნქციური კონტროლის, ინდიკაციის და კომპლექსის მართვის ამოცანები, რომელთა გადაწყვეტა ხდება მთელი მიწოდებული ინფორმაციის განზოგადების საფუძველზე.

რეალური საბორტო გამოთვლითი სისტემის სტრუქტურული სქემა განვიხილოთ თვითმფრინავ მიგ-31-ის მაგალითზე (ნახ.12.8).



ნახ.12.8

ყველაზე უფრო მაღალ დონეზე განლაგებულია საბორტო ციფრული გამოთვლითი მანქანა (საბორტო კომპიუტერი) A-15A. მისი დანიშნულებაა საერთო კომპლექსური ამოცანების გადაწყვეტა იმ ინფორმაციის საფუძველზე, რომლებსაც ღებულობს სისტემის ყველა დაკავშირებული კომპლექსისგან. ეს კომპიუტერი უზრუნველყოფს ასევე ფუნქციური კონტროლის, ინდიკაციის და მართვის ამოცანების გადაწყვეტას და რადიოელექტრონული კომპლექსის, თვითმფრინავის ფრენის და განადგურების საშუალებების მუშაობის მართვასაც. საბორტო ციფრული გამოთვლითი მანქანის კავშირი სისტემის გარე ელემენტებთან ხორციელდება შეტანა-გამოტანის მოწყობილობების დახმარებით.

შეტანა-გამოტანის მოწყობილობა ასრულებს შემდეგ ფუნქციებს:

- მასინქრონიზებული სიგნალების გამომუშავება საავიაციო რადიოელექტრონული კომპლექსის ყველა სისტემისათვის;
- საბორტო ციფრული გამოთვლითი მანქანა A-15A-სა და საავიაციო კომპლექსის სისტემებს შორის მონაცემების გადაცემა მათი გარდაქმნებით;

- მოთხოვნათა ფორმირება გაცვლის რეჟიმების რეალიზაციაზე.

შეტანა-გამოტანის მოწყობილობას გააჩნია გაცვლის არხები თანმიმდევრობითი და პარალელური ორობითი კოდებით, ასევე ანალოგური სიგნალების შესასვლელები და გამოსასვლელები მუდმივი და ცვლადი დენების ძაბვის და იმპულსების სახით.

საავიაციო კომპლექსის საბორტო კომპიუტერი უზრუნველყოფს ინფორმაციის მეორად დამუშავებას, რომელიც მიეწოდება სანავიგაციო კომპლექსის ელემენტებიდან. ყველაზე დაბალ - პირველ დონეზე განლაგებულია სხვადასხვა ქვესისტემების ჩაშენებული გამომთვლელები.

12.2. საავიაციო რადიოელექტრონული კომპლექსის საბორტო ციფრული გამოთვლითი სისტემების პროგრამული უზრუნველყოფა

საავიაციო რადიოელექტრონული კომპლექსის საბორტო ციფრული გამოთვლითი სისტემის პროგრამული უზრუნველყოფის ქვეშ იგულისხმება პროგრამების ერთობლიობა, რომლებიც უზრუნველყოფენ საერთო კომპლექსური იმ ამოცანების გადაწყვეტას, რომლებიც დაკისრებული აქვს მაღალი დონის კომპიუტერს.

გამოთვლითი სისტემის პროგრამული უზრუნველყოფის ფუნქციური მნიშვნელობა მდგომარეობს თვითმფრინავის მართვაში, ასევე მისი საბორტო მოწყობილობებისა და შეიარაღების მართვაში. მმართველი სისტემების პროგრამული უზრუნველყოფისთვის ნიშანდობლივია შემდეგი თავისებურებები:

- გამოთვლების შესრულება დროის რეალურ მასშტაბში;
- პროგრამებისა და გამოყენებული რიცხვითი ინფორმაციის დიდი მოცულობა, რომლებიც ერთდროულადაა არსებული საბორტო კომპიუტერში;
- პროგრამების ძლიერი ურთიერთდაკავშირებულობა მათი ჩართვის თანმიმდევრობითობის მიხედვით და გამოყენებული ცვლადი სიდიდეების მიხედვით;
- დროის დაყოფის რეჟიმში პროგრამების შესრულების მულტიპროგრამული ხასიათი და გარე მოწყობილობებთან ინფორმაციის გაცვლის არსებობა ძირითადი ფუნქციური ამოცანების გადაწყვეტის დროს;
- პროგრამების ფუნქციონირების მრავალრეჟიმულობა ერთი და იგივე პროგრამის სხვა პროგრამებთან სხვადასხვა შეხებებით მმართველი სისტემის მუშაობის რეჟიმების ცვლილებით;

- ბუფერული დამგროვებლების გამოყენება შემავალი და გამომავალი ინფორმაციის შენახვისათვის, რომლებიც თავის მხრივ დამგროვებელში მიეწოდება დროის შემთხვევით მომენტებში რამდენიმე აბონენტთან ურთიერთქმედების დროს;

- პროგრამულ-ლოგიკური ხერხების გამოყენება აპარატურის შეფერხებებისა და მტყუნებებისგან დაცვისათვის, ასევე მიწოდებული ინფორმაციის დაცვისათვის.

საბორტო კომპიუტერის პროგრამული უზრუნველყოფის შემადგენლობაში შედის პროგრამები თვითმფრინავის და შეიარაღების მართვისათვის, ასევე ნავიგაციის პროგრამები, ჩაშენებული კონტროლის პროგრამები და კავშირის პროგრამები სხვა თვითმფრინავებთან ჯგუფურად მუშაობის დროს. ეს პროგრამები წყვეტენ ძირითად ფუნქციურ ამოცანებს და ამიტომ იწოდებიან ფუნქციურ პროგრამებად.

მათ გარდა საბორტო კომპიუტერის პროგრამულ უზრუნველყოფაში შედის ასევე პროგრამები, რომლებიც ორგანიზებას უკეთებენ გამოთვლით პროცესებს და იწოდებიან მმართველ და დისპეტჩერულ პროგრამებად.

მმართველი პროგრამების დამუშავებისას გათვალისწინებული უნდა იქნეს ციფრული გამოთვლითი სისტემის შემდეგი თავისებურებები:

- ციფრული გამოთვლითი სისტემა ასრულებს მრავალ ფუნქციას და ახდენს ინფორმაციის გაცვლას დიდი რაოდენობის გარე აბონენტებთან;

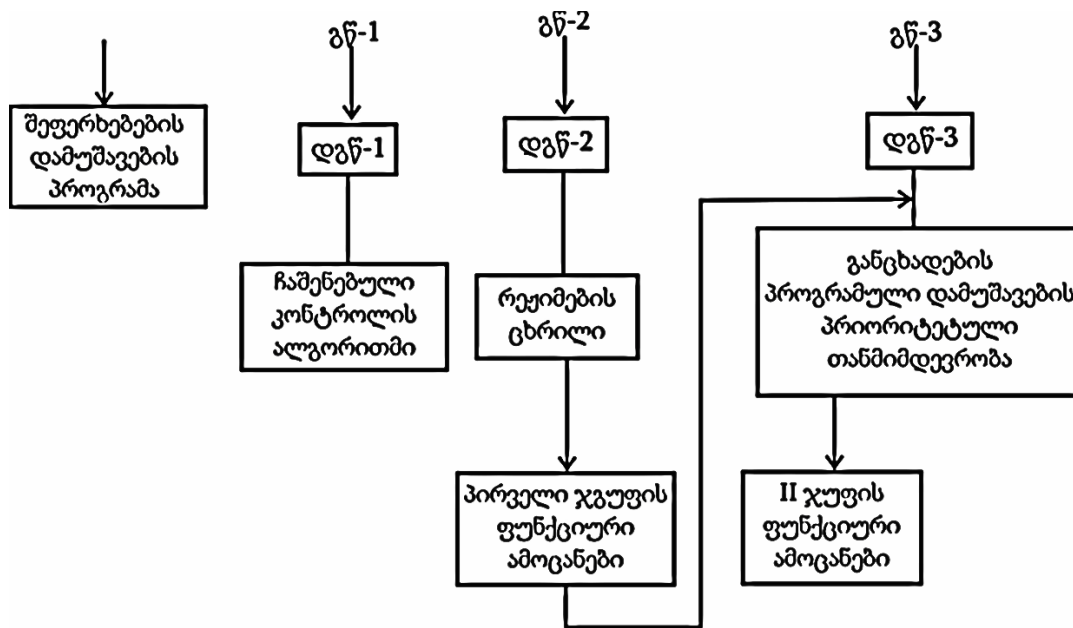
- მონაცემები აბონენტებიდან კომპიუტერში შეიძლება მიეწოდოს პერიოდულად, ან დროის შემთხვევით მომენტებში, რაც გამოწვეულია კომპლექსის სხვადასხვა სისტემების მუშაობის ასინქრონულობით, ამიტომ ცალკეული ამოცანების გადაწყვეტისათვის მოთხოვნები ციფრულ გამოთვლითი სისტემაში ფორმირდება განსაზღვრული სიხშირით ან აღიმკვრება ასინქრონულად გარე აბონენტებისგან;

- ციფრულ გამოთვლითი სისტემაში გათვალისწინებული უნდა იყოს ყველა განაცხადი თითოეული ამოცანის გადაწყვეტაზე და უნდა განისაზღვროს თითოეულ სიტუაციაში მათი რეალიზაციის ოპტიმალური თანმიმდევრობა;

- ამოცანის გადაწყვეტის ხანგრძლივობა და ინფორმაციის გაცემის ტემპი უნდა შეესაბამებოდეს მუშაობის რეჟიმებს და სამართავი ობიექტების მიმდინარე მდგომარეობას.

ამ მოთხოვნების გათვალისწინებით და საბორტო კომპიუტერის A-15A-ს წყვეტის სისტემის თავისებურებების გამო თვითმფრინავ მიგ-31-ის საბორტო

კომპიუტერში არჩეულია გამოთვლითი პროცესის შემდეგი ორგანიზაცია (ნახ.12.9). სისტემაში გამოყენებულია პირველი, მეორე და მესამე კლასის გარე წყვეტები (გწ-1, გწ-2, გწ-3).



ნახ.12.9

გარე წყვეტები (გწ. 0) გამოიყენება კომპიუტერის კონტროლისთვის და საფრენოსნო ინფორმაციის რეგისტრაციის რეჟიმის მომსახურებისთვის. ამ კლასის ამოცანები ითვლება ყველაზე უფრო პრიორიტეტულად და შეუძლიათ შეწყვიტონ ნებისმიერი სხვა ამოცანები, რომელთაც არ გააჩნიათ ნილაბი (აკრძალვა) გწ. 0-ზე.

ფუნქციური ამოცანები დანაწილებულია ორ ჯგუფად:

- რეჟიმების ცხრილის (რც) ამოცანები, რომლებიც ხისტად არიან დაკავშირებული დროით დიაგრამასთან, პერიოდულები არიან და თანმიმდევრებიან ერთმანეთს მკაცრი თანმიმდევრობითობის დაცვით;
- ამოცანები, რომელთა გადაწყვეტა ხდება განაცხადების მიხედვით, როგორც წესი, ნაკლებად არიან დაკავშირებული დროით დიაგრამასთან.

პირველი ჯგუფის ამოცანების გადაწყვეტის დრო არაა დიდი და ის ნაკლებია ტაქტის ხანგრძლივობაზე. ამოცანების გადაწყვეტის დრო, რომელთა გადაწყვეტა ხდება განაცხადების მიხედვით, როგორც წესი, ერთ ტაქტზე მეტია.

ამოცანები, რომლებიც გადაიჭრებიან განაცხადების მიხედვით, შეიძლება შეწყდნენ რეჟიმების ცხრილის ამოცანებით. აქედან გამომდინარეობს აუცილებლობა იმისა, რომ მოხდეს შეწყვეტილი ამოცანების მდგომარეობის დამახსოვრება მისი შემდგომი აღდგენისა და გადაწყვეტის გაგრძელებისთვის. ამოცანები განაცხადების მიხედვით დგებიან რიგში და მათი შესრულება ხდება განაცხადების პროგრამული დამუშავების თანმიმდევრობითი პრიორიტეტული რიგის დაცვით.

რეჟიმების ცხრილის ამოცანები ჩაირთვებიან გწ-2-ის მიხედვით, ჩაშენებული კონტროლის ამოცანები კი გწ-1-ის მიხედვით. ასეთი დანაწილება შემოტანილია სისტემის დიდი დაბრკოლებადაცულობისათვის.

ქვესისტემების მუშაობის შეთანხმებისთვის, რომელთაც, როგორც წესი, გააჩნიათ საკუთარი დროითი დიაგრამები, მთლიანობაში სისტემასთან მუშაობისათვის გამოიყენება გწ-3. ისინი ემსახურებიან სასწრაფო ამოცანების ჯგუფს და შეუძლიათ შეწყვიტონ როგორც განაცხადების, ისე რეჟიმების ცხრილის ამოცანები.

შეწყვეტებიდან თითოეულისათვის არსებობს დამუშავების შესაბამისი პროგრამა. პროგრამა - დამუშავება გარე წყვეტა 1 (დგწ-1) ორგანიზაციას უკეთებს ჩაშენებული კონტროლის პროგრამების ჩართვას 96 ჰერცი სიხშირით. პროგრამა - დამუშავება გარე წყვეტა 2 (დგწ-2) ორგანიზებას უკეთებს სისტემის მუშაობას დროის რეალურ მასშტაბში და შეიცავს რეჟიმების ცხრილის პროგრამას ასევე 96 ჰერცი სიხშირით. პროგრამების შესრულების მაქსიმალური სიხშირე განისაზღვრება იმპულსების მიდევნების პერიოდით (რადიოლოკაციური კომპლექსის სინქრონიზაციის იმპულსები), რომელიც ტოლია 10,24 მილიწამის. ეს დროითი ინტერვალი ასევე იწოდება კომპლექსის მუშაობის ელემენტარულ ტაქტად. პროგრამა - დამუშავება გარე წყვეტა (დგწ-3) ორგანიზებას უკეთებს ცალკეული ამოცანების შესრულებას მათი ჩართვის მოთხოვნათა შესაბამისად, ასევე სპეციალურ სერვისულ ხელსაწყოთან ერთად უზრუნველყოფს პროგრამების კონტროლს და გამართვას.

12.3. პროგრამების შესრულება, რომლებიც ხისტად არიან დაკავშირებული დროით დიაგრამასთან

კერძო (ფუნქციური) პროგრამები, რომლებიც ხისტად არიან დაკავშირებული დროით დიაგრამასთან შესრულებაზე გაიშვება რეჟიმების ცხრილის პროგრამით. რეჟიმების ცხრილი წარმოადგენს საერთო დისპეტჩერის ნაწილს. ის უზრუნველყოფს პროგრამების თანმიმდევრობით მიერთებას მოცემული დროითი დიაგრამის და რადიოელექტრონული კომპლექსების მუშაობის ლოგიკის შესაბამისად, ახორციელებს განაცხადების გენერაციას სიხშირულ და ასინქრონულ ამოცანებზე, ასევე ორგანიზაციას უკეთებს კომპლექსის მუშაობის მრავალნაირ რეჟიმებს.

ყველა სიხშირული პროგრამები, რომლებიც შედის რეჟიმების ცხრილში, იყოფა რამდენიმე კლასად (დონედ). თითოეული დონე აერთიანებს ერთნაირი ფუნქციური დანიშნულების მქონე პროგრამებს, რა თქმა უნდა რადიოლოკაციური კომპლექსის სხვადასხვა რეჟიმების შესაბამისად. ამრიგად, ნებისმიერი კონკრეტული რეჟიმი გულისხმობს თითოეულ დონეზე ერთი პროგრამის ჩართვას და შესრულებას. თითოეულ დონეზე ამოცანის ამორჩევა ხორციელდება კომპლექსის მუშაობის რეჟიმის შესაბამისად და მოიცემა თითოეული დონის გასაღებების ფორმირებით. გასაღები – ეს არის ოპერატიული დამხსომებელი მოწყობილობის უჯრედი, რომელშიც მოთავსებულია კონკრეტული პროგრამის საწყისი მისამართი. გასაღებები ფორმირდებიან როგორც უშუალოდ რეჟიმების ცხრილში, ასევე სხვა პროგრამებშიც. თავდაპირველად გასაღებებს აფორმირებს პროგრამა გაშვება, ასეთი სტრუქტურა უზრუნველყოფს რეჟიმების გადართვის სიმარტივეს უჯრედების შიგთავსის შეცვლის გზით. რეჟიმების ცხრილის ასეთი აგების სხვა ღირსებას წარმოადგენს ის, რომ ასეთ შემთხვევაში შესაძლებელია ახალი რეჟიმების მიერთების შესაძლებლობა თითოეულ დონეზე პროგრამების დამატების გზით, ისე, რომ არ იქნეს შეცვლილი პროგრამული უზრუნველყოფის ზოგადი სტრუქტურა.

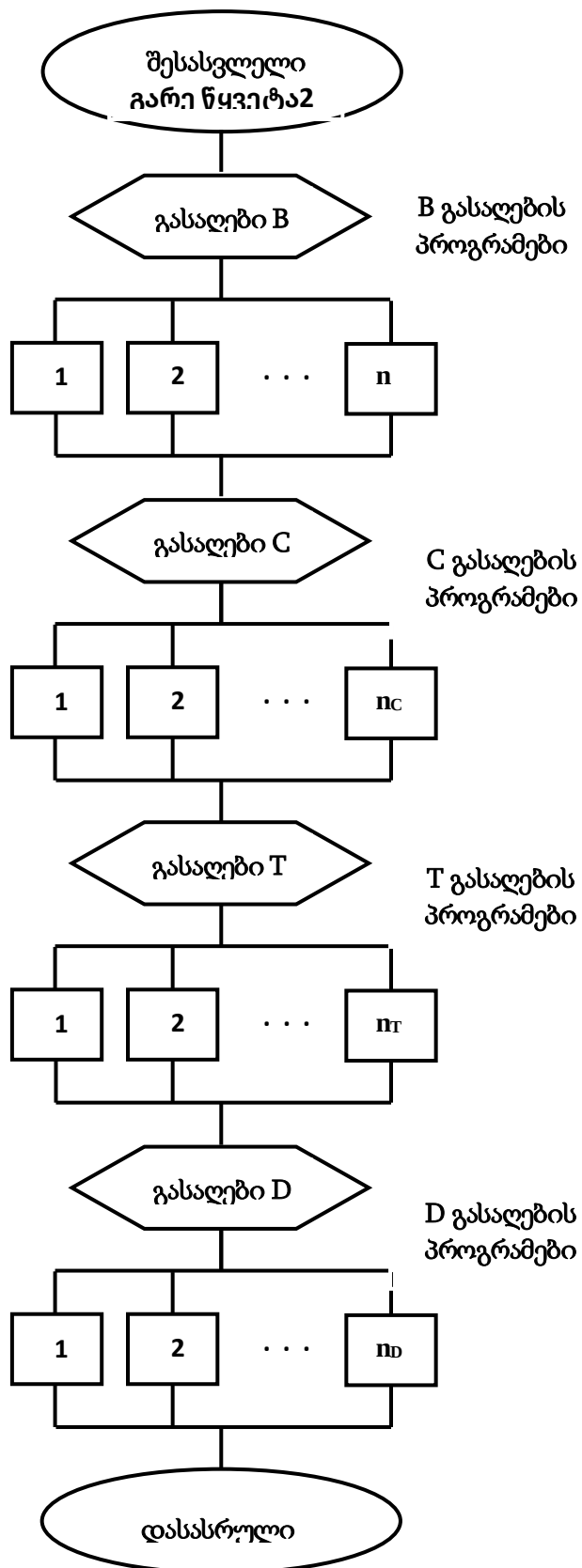
პროგრამის მიმდინარეობის კონტროლის განხორციელებისთვის და ამ პროგრამის უწყისიერობის (შეფერხების) შემთხვევაში აღდგენისათვის რეჟიმის ცხრილის თითოეული დონე აღჭურვილია ეგრეთწოდებული საკონტროლო რომბით. რაც გულისხმობს, რომ დონის დასაწყისში ხდება პროგრამის მდგომარეობის სიტყვის დამახსოვრება, რომლის ერთ-ერთ ელემენტს წარმოადგენს დონის საწყისი მისამართი.

დონის ბოლოს მოწმდება საწყისი მისამართი და, თუ სამოწმებელი მისამართი არ ემთხვევა პროგრამის მდგომარეობის სიტყვაში (პმს) ჩაწერილ მისამართს, მაშინ ეს მიუთითებს იმაზე, რომ მოხდა შეფერხება და აუცილებელია აღდგენილ იქნეს პროგრამის მიმდინარეობის სისწორე. პმს-ის ჩაწერის და შემოწმების ოპერაციები ალგორითმის სქემაზე გამოსახულია რომბებით, ამიტომ ისინი ატარებენ სწორედ საკონტროლო რომბების სახელწოდებას.

B დონის პროგრამები (ნახ.12.10) განკუთვნილია საბორტო რადიოლოკაციური სადგურიდან მიღებული ინფორმაციის დამუშავებისათვის სხვასდასხვა რეჟიმებში. თითოეული ინტერვალისათვის კონკრეტული რეჟიმი მოიცემა ინტერვალის კოდის პროგრამის საშუალებით, ის ასევე უზრუნველყოფს B დონის შესაბამისი შტოს დანიშვნას. B დონის ჩართვის სიხშირე, ისე როგორც რეჟიმების ცხრილების მთელი პროგრამისა შეადგენს 96 ჰერცს, ხოლო ამ დონის თითოეული კონკრეტული შტო უერთდება მას თავისი განსაზღვრული სიხშირით. თუ ტაქტში არ არის მოცემული დონის ამოცანები, მაშინ B-ს გასაღებში ჩაიწერება უქმი შტოს მისამართი, რომელიც მართვას გადასცემს შემდგომ დონეს.

რეჟიმების ცხრილის მეორე დონე იწოდება როგორც „ ელემენტარული ტაქტების მრიცხველის დეშიფრატორი” და სქემაზე (ნახ.12.10) აღნიშნულია როგორც C დონე.

ამ დონის საერთო შტოში 96 ჰერცი სიხშირით მუშაობს ინფორმაციის გაცემის პროგრამა, რომელიც ინფორმაციას აგზავნის რეგისტრაციის სისტემაში და მიმოხილვით ინდიკატორზე.



ნახ.12.10

T დონე წარმოადგენს განმსაზღვრელს რეჟიმების ცხრილისთვის. პროგრამის სახელწოდება განპირობებულია სწორედ მისით. ამ დონეზე შეკრებილია პროგრამები

სახელწოდებით **ინტერვალის კოდი**, რომლებთან თითოეული განსაზღვრავს შესაბამისი ინტერვალის შინაარსს. სახელდობრ აფორმირებს ინტერვალის კოდს, გაითვლის და გასცემს გაცვლით უჯრედებში ამ ინტერვალისათვის აუცილებელ ინფორმაციას, ახდენს **B** და **D** გასაღებების ფორმირებისათვის დამუშავების შტოების დანიშვნას. კომპლექსის მუშაობის თითოეულ რეჟიმს შეესაბამება ინტერვალის თავისი განსაზღვრული თანმიმდევრობა, რომელიც მოიცემა ინტერვალის კოდის პროგრამის საწყისი მისამართების ცხრილებით. ეს ცხრილები შეიცავს 25 მისამართს, რომელიც იძლევა ინტერვალის კოდის პროგრამის თანმიმდევრობას სრული ციკლისთვის. ცხრილების რაოდენობა განისაზღვრება კომპლექსის მუშაობის რეჟიმების რაოდენობით და შესაძლებელია მისი გაზრდა, თუ შემოტანილ იქნება მუშაობის ახალი რეჟიმები.

T დონიდან ხორციელდება **D** დონეზე მართვის გადაცემა. ამ დონეზე შეკრებილია პროგრამები, რომელთა ფუნქციური დანიშნულებაა – კუთხური პოზიციების გაანგარიშება და მათი გაცემა საბორტო რადიოლოკაციურ სადგურში. **D** დონე ბოლოა რეჟიმების ცხრილში, რომლის შესრულების შემდეგ მართვის გადაცემა ხდება განაცხადების დამუშავების პროგრამაში.

12.4. პროგრამების შესრულება ასინქრონული განაცხადების მიხედვით

საბორტო ციფრული გამოთვლითი სისტემების პროგრამული უზრუნველყოფა შეიცავს პროგრამათა მთელ რიგს, რომელთა შესრულების დრო მეტია ტაქტის ხანგრძლივობაზე ტაქტში უფრო პრიორიტეტული ამოცანების გადაწყვეტის აუცილებლობის გათვალისწინებით. გარდა ამისა, არსებობს ასევე ორი ან უფრო მეტი პროგრამის შესრულებისას დროში ზედდების შესაძლებლობა, რომელიც განპირობებულია მათი პრიორიტეტულობით.

ამ ამოცანების გადაწყვეტისათვის მათი პრიორიტეტულობისა და ცენტრალური ციფრული გამოთვლითი მანქანის და გარე მოწყობილობების მუშაობის სინქრონულობის შენარჩუნების პირობებში, ასეთი ამოცანების შესაბამისი პროგრამების შესრულება ორგანიზებულია განაცხადების მიხედვით, განაცხადების წყვეტის ნების დართვით. მმართველი პროგრამა, რომელიც ემსახურება ფუნქციური პროგრამების მოცემულ

ჯგუფს, წარმოადგენს შედგენილს, დისპეტჩერის ნაწილს და იწოდება განაცხადების დამუშავების პროგრამად (გდპ).

ამ პროგრამის პრინციპი მდგომარეობს შემდეგში. ყველა ამოცანა, რომელიც უნდა გადაიჭრას განაცხადების მიხედვით, განაწილებულია პრიორიტეტის მიხედვით. თითოეულ კერძო პროგრამას შეესაბამება განაცხადების რეგისტრის განსაზღვრული თანრიგი. განაცხადების რეგისტრი გრ1 – გრ2 - ეს არის ოპერატიული დამხსომეხელი მოწყობილობის ორმაგი სიტყვა. განაცხადების პრიორიტეტულობა მცირდება თანრიგის ნომრის ზრდასთან ერთად. სიტუაცია, რომელიც შეესაბამება კერძო პროგრამიდან განაცხადს, ხასიათდება 1-ის არსებობით განაცხადების რეგისტრში მისთვის განკუთვნილ თანრიგში. განაცხადების რეგისტრის გარდა ოპერატიულ დამხსომეხელ მოწყობილობაში ორგანიზებულია შეწყვეტილი განაცხადების რეგისტრის (შგრ1, შგრ2) არსებობა, ხოლო ხანგრძლივ დროით მეხსიერებაში ინფორმაციის შეცვლის შესაძლებლობით ორგანიზებულია შეუსრულებელი განაცხადების რეგისტრის (შეგრ1, შეგრ2). ორივე რეგისტრს გააჩნია თანრიგების განაწილება, რომელიც ანალოგიურია გრ1 – გრ2-ის. კერძო წყვეტების განაცხადები შესრულებაზე ისმება რეჟიმების ცხრილის პროგრამის წინაშე ან შემოდის მოთხოვნები გარე აბონენტებისგან.

განაცხადების დამუშავების პროგრამა (გდპ) პირველ რიგში უზრუნველყოფს იმ განაცხადის მომსახურებას, რომელსაც გააჩნია ყველაზე უმცირესი ნომერი თანრიგებში დაყენებული ყველა ერთიანების სიმრავლიდან. ამიტომ შესაძლებელია ერთი კერძო პროგრამის წყვეტა უფრო პრიორიტეტული პროგრამით. ამ შემთხვევაში განაცხადების დამუშავების პროგრამა დამუშავებზე აყენებს უფრო პრიორიტეტულ პროგრამას, რომელიც იმახსოვრებს ყველა იმ სიდიდეს, რომელიც აუცილებელია შეწყვეტილი პროგრამის შესრულების გაგრძელებისთვის. დამახსოვრება ხდება ოპერატიული დამხსომეხელი მოწყობილობის განსაზღვრულ უბანზე, რომელიც მაღაზიის ჯიბედ იწოდება. სულ არსებობს 5 ჯიბე, რომელთაგან შესაძლებელია შენახულ იქნეს ხუთი შეწყვეტილი კერძო პროგრამა. მიმდინარე კერძო პროგრამის შესრულების შემდეგ და უფრო პრიორიტეტული განაცხადების არარსებობის შემთხვევაში ჯიბეებიდან ამოირჩევა შეწყვეტილი პროგრამები რიგის მიხედვით და ხდება მათი შესრულება.

მაღაზიის თითოეულ ჯიბეში არის 11 უჯრედი შემდეგი სიტყვებით:

ბაზის ნომერი – მასში დაფიქსირებულია მოცემული უჯრედის საკუთარი მისამართი მაღაზიის ჯიბეზე აწყოების კონტროლისათვის;

პმს1 * პმს2 მიმდ.– პროგრამის მდგომარეობის სიტყვა წყვეტის მომენტში;

პმს1 * პმს2 ბლ.– მწვეტარა პროგრამის საწყისი ბლოკის პროგრამის მდგომარეობის სიტყვა;

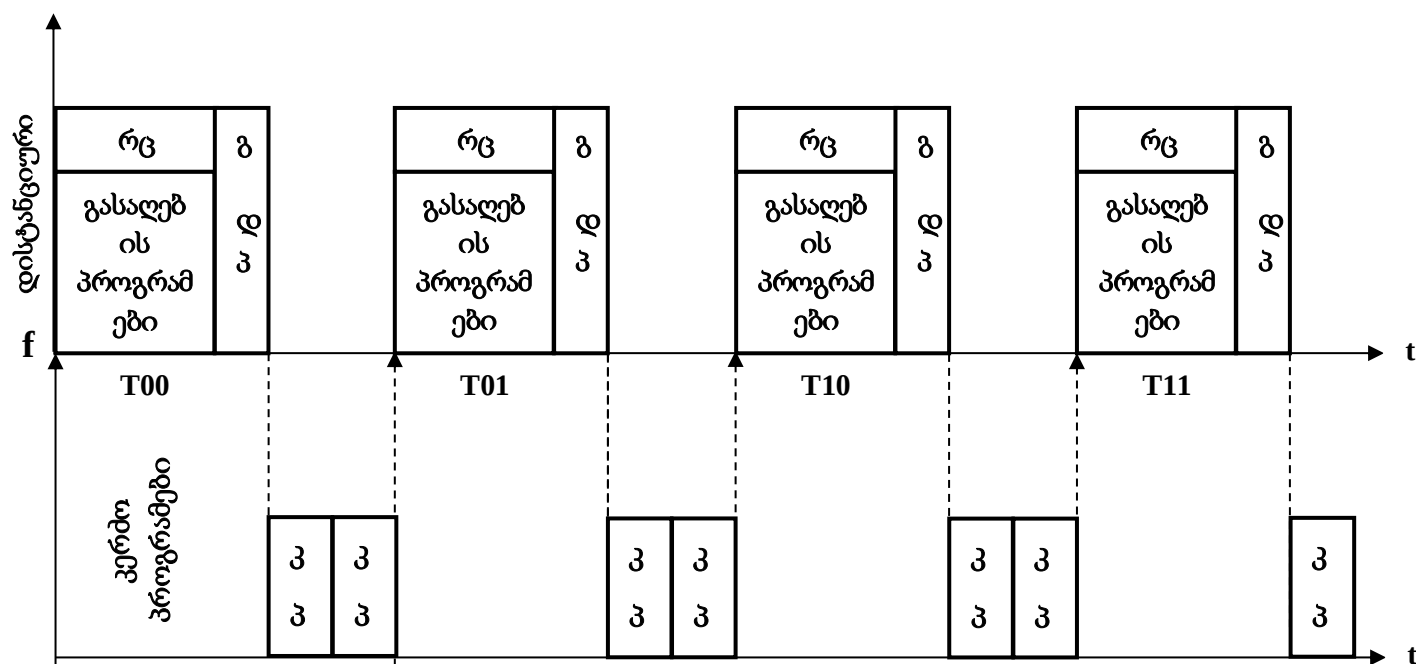
რა1 * რა2, რი1 * რი2– ამჯამავის რეგისტრებისა და ინდექსური რეგისტრების შიგთავსი;

დმრ1 * დმრ2 - დაციკვლის მრიცხველი - დროის მომენტი, რომელმდეც ამოცანა უნდა იყოს გადაწყვეტილი. ის განისაზღვრება ახალი ამოცანის ამორჩევის დროს, როგორც დროის მიმდინარე მთვლელი პლიუს დრო, რომელიც გამოყოფილია მოცემული ამოცანის გადაწყვეტისათვის. ამოცანის გადაწყვეტის დრო განისაზღვრება სახელდობრ ამ მოცემული ამოცანის გადაწყვეტისთვის საჭირო დროის შეჯამებით იმ უფრო პრიორიტეტული ამოცანების გადაწყვეტის დროსთან, რომლებმაც გამოიწვიეს მისი შეწყვეტა.

დაციკვლის მრიცხველის უმცროსი თანრიგის ფასი ტოლია ერთი ტაქტის (10,24 მწმ).

მაღაზიის ჯიბეზე აწყოებისათვის არსებობს მაღაზიის ინდექსური რეგისტრი (მირ), რომელშიც ყოველთვის ინახება მაღაზიის უკანასკნელი დაკავებული ჯიბის მისამართი. იმ შემთხვევისათვის, როცა წყვეტაზე განაცხადები არ არის, მაღაზიის ინდექსურ რეგისტრში იგზავნება მაღაზიის ნულოვანი ჯიბის მისამართი (მირ0), რომელიც შედგება ერთი უჯრედისგან – ჯიბის ბაზისგან.

მთლიანობაში პროგრამის შესრულების დროით დიაგრამას გააჩნია სახე, რომელიც წარმოდგენილია ნახ.12.11-ზე.



ნახ.12.11. პროგრამის შესრულების დროითი დიაგრამა

რეგ - რეგისტრი

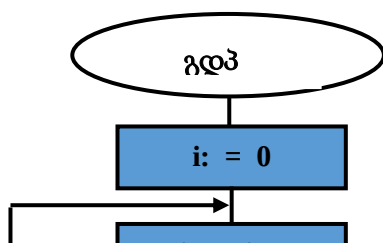
გდპ - განაცხადების დამუშავების პროგრამა

რე - რეჟიმების ცხრილი

კპ - კერძო პროგრამები

რეჟიმების ცხრილის პროგრამები დაკავშირებულია გარე წყვეტებთან და სრულდება ერთ ციკლის განმავლობაში 96 ჰერცი სიხშირით. კერძო პროგრამები, რომელთა შესრულება მოითხოვს დიდ დროს, ჩაიტვირთება შესრულებაზე განაცხადების დამუშავების პროგრამით. ისინი შეიძლება შეწყდეს, როგორც ერთმანეთით, ისე რეჟიმების ცხრილის პროგრამით. ნახ.12.12-ზე ნაჩვენებია განაცხადების დამუშავების პროგრამის ალგორითმის სქემა. თვითონ განაცხადების დამუშავების პროგრამის წყვეტა შეუძლებელია.

განაცხადების დამუშავების პროგრამაში (გდპ) შესვლა ხორციელდება რეჟიმების ცხრილის პროგრამიდან. შესვლის შემდეგ ხორციელდება ყველა წყვეტის შენიღბვა, გარდა მოწყობილობის შეფერხების წყვეტისა და განაცხადების რეგისტრიდან და შეწყვეტილი განაცხადების რეგისტრიდან ამოიღება უფროსი თანრიგების მნიშვნელობები.



არსებული თანრიგებიდან თითოეული თანრიგისათვის შესაძლებელია ოთხი სხვადასხვა სიტუაცია, რომელიც განისაზღვრება a და b ბიტების შეხამებით.

იმ შემთხვევაში, როცა ბიტებიდან $a=1$ და $b=0$, ეს იმის მაჩვენებელია, რომ ფორმირებულია ახალი განაცხადი, ხოლო ამ დონის შეწყვეტილი განაცხადები არ არის და ხორციელდება ახალი კერძო პროგრამის შესრულება. შესრულება იწყება გარკვევით არის თუ არა ეს განაცხადი შესრულებული განაცხადების რეგისტრში, რისთვისაც ხდება შეუსრულებელი განაცხადების რეგისტრის (შგრ) შემოწმება. თუ განაცხადი არის შეუსრულებული განაცხადების რეგისტრში, მაშინ აუცილებელია ის შეჩერდეს შესრულების გარეშე და უნდა მოხდეს გადასვლა ახალი განაცხადის ამორჩევაზე, ე.ი. განაცხადების დამუშავების პროგრამის დასაწყისზე. თუ შეუსრულებული განაცხადების რეგისტრის i -ური თანრიგის მდგომარეობა ტოლია 0-ის, მაშინ მუდმივი დამხსომეხელი მოწყობილობიდან ჩაიტვირთება პროგრამის მდგომარეობის სიტყვა, რომელიც განაცხადის შესაბამისი იქნება და მოხდება მისი შესრულება. შესრულების პროცესში შესაძლებელია კერძო პროგრამის შეწყვეტა უფრო მაღალი პრიორიტეტის მქონე

პროგრამებით ან მმართველი პროგრამით (რეჟიმების ცხრილის პროგრამით). ამ შემთხვევაში შეწყვეტილი განაცხადების რეგისტრში i - ურ თანრიგში ჩაიწერება ერთიანი და ხდება განაცხადების დამუშავების პროგრამიდან გამოსვლა და რეჟიმების ცხრილში შესვლა. კერძო პროგრამის ნორმალური დამთავრებისას განაცხადების პროგრამული დამუშავების პროგრამა ანულებს განაცხადების i -ურ რეგისტრს, შემდეგ უზრუნველყოფს i - ის გადიდებას ერთით და გადადის შემდგომ (დაბალ) პრიორიტეტის განაცხადებზე.

როცა $a=0$ და $b=1$, ბიტების ასეთი კომბინაციისას არ ხდება ახალი გენერირება, ხოლო ძველი შეწყვეტილია და ხდება შეწყვეტილი პროგრამის მდგომარეობის სიტყვის აღდგენა მაღაზიის შესაბამისი ჯიბის შიგთავსის მიხედვით და შემდეგ შესრულება.

როცა $a=0$ და $b=0$, ასეთ შემთხვევაში ახალი განაცხადი არაა გენერირებული, ხოლო ძველი განაცხადი შესრულებულია და განაცხადების დამუშავების პროგრამა გადადის ბიტების შემდგომი წყვილის შემოწმებაზე განაცხადების რეგისტრის გრ $i+1$ და შეწყვეტილი განაცხადების რეგისტრის შგრ $i+1$, რომლებიც შეესაბამებიან უფრო ნაკლები პრიორიტეტის განაცხადებს. ყველა განაცხადის შესრულების შემდეგ ხდება მიმდინარე თვითშემოწმების ტესტების გაშვება.

როცა $a=1$ და $b=1$ - ეს იმის მაჩვენებელია, რომ ახალი განაცხადი გენერირებულია, რის გამოც წარმოიშობა კონფლიქტური სიტუაცია. ასეთ შემთხვევაში ახალი განაცხადი ნულდება და გრძელდება ძველი განაცხადის შესრულება – პროგრამის მდგომარეობის სიტყვის აღდგენა. ამრიგად, მოცემული განაცხადის თვლის სიხშირე მცირდება. თუ პროგრამის შესრულების დრო გადააჭარბებს დადგენილ დროს, მაშინ განაცხადი ჩამოიყრება (განუღდება) და ეს ჩამოყრა ფიქსირდება მტყუნებათა მრიცხველში. თუ ადგილი აქვს აპარატულ შეფერხებას, ის ასევე ფიქსირდება მტყუნებების მრიცხველში. ასეთ შემთხვევაში განაცხადების და შეწყვეტილი განაცხადების რეგისტრები განუღდებიან და რჩება მხოლოდ განაცხადი კონტროლზე, ხოლო მაღაზიის ინდექსის რეგისტრში (მირ) იგზავნება მაღაზიის ნულოვანი ჯიბის მისამართი.

“ბაგეტის” ტიპის კომპიუტერები და მათი გამოყენება

13.1. აპარატულ -პროგრამული გამოთვლითი პლატფორმის „ბაგეტის“ კონცეფცია და ელემენტური ბაზა

თანამედროვე ინფორმაციული ტექნოლოგიების ერთ-ერთ ძირითად განმსაზღვრელ ასპექტს წარმოადგენს ღია სისტემების ტექნოლოგია, რომელიც უზრუნველყოფს სისტემების კომპონენტების ურთიერთქმედებას, როგორც პროგრამულ ისე აპარატულ დონეზე. ასეთი მიდგომა განსაზღვრავს ყველა დონის გამოთვლითი სისტემების გამოყენების ეფექტურობას.

ღია სისტემების რეალიზაციის უზრუნველყოფის საფუძველს წარმოადგენს ღია ზოგადდაშვებადი სტანდარტების ერთობლიობა, რომელთა დახმარებით უნიფიცირდება აპარატურის და პროგრამული გარემოს ყველა კომპონენტის ურთიერთქმედება, ისეთებისა როგორიცაა: დაპროგრამების ენები, შეტანა-გამოტანის საშუალებები, გრაფიკული ინტერფეისები, მონაცემთა ბაზების მართვის სისტემები ქსელებში და ა.შ. მრავალი ეროვნული და საერთაშორისო ორგანიზაციების მიერ განსაზღვრული იქნა სტანდარტების ნაკრები, რომლებიც ითვალისწინებენ ღია სისტემების გამოყენების სხვადასხვა ასპექტებს.

ახალი მიდგომა მდგომარეობს იმაში, რომ შესაქმნელი სისტემის დამუშავებისას საჭიროა ფართოდ იქნეს გამოყენებული ღია ტიპის უკვე მზა და ადრე ფართოდ აპრობირებული და სტანდარტიზებული აპარატული და პროგრამული კომპიუტერული ტექნოლოგიები. ეს ეგრეთწოდებული COTS ტექნოლოგიები (Commercial OFF-The-Shelf) წარმოადგენენ ტექნოლოგიებს, რომლებიც მზად არიან გამოყენებისათვის.

ისინი იძლევიან საშუალებას გამოყენებულ იქნეს მზა ყიდვადი აპარატული და პროგრამული კომპონენტის დიდი პარკი, რომელიც საჭიროების შემთხვევაში შესაძლებლობას იძლევა დამუშავებულ და წარმოებულ იქნეს საკუთარი ორიგინალური მოდულები და პროგრამული უზრუნველყოფა საკუთარი სისტემის დამუშავებისათვის მინიმალურ ვადებში.

COTS - ტექნოლოგია-ეს არის ტექნოლოგია, რომლის ნორმატიული ბაზა ვითარდება და მხარდაჭერილია როგორც საერთაშორისო (IEC/MЭК, ISO) ისე ეროვნული (ANSI, DIN, IEEE, ГОСТ) სტანდარტიზაციის ორგანიზაციებისგან.

ქვემოთ განხილვადი აპარატულ-პროგრამული პლატფორმა „ბაგეტი“ წარმოადგენს ღია სისტემას. პლატფორმის ელემენტური ბაზას ქმნის ზედიდის ინტეგრალური სქემების ნაკრები, კერძოდ მიკროპროცესორები მონაცემებისა და სიგნალების დამუშავებისათვის, სისტემური და ინტერფეისების კონტროლერები.

პლატფორმის აპარატული და პროგრამული მდგენელები დაფუძნებულია „ბაგეტის“ სტანდარტების პროფილზე, რომელიც თავის მხრივ იყენებს აპარატურისა და პროგრამული უზრუნველყოფის სტანდარტიზაციის მსოფლიო გამოცდილებას. „ბაგეტის“ სტანდარტების პროფილი მოიცავს მექანიკურ კონსტრუქციებს, სალტეებს და ინტერფეისებს („ევრომექანიკა 6U, VME, PCI და სხვა), ხოლო პროგრამული უზრუნველყოფის ნაწილში ბაზირებულია დაპროგრამების ენა C-ს სართაშორისო სტანდარტებზე, ოპერაციული სისტემის POSIX მობილური ინტერფეისის, ქსელური და გრაფიკული ინტერფეისების TCP/IP და Window-ის საერთაშორისო სტანდარტებზე.

საბოროტო ციფრულ გამოთვლით სისტემებში გამოყენებული „ბაგეტის“ ტიპის კომპიუტერები იყოფა სამ კლასად:

- ერთი და მრავალპროცესორული კომპიუტერები, რომლებიც განხორციელებულია „ევრომექანიკა“ 6U კონსტრუქტივში სისტემწარმომქმნელი სალტით VME (VERSA Module Eurocard (მოდულური სალტეები VERSA, Eurocard-ის მექანიკურ სტანდარტში));
- ერთი და მრავალპროცესორული კომპიუტერები კონსტრუქტივში „ევრომექანიკა“ 6U სისტემწარმომქმნელი სალტით Compact PCI;
- მცირეგაბარიტიანი ჩაშენებული ერთპროცესორიანი კომპიუტერები კონსტრუქტივში PC/104-Plus PCI Only PCI სისტემწარმომქმნელი სალტით PCI.

„ბაგეტის“ ტიპის ნებისმიერი კომპიუტერი თავის მხრივ წარმოადგენს ერთმანეთთან დაკავშირებულ მოდულების ნაკრებს PCI და VME სალტეების საშუალებით. ზოგიერთ მათგანზე შესაძლებელია დაყენებულ იქნეს მეზონინური მოდულები.

უნდა ხაზგასმით ავღნიშნოთ, რომ ყველა სამივე კლასისთვის საერთო მახასიათებელს წარმოადგენს:

- სტანდარტიზაცია და უნიფიკაცია, რომელიც საშუალებას იძლევა ნაკეთობაში შენარჩუნებულ იქნეს წინა თაობის ნაკეთობების მემკვიდრეობითობა;
- მოდულური კონსტრუქცია პროცესორული და ინტერფეისული მოდულების სტანდარტული კონსტრუქტივების საფუძველზე;
- სტანდარტული სისტემწარმომქმნელი VME და PCI სალტების გამოყენება;
- სტანდარტული კომუნიკაციური და პერიფერიული ინტერფეისების გამოყენება;
- კომპიუტერის ელექტრონული მოდულების შიგა არქიტექტურის და ძირითადი ზედიდი ინტეგრალური სქემების - მიკროპროცესორების, სისტემური და ინტერფეისული კონტროლერების უნიფიკაცია;
- უნარი იფუნქციონიროს საავიაციო ობიექტებში ხისტ პირობებში გამართული მუშაობის უზრუნველყოფით;

„ბაგეტის“ ტიპის კომპიუტერების სპეციფიურ განსაკუთრებულობას წარმოადგენს ის, რომ ისინი რეალიზებულია ზედიდი ინტეგრალური სქემების უნიფიცირებული ნაკრების საფუძველზე.

ამ ტიპის ნებისმიერი კომპიუტერის მინიმალურ კონსტრუქციულ და ფუნქციურ ერთეულს წარმოადგენს მოდული სტანდარტებში, Compact PCI ან PC VME /104 Plus Only (მცირეგაბარიტიან „ბაგეტის“ ტიპის კომპიუტერებისათვის) ან მეზონინური მოდული ფორმატში PMC. კომპიუტერის მოდულების ნაკრები VME, Compact PCI სტანდარტში თავიანთი რაოდენობიდან გამომდინარე განლაგდებიან „ევრომექანიკა“ 6U კონსტრუქტივში ერთ-ერთ ტიპურ კრეიტში. სტანდარტში PC/104- Plus „ბაგეტის“ ტიპის კომპიუტერებისათვის გათვალისწინებულია მხოლოდ უკორპუსო შესრულება. შიგამოდულურ სისტემურ სალტეს, რომელიც უზრუნველყოფს მიკროპროცესორის კავშირს ნაბეჭდ ფირფიტაზე დაყენებულ ფუნქციურ მოწყობილობებთან, ასევე გარე VME მაგისტრალზე გასასვლელად გამოიყენება PCI სალტე. ეს გარემოება ნაწილობრივ უზრუნველყოფს შესაძლებლობას სტანდარტებში VME და Compact PCI მოხდეს მოდულების ფუნქციების გაფართოება PMC კონსტრუქტივში ერთი ან ორი მეზონინური მოდულის დაყენების გზით.

ყველა მოდული შეიძლება დაიყოს ორ ძირითად ჯგუფად:

- პროცესორული მოდულები;

- დამატებითი ინტერფეისების მოდულები.

თითოეული მოდული და მეზონინური მოდული ტიპისგან დამოკიდებულებიდან გამომდინარე შეიცავს სტანდარტული გარე ინტერფეისების განსაზღვრულ ნაკრებს: Ethernet, SCSI, IDE, IEE-1284, RS-232.

„ბაგეტის“ ტიპის კომპიუტერებიდან ნებისმიერი „ევრომექანიკა“ 64 კონსტრუქტივში სალტეებით VME ან Compact PCI უკიდურეს შემთხვევაში შეიცავს ერთ პროცესორულ მოდულს RISC არქიტექტურით. ეს მოდულები ერთმანეთისგან განსხვავდება დანიშნულებით და ტექნიკური მაჩვენებლებით. პროცესორულ მოდულებს გააჩნიათ 1-2 ჩასასმელი ადგილი გაფართოების მოდულების დაყენებისათვის. ამ მოდულს კი წარმოადგენს მეზონინების მოდულები PMC კონსტრუქტივში PMC IEEE P1386.1 (PMC- PCI mezzanine card), რომელიც იყენებს PCI ინტერფეისს. მეზონინები შეიძლება ასევე დაყენებულ იქნეს ინტერფეისულ მოდულებზე Compact PCI.

„ბაგეტის“ ოჯახის ტიპის კომპიუტერებში VME და PCI სალტეების და გარე ინტერფეისების (Ethernet, SCSI და სხვ.) გამოყენებამ პრაქტიკულად სრულად განსაზღვრა ფუნქციები და მათი რეალიზაციისათვის საჭირო ელემენტური ბაზა. „ბაგეტის“ ტიპის კომპიუტერებში გამოყენებულია ზედიდე ინტეგრალური სქემების შემდეგი ძირითადი ჯგუფები:

- მიკროპროცესორები, რომლებიც განსაზღვრავენ შექმნადი გამოთვლითი საშუალებების საბაზო პლატფორმებს: პერსპექტიული მაღალმწარმოებლური კომპიუტერებისთვის, მმართველი კომპიუტერებისათვის და გამოთვლითი კომპლექსებისთვის, სიგნალების დამუშავების პროგრამირებადი პროცესორებისთვის;
- სისტემური კონტროლერები, რომლებიც უზრუნველყოფენ მიკროპროცესორის (მიკროპროცესორების) ინტერფეისს სხვადასხვა ტიპის ოპერატიულ დამხსომებელ მოწყობილობებთან, Flash - მეხსიერებასთან, მუდმივ დამხსომებელ მოწყობილობასთან და შიგამოდულურ სალტესთან PCI;
- სალტეების კონტროლერები, რომლებიც უზრუნველყოფენ PCI - PCI, PCI - VME და PCI-Slave ინტერფეისებს;

- გარე ინტერფეისების კონტროლერები PCI სალტეზე გასვლით: Ethernet, SCSI, IEEE-1284, RS-232 და სხვ.;

- გრაფიკული კონტროლერები PCI ინტერფეისით.

ელექტრონულ მოდულებს VME სალტით, როგორც წესი, გააჩნიათ ორი შესრულება.

პირველი შესრულების შემთხვევაში მოდულების მუშაობა გათვალისწინებულია კომპიუტერის კორპუსებში იძულებითი საჰაერო გაგრილებით, მოდულის ნაბეჭდი ფირფიტა ჩვეულებრივ მრავალშრიანია. მოდულების სისტემურ გასართებზე გამოყვანილია მხოლოდ სალტე VME. ყველა გარე ინტერფეისი გამოყვანილია გასართებზე, რომლებიც დაყენებულია მოდულების წინა თამასებზე. წინა თამასები აღჭურვილია ექსტრაქტორებით, რომლებიც იძლევა მოდულების კორპუსებიდან ამოღების საშუალებას.

„მძიმე“ შესრულების მოდულები (მეორე შესრულება) გამიზნულია გერმეტულ კორპუსებში მუშაობისთვის შიგა ვენტილატორების გარეშე. „მძიმე“ მოდულში ორგანიზებულია ელექტრონული კომპონენტებიდან სითბოს კონდუქციური მოცილება. მოდულის ნაბეჭდი ფირფიტა წარმოადგენს მრავალშრიან სპეციალურ ნაბეჭდ ფირფიტას დამატებითი მეტალური სითბოგამტარი შრეებით. სითბოგამტარ შრეებს გააჩნიათ კომპონენტების კორპუსებთან სითბური კონტაქტი და მათგან ხსნიან სითბოს ნაბეჭდი ფირფიტის მეტალიზებული მხარის საშუალებით, რომლისგანაც სითბო გადაეცემა მასიურ კორპუსს, რომელიც თავის მხრივ გრილდება გარედან იძულებით. მოდულების სისტემურ გასართებზე გამოიყვანება, როგორც სალტე VME, ისე ყველა გარე ინტერფეისი. კონსტრუქციის აუცილებელი სიხისტე მიიღწევა მეტალური თამასით, რომელიც დაყენებულია მოდულის შუა ნაწილში ნაბეჭდი ფირფიტის გრძელი გვერდის გასწვრივ.

„ბაგეტის“ ტიპის კომპიუტერებში მოდული წარმოადგენს მინიმალურ კონსტრუქციულ ერთეულს. პირობითად მოდულები დაყოფილია ორ ჯგუფად: პროცესორული მოდულები და დამატებითი ინტერფეისების მოდულები.

თითოეული პროცესორული მოდული ფაქტიურად წარმოადგენს ერთ-ფირფიტიან კომპიუტერს, რომელიც შეიცავს ერთიდან ოთხამდე მიკროპროცესორს, სისტემურ კონტროლერს, ოპერატიულ დამხსომებელ მოწყობილობას, Flash -

მეხსიერებას, ასევე ძირითად ინტერფეისებს: IDE, SCSI, Ethernet, გრაფიკულს, IEEE-1284, RS-232 და სხვ. პროცესორულ მოდულზე შესაძლებელია დაყენდეს მეზონინის ორი PMC დამატებითი ინტერფეისებით. მოდულთშორის სალტეს, რომელიც უზრუნველყოფს მიკროპროცესორის დაშვებას გარე ინტერფეისების კონტროლერებთან და VME სალტესთან, წარმოადგენს სალტე PCI, ხოლო ყველა მოდულების სისტემწარმომქმნელ ზედიდ ინტეგრალურ სქემას წარმოადგენენ ხიდი PCI- PCI და ხიდი PCI- VME.

სტანდარტში Compact-PCI პროცესორული მოდული გამოყენებული სერვისული მომსახურებისათვის წარმოადგენს სიმეტრიულ მულტიპროცესორს. ამ მოდულის სისტემური კონტროლერი MBUS - PCI უზრუნველყოფს პარალელური მუშაობის შესაძლებლობას ერთიდან ოთხ მიკროპროცესორამდე, ოდმ-თან, Flash- მეხსიერებასთან, ასევე გასვლას ორ დამოუკიდებელ შიგამოდულურ სალტეებზე PCI-1 და PCI-2. PCI-ის ამ სალტეებზე შესაძლებელია მიერთდეს ინტერფეისები IDE, Ethernet, IEEE-1284, და სხვ. ასევე სპეციალურად ორი სერვისული დანართისათვის ორი ინტერფეისი SCSI-2. გარდა ამისა არსებობს ორი გასართი: PMC1 და PMC2 მეზონინების დაყენებისთვის დამატებითი ინტერფეისებით.

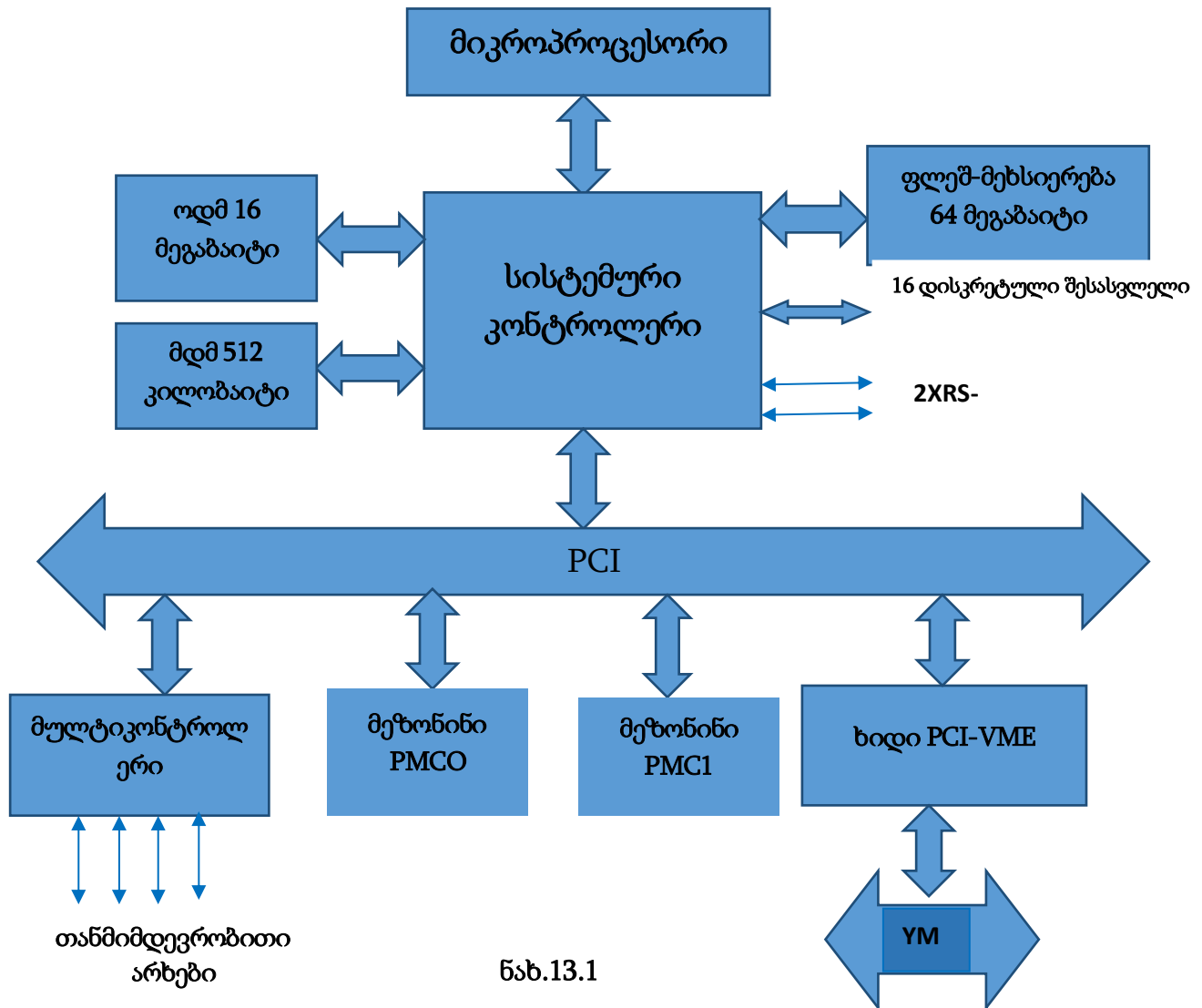
დამატებითი ინტერფეისების მოდულების დანიშნულებაა ბაგეტი ტიპის კომპიუტერის ფუნქციური შესაძლებლობების გაფართოება და სისტემების აგება დიდი რაოდენობის პერიფერიული მოწყობილობებით და კავშირის მისაერთებელი არხებით.

13.2. ბაგეტის ტიპის საბორტო კმპიუტერებში გამოყენებული პროცესორული მოდულები

13.2.1. პროცესორული მოდული BT 33-205

პროცესორული მოდული BT33-205 გამოიყენება „ბაგეტის“ ტიპის კომპიუტერებში და გათვალისწინებულია ექსპლუატაციის ხისტ პირობებში მუშაობისთვის. მისი დანიშნულებაა ერთი და მრავალპროცესორული სისტემების შექმნა სისტემური WME-64 სალტის ბაზაზე. ასევე გამოიყენება როგორც ინტელექტუალური VME -

კონტროლერების ასაგებად ორი სლოტის გამოყენებით მეზონინური მოდულების გაფართოებისათვის PMC კონსტრუქტივში. მოდულის სტრუქტურა წარმოდგენილია ნახ.13.1-ზე.



ნახ.13.1

ცენტალური პროცესორის მოდული **BT 33-205**, რომელიც დამუშავებულია

მიკროპროცესორ „კომდივის“ დონეზე თავის შემადგენლობაში შეიცავს:

- მიკროპროცესორს შიგა ტაქტური სიხშირით 82,5 მგჰერცი;
- სტატიკურ ოდმ-ს მოცულობით 16 მეგაბაიტი;
- ჩამტვირთავ (სისტემურ) მუდმივ დამხსომებელ მოწყობილობას 512

კილობაიტი მოცულობით;

- სამომხმარებლო გადაპროგრამირებად მდმ-ს (ფლეშ-მეხსიერება) 64

მეგაბაიტი მოცულობით;

- სისტემურ კონტროლერს, რომელიც შეიცავს წყვეტების კონტროლერს;
- სამ 32 თანრიგიან პროგრამირებად ტაიმერს;
- ორ ტექნოლოგიური თანმიმდევრობით ინტერფეისს RS-232C, რომლებიც თავსებადია NS16450 -თან (გადაცემის სიჩქარე 115 200 ბოდამდე);
- PCI სალტის კონტროლერს - წამყვანს და მიმყოლს;
- დისკრეტული სიგნალების კონტროლერებს (16 შესასვლელი ხაზით, ორი სამომსახურო ხაზის ჩათვლით პროცესორის რეჟიმის მართვისთვის, 6 გამოსასვლელით) TTL ლოგიკის დონეებით;
- კონტროლერს (ზედიდი ინტეგრალური სქემის მულტიკონტროლერის ბაზაზე), რომელიც უზრუნველყოფს თანმიმდევრობით შეერთებული RS-232C არხების მუშაობას, რომლებიც თავსებადია NS 16650-ის (გადაცემის სიჩქარე 115200 ბოდი);
- ხიდი PCI-VME, რომელიც შეიცავს VME კონტროლერს;
- PMC სტანდარტში ორ PCI სლოტს გაფართოების ორი მეზონინური მოდულების მიერთებისთვის.

მოდულს გააჩნია მრავალსაღატური სტრუქტურა. მიკროპროცესორის საღატე

წარმოდგენს მოდულის ძირითად მაგისტრალს და ერთმანეთთან აკავშირებს პროცესორს, მეხსიერებას და კონტროლერებს. საღატის არბიტრაჟისას ყველაზე უმაღლესი პრიორიტეტი გააჩნია მიკროპროცესორს, ხოლო ყველაზე დაბალი პრიორიტეტი - PCI კონტროლერს.

ნელად მომუშავე მოწყობილობებთან დაშვება ხორციელდება მხოლოდ მიკროპროცესორის მხრიდან და ლოგიკურად ისინი მიერთებული უნდა იყვნენ მის საღატესთან. თუმცა იმის გათვალისწინებით, რომ მოითხოვება მიკროპროცესორის საღატეზე იყოს რაც შეიძლება დატვირთვის ნაკლები რიცხვი, ამდენად მიკროპროცესორთან პირდაპირი მიერთება არასასურველია. გარდა ამისა, მიკროპროცესორის საღატე წარმოდგენს მულტიპლექსურს, რაც განასხვავებს მას ზემოთ ჩამოთვლილი მოწყობილობების საღატეებისგან (გარდა მიმდევრობითი მუდმივი დამხსომებელი მოწყობილობისა) რომელთა მისამართისა და მონაცემების საღატეები

განცვაღკავებული არიან, ამიტომ ე.წ „ნელი“ მოწყობილობები ურთიერთქმედებენ მიკროპროცესორთან სისტემური კონტროლერის საშუალებით, რომელიც თავის მხრივ ემსახურება რამდენიმე სალტეს (ნახ.13.1).

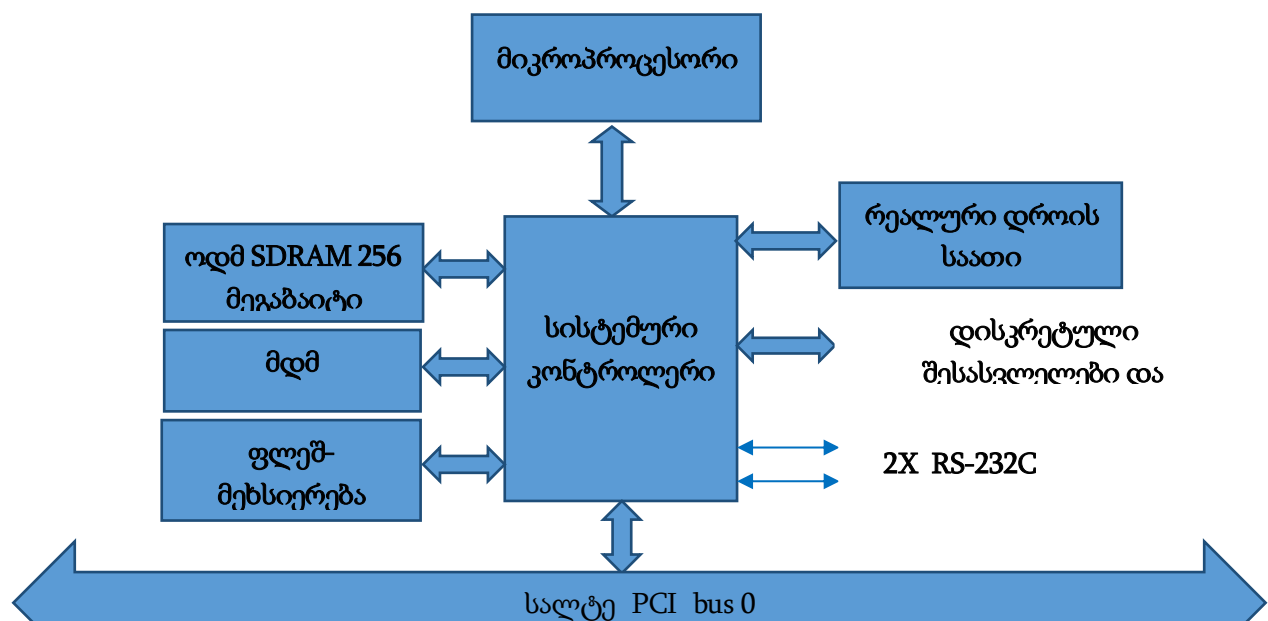
ნელად მომუშავე მოწყობილობებთან მიკროპროცესორების მხრიდან მიმართვა ხდება ისეთნაირად, რომ თითქოს ისინი განლაგებული იყვნენ მასთან ერთ სალტეზე. განსხვავება მდგომარეობს მხოლოდ დაშვების დროში. სისტემური კონტროლერი ამ შემთხვევაში გამოდის მონაცემების სალტისა და მისამართის დეშიფრატორის რეტრანსლიატორის როლში.

BT33-205 მოდულს შეუძლია იმუშაოს VME სალტეზე წამყვანის როლში, ასევე მიმყოლისა და სალტეზე არბიტრის როლშიც. ამასთან ერთად, მხარდაჭერილია მუშაობა სამისამართო ხაზებისა A32, A24, A16 სამისამართო სივრცეში მონაცემების ხაზების ფორმატებით D32, D16 და D8. VME სალტის ინტერფეისი მხარს უჭერს გეოგრაფიულ დამისამართებას (განლაგება კრეიტში).

მოდული იკავებს კრეიტის ერთ პოზიციას.

13.2.2. პროცესორული მოდული BT01-212

BT01-212- პროცესორული მოდული გამოიყენება სხვადასხვა კონფიგურაციისა და მწარმოებლურობის უნივერსალური კომპიუტერების ასაგებად, მართვის ერთ და მრავალპროცესორული ავტომატიზებული სისტემების შესაქმნელად. მოდულის სტრუქტურა წარმოდგენილია ნახ 13.2-ზე.



5sb.13.2

ცენტრალური პროცესორის მოდული მიკროპროცესორ “კომდივის“ ბაზაზე თავის შემადგენლობაში შეიცავს:

- მიკროპროცესორს შიგა ტაქტური სიხშირით 100 მგჰც;
- დინამიკურ ოდმ-ს 256 მგბაიტი ტევადობით და ხემინგის კოდით კონტროლით;
- სისტემური მდმ-ს 512 კბაიტი ტევადობით;
- სამომხმარებლო რმდმ-ს (ფლეშ-მეხსიერებას) 16-32 მგბაიტი მოცულობით;
- სისტემურ კონტროლერს, რომელიც რეალიზებულია ზედიედი ინტეგრალური სქემის სახით და შეიცავს წყვეტის კონტროლერს, 3 ცალ 32-თანრიგიან ტაიმერს, ორ ტექნოლოგიურ თანმიმდევრობით ინტერფეისს RS-232C, PCI სალტის კონტროლერს-წამყვანი და მიმყოლი, პროგრამირებადი დისკრეტული სიგნალების კონტროლერს (16 შესასვლელით და 6 გამოსასვლელით);
- მულტიკონტროლერს, რომელიც შეიცავს და უზრუნველყოფს ოთხი თანმიმდევრობითი გაცვლის RS-232C ინტერფეისისა და ერთი RS-485 ინტერფეისის მუშაობას, შეიცავს ასევე პარალელური პორტის ერთ ინტერფეისს (სტანდარტი IEEE 1284), ახორციელებს IDE ინტერფეისის მუშაობას(სტანდარტი ANSI ATA-2) SPP, ECP და EPP რეჟიმებში;
- Ethernet კონტროლერს 10/100 მგბიტი/წმ-ში გადაცემით;
- DCI-ის ერთ სლოტს, სპეციალიზებულს PMC სტანდარტში ერთი მეზონინური ფირფიტის ჩართვისთვის SVGA გრაფიკის კონტროლერთან (1600x1280, 256 ფერი, 85ჰც, ვიდეო მეხსიერება 16 მგბაიტი);
- ერთ უნივერსალურ სლოტს (PMC1) შესასვლელი და გამოსასვლელი დისკრეტული სიგნალების კონტროლერით;
- რეალური დროის საათს;
- ხიდს PCI-PCI.

BT01-212 მოდულის სისტემური კონტროლერი იყენებს მიკროკონტროლერების ოჯახს MIPS - ერთკრისტალიანი მიკროპროცესორის არქიტექტურის ბაზაზე.

სისტემური კონტროლერის ზედიდი ინტეგრალური სქემა უზრუნველყოფს მუშაობას ჩატვირთვად მეხსიერებასთან (მდმ), დინამიკურ ან სტატიკურ მეხსიერებასთან (ოდმ), შეტანა-გამოტანის პროგრამირებადი ხაზებით და PCI სალტით.. სისტემური კონტროლერი ასევე შეიცავს წყვეტის ვექტორულ კონტროლერს, ტაიმერებს და RS -232C პორტებს.

ზედიდი ინტეგრალური სქემის შემადგენლობაში შედის:

- რეგისტრული ფაილი, რომელიც შეიცავს პროგრამულად დაშვებად რეგისტრებს, რომელთა მდგომარეობა განსაზღვრავს სისტემური კონტროლერის მუშაობას მეხსიერებასთან და პერიფერიულ მოწყობილობებთან;
- ინტერფეისის მართვის სქემა მიკროპროცესორით, რომელიც პასუხისმგებელია მიკროპროცესორისთვის სიგნალების გამომუშავებაში, შეიცავს ტაიმ-აუტის სქემას შეცდომითი ციკლების შეწყვეტისთვის და სქემას შეცდომითი სიტუაციების დამუშავებისათვის. მიკროპროცესორის სალტეზე სისტემური კონტროლერი წარმოადგენს მიმყოლ მოწყობილობას, გარდა იმ შემთხვევისა როცა ხორციელდება PCI Seave-ს მხრიდან მიმართვა, როცა სისტემური კონტროლერი მოითხოვს სისტემური PCI სალტის მრთვას და მაშინაც, როცა ინიცირებას უკეთებს მეხსიერებასთან პირდაპირი დაშვების ციკლებს;
- ოდმ-ის კონტროლერს, რომელიც შედგება ორი ავტონომიური მოწყობილობისგან: სტატიკური ოდმ-ის კონტროლერისგან და დინამიკური ოდმ-ის კონტროლერისგან.
- მდმ-ის კონტროლერს, რომელიც გათვლილია როგორც 32 თანრიგიანი მდმ-ის მუშაობისთვის ისე 8 თანრიგიანი მდმ-თან მუშაობისთვის. მდმ-ის ტიპი მოცემა შესაკრავით და არ იცვლება მუშაობის დროს.
- წყვეტის კონტროლერს, რომელიც საშუალებას იძლევა გამოიტანოს მიმართვები წყვეტების შიგა წყაროებიდან წყვეტის ოთხიდან არსებულ ერთ-ერთ ხაზზე, რაც იძლევა საშუალებას დაპროგრამდეს წყვეტების ხაზი და მიმართვის პრიორიტეტი;
- სამ 32 თანრიგა ტაიმერს, რომელთა მართვა დასაშვებია პროგრამულად;
- PCI კონტროლერს, რომელიც შეიცავს PCI სალტის წამყვანი მოწყობილობის კონტროლერს, PCI ასალტის მიმყოლი მოწყობილობის კონტროლერს, PCI

არბიტრს, რომელიც თავის მხრივ ახორციელებს სამამადე PCI მოწყობილობების არბიტრაჟს პრიორიტეტული არბიტრაჟის რეჟიმში;

- RS-232 პორტის ორ კონტროლერს, რომლებიც პროგრამულად თავსებადია NS 16450-თან;

- დისკრეტული შეტანა გამოტანის ხაზებს (16 შესასვლელით, 6 გამოსასვლელით), რომლებიც შესაძლებელია დაპროგრამირდნენ მომხმარებლების მიერ მონაცემების შეტანა-გამოტანისთვის, ასევე შესაძლებელია გამოიყენებოდნენ სპეციალური ფუნქციების შესასრულებლად: ტაიმერის სიხშირის მოსაცემად და გარე წყვეტების შესასვლელის განსაზღვრისთვის.

სისტემური კონტროლერის ზედიდე ინტეგრალური სქემა მუშაობს მიკროპროცესორის გარე სალტის სიხშირით. შიგა მოწყობილობების ნაწილი მუშაობს ფისქირებული სიხშირიდან 24 მგჰერციადნ. ეს ეხება მოწყობილობებს, რომლებმაც უნდა უზრუნველყონ ფისქირებული დროითი პარამტერები ე.ი ტაიმერებს და RS 232-ის თანმიმდევრობით პორტებს.

მულტიკონტროლერის ზედიდე ინტეგრალური სქემა PCI სალტის მიხედვით მართვით ემსახურება პარალელური და თანმიმდევრობითი პორტების მიხედვით ინფორმაციის შეტანას და გამოტანას. სალტის მაქსიმალური (PCI სალტის) ტაქტური სიხშირე 33 მგჰერცია.

მულტიკონტროლერის შემადგენლობაში შედის:

- პარალელური პორტის ერთი კონტროლერი - LPT (IEEE 1284);
- ერთი კონტროლერი IDE (ორი არხი), ოთხი კონტროლერი მიმდევრობითი პორტების 16560 (COM1, COM2, COM3, COM4);
- კლავიატურის კონტროლერი, კონტროლერი PS2;
- მიმდევრობითი კონტროლერი, რომელიც თავის მხრივ წარმოადგენს მიმდევრობით პორტს NS16450 (COM5);

ხიდი PCI-VME წარმოადგენს PCI და VME სალტების შეუღლების კონტროლერს. ის უზრუნველყოფს მონაცემების გაცვლას პერიფერიულ PCI მოწყობილობებსა და VME სალტეზე წამყვან მოწყობილობებს შორის, ასევე უზრუნველყოფს VME-ს დაქვემდებარებულ ოწყობილობებსა და პროცესორულ

კრისტალებს შორის გაცვლას PCI სალტეზე. კონტროლერი ასრულებს 32 თანრიგიანი PCI ინტერფეისის ფუნქციებს და VME 64 თანრიგიანი სალტის ფუნქციებსაც.

PCI სალტე გამოიყენება დამატებითი მოწყობილობის მიერთებისთვის, რომელიც შესრულებულია ორი მეზონინური ფირფიტის სახით P1396 სტანდარტის შესაბამისად (მეზონინები PMCO და PMC1).

VME სალტე უზრუნველყოფს მონაცემების გაცვლას კომპიუტერის სხვა VME მოდულებთან.

ხიდი PCI-PCI რეალიზებას უკეთებს PCI სალტის გაფართოების კონტროლერს. კონტროლერი აკმაყოფილებს PCI 2.1-ის მოთხოვნებსა და სპეციფიკაციას. კონტროლერი იყენებს ორ PCI 32 თანრიგიან სალტეს. PCI პირველადი და მეორეადი სალტეებისა მაქსიმალური ტაქტური სიხშირე შეადგენს 33 მეგაჰერცს.

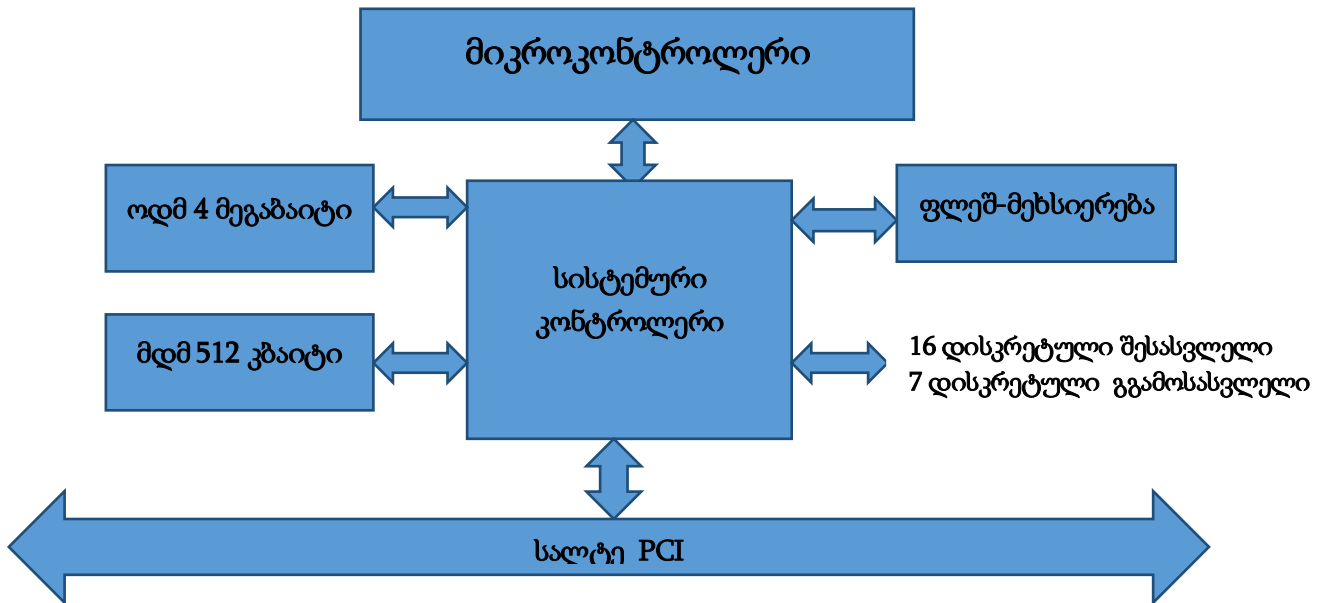
პირველადი PCI ინტერფეისი უერთდება სისტემური კონტროლერს PCI სალტესთან. ხიდის მეორადი PCI ინტერფეისი ქმნის ახალ და დამოუკიდებელ სალტეს PCI-ს. ხიდის ძირითადი დანიშნულებაა - განახორციელოს გადაცემა ინიციატორსა (რომელიც იმყოფება ერთ PCI სალტეზე) და შემსრულებელს შორის (რომელიც იმყოფება მეორე PCI სალტეზე).

ხიდი იძლევა საშუალებას, რომ PCI ორმა სალტემ იმუშაონ ერთმანეთისგან დამოუკიდებლად. ინიციატორი და შემსრულებელი იმყოფებიან რა PCI ერთ სალტეზე, უკავშირდებიან ერთმანეთს დამოუკიდებლად იმისა დაკავებულია თუ არა მეორე სალტე. როგორც შედეგი, ხიდი იზოლირებას უკეთებს ორ ტრაფიკს მოწყობილობებს შორის, რომლებიც იმყოფებიან ერთ PCI სალტეზე. ეს განსაკუთრებული საშუალებას იძლევა ამაღლდეს სისტემის მწარმოებლურობა გამოყენების შემთხვევაში.

იმდენად რამდენადაც PCI სალტე მუშაობს დატვირთვის შეზღუდული რიცხვის ფარგლებში, ამდენად ხიდის გამოყენება საშუალებას იძლევა გაფართოვდეს PCI მოწყობილობების რაოდენობა სისტემაში.

13.2.3. პროცესორული მოდული BT83-201

პროცესორული მოდული BT83-201 ჩაშენებული მცირეგაზარიტიანი კომპიუტერისთვის წარმოადგენს ფაქტობრივად ერთფირფიტიან უკორპუსო კომპიუტერს, რომელიც შესაძლებელია გამოყენებულ იქნეს, როგორც დამოუკიდებლად, ასევე დამატებით ინტერფეისულ მოდულებთან ერთად. პროცესორული მოდული შედგება 90 x 96 მმ ზომის ნაბეჭდი ფირფიტისგან. მოდულის სტრუქტურა ნაჩვენებია ნახ.13.3-ზე.



ნახ.13.3

მოდულის შემადგენლობაში შედის:

- RISC-მიკროპროცესორი 33-82,5 შიგა ტაქტური სიხშირით და PCI სისტემური სალტე 33 მეგა ჰერცი სიხშირით;
- სტატიკური ოდმ 4 მეგაბაიტი ტევადობით;
- სისტემური მდმ 512 კილობაიტი ტევადობით;
- სამომხმარებლო ნახევრადმუდმივი (რეპროგრამირებადი) მუდმივი დამხსომე-ბელი მოწყობილობა (ფლეშ-მეხსიერება) 4 მეგაბაიტი ტევადობით;
- სისტემური კონტროლერი (BT33-212-ში გამოყენებული კონტროლერის ანალოგიური), რომელიც შეიცავს წყვეტების კონტროლერს, სამ 32 თანრიგის ტაიმერს, RS-232C თანმიმდევრობითი ინტერფეისების ორ კონტროლერს, PCI სალტის

კონტროლერს - წამყვანს და მიმყოლს, დისკრეტულ შეტანა-გამოტანას (16 შესასვლელი და 6 გამოსასვლელი ხაზებით) ტრანზისტორულ - ტრანზისტორული ლოგიკის დონეებით.

მიკროპროცესორული მოდულის BT83-201-ის გაზარიტული ზომებია 103 x 96x 16,5 მმ, მასა ტოლია 0,15 კგ.

ამ მოდულის ფუნქციონირება BT 33-205 და BT 33-212 მოდულების ანალოგიურია.

13.3. საფრენ აპარატებზე გამოყენებადი „ბაგეტი“ ტიპის მმართველი საბორტო კომპიუტერები

ქვემოთ ჩვენ მოვიყვანთ იმ კომპიუტერების მოკლე დახასიათებას, რომლებიც რეალიზებულია აპარატულ-პროგრამული გამოთვლითი პლატფორმის „ბაგეტი“ ბაზაზე და გათვლილია საფრენი აპარატების განსაკუთრებით ხისტ და რთულ პირობებში ფუნქციონირებისათვის, კერძოდ მაღალი და დაბალი ტემპერატურა, ვიბრაცია, დარტყმები და გადატვირთვები. ასეთ კომპიუტერთა ოჯახს მიეკუთვნებიან ბაგეტი 33, ბაგეტი 53-18, ბაგეტი 53-28, ბაგეტი 53-31, ბაგეტი 55 და ბაგეტი-83. ჩამოთვლილი კომპიუტერებიდან პირველი ხუთი რეალიზებულია გერმეტულ კორპუსში, მათი ბუნებრივი ან იძულებითი ჰაერის ნაკადით გაგრილების უზრუნველყოფით. ელექტრულ კვებას ღებულობენ საბორტო ქსელიდან. რაც შეეხება საბორტო კომპიუტერს ბაგეტი-83, ის უკორპუსოა და თავსდება სხვა მოწყობილობებთან ერთად ერთ საერთო კორპუსში.

13.3.1. მმართველი კომპიუტერი ბაგეტი-33

მმართველი კომპიუტერი ბაგეტი-33 გათვალისწინებულია საფრენ აპარატებზე განთავსებისა და მუშაობისთვის. ის შესრულებულია გერმეტულ კორპუსში შიგა ვენტილატორების გარეშე კორპუსის იძულებითი გაგრილების გზით, რომელიც მიღწეულია კორპუსის გვერდითი კედლების შიგნით ჰაერის შებერვის განხორციელებით.

ბაგეტი-33-ის შემადგენლობაში შედის:

- IB 812 ტიპის მიკროპროცესორი ტაქტური სიხშირით 50 მეგაჰერცი;
- VME სალტე;
- ოდმ -16 მეგაბაიტი, რეპროგრამირებადი მდმ -32 მეგაბაიტი;
- ორი RS-232C არხი;
- სამი დაპროგრამებადი ტაიმერი;
- კვების ბლოკი, რომელიც უზრუნველყოფს 120 ვატი სიმძლავრის მოხმარებას, პირველადი ქსელი 220 ვოლტი ძაბვით და 50-400 ჰერცი სიხშირით ან მუდმივი ძაბვა 27 ვოლტი +10%-18%.

1. კომპიუტერი ნორმალურად ფუნქციონირებს შემდეგ პირობებში:

- ვიბრაცია არ აჭარბებს 6g-ს დიაპაზონში 1-500 ჰერცი;
- ერთჯერადი დარტყმები 500g;
- მრავალჯერადი დარტყმები 20g;
- მუშა ტემპერატურა დიაპაზონში -50 + 55°C-მდე;
- ტენიანობა 100%-მდე 35° C-ის დროს;
- წნევა არა ნაკლები 5 მილიმეტრი ვერცხლისწყლის სვეტისა.

პროგრამული უზრუნველყოფა შეიცავს:

- რეალური დროის ოპერაციულ სისტემას და რეპროგრამირებადი მუდმივი დამხსმომეხელი მოწყობილობის ჩამტვირთავს;
- პროგრამულ-ტექნოლოგიურ კომპლექსს გამოყენებითი პროგრამების კროსდამუშავებისა და გამართვისათვის **Unix-ის** ტიპის ოპერაციული სისტემის გარემოში.

კომპიუტერის კონსტრუქცია წარმოადგენს კორპუსს მუსბუქი შენადნობისგან დამზადებულს გერმეტული გასართებით, კროსფირფიტას VME სალტით, კონსტრუქტივში „ევრო მექანიკა-64“-ის 5 მოდულის განლაგების შესაძლებლობით, მოდულებს სითბოარინების და სითბოგადაცემის საშუალებების გამოყენებით.

კომპიუტერის გაზომვები 400x203x124მმ, მასა 8 კგ.

საბაზო კომპლექტში შედის კორპუსი კროსფირფიტით, გაგრილებისა და ელექტროკვების სისტემებით და პროცესორული მოდული BT33-202. კომპიუტერის კორპუსში შესაძლებელია დაყენდეს ოთხი დამატებითი მოდული VME სალტით.

ცენტრალური პროცესორის მოდულზე BT33-202 (205) შესაძლებელია დაყენდეს ორი მეზონინური ფირფიტა. შესაძლებელია კომპიუტერის მრავალპროცესორული კონფიგურაცია.

13.3.2. საბორტო კომპიუტერი ბაგეტი-53-18

მისი ვიზუალური სახე ნაჩვენებია სურ, 13.4-ზე.



საბორტო კომპიუტერი „ბაგეტი-53-18“-ის
შემადგენელი კომპონენტები და საექსპლუა-
ტაციო მახასიათებლები:

მიკროპროცესორი

ტიპი: 1B578

მიკროპროცესორის არქიტექტურა RISC

სიხშირე : 50 MHz

სურ.13.4

პროცესორის მოდული: BT33-202

პროცესორის მოდულის დიზაინი: IEEE 1101.2;

პროცესორის მოდულების რაოდენობა: 1;

ინტერფეისი ანტრესოლით მოდულების დასაკავშირებლად: PCI სლოტი PMC სტანდარტი.

შესასვლელ-გამოსასვლელი არხები:

2 დამოუკიდებელი შესასვლელი არხი;

2 დამოუკიდებელი გამოსასვლელი არხი;

RS-232C ინტერფეისის 3 დამოუკიდებელი შესასვლელი არხი;

RS-232C ინტერფეისის 3 დამოუკიდებელი გამოსასვლელი არხი;
ინფორმაციის გაცვლის 1 მულტიპლექსური არხი.

პროგრამული უზრუნველყოფა:

ინიციალიზაციის პროგრამა;

პროგრამირების ენები: C, C/C++.

მოხმარებული სიმძლავრე: არაუმეტეს 50 ვტ;

ნაკეთობის წონა: არაუმეტეს 7 კგ;

გაზარიტული ზომები: 325x125x247 მმ;

ფრენისას მტყუნებამდე ნამუშევარის საშუალო დრო: 5000 სთ.

13.3.3. საბორტო კომპიუტერი ბაგეტი-53-28

საბორტო კომპიუტერი ბაგეტი 53-28 (სურ.13.5) განკუთვნილია საბორტო რადიოელექტრონული კომპლექსის ქვესისტემების და შეუღლებადი მოწყობილობების ურთიერთქმედების ორგანიზაციისათვის, ინფორმაციის კომპლექსური დამუშავებისათვის, მართვის, ნავიგაციის და სპეციალური ამოცანების გადაწყვეტისათვის.

„ბაგეტი-53-28“ საბორტო კომპიუტერის ტექნიკური მახასიათებლები:



მიკროპროცესორი :

ტიპი: KOMDIV- 64SMP (1890VM5F);

მიკროპროცესორის არქიტექტურა: MIPS IV;

ბირთვის ტაქტური სიხშირე: 200 MHz.

პორცესორის მოდული : MPON-A

პროცესორის მოდულის კონსტრუქცია:

Euromechanics 6U;

სურ.13.5

პროცესორებშორისი ურთიერთქმედების მექანიზმი:

- კომუტაციური ტექნოლოგია პარალელური Rapidio;
- ტოპოლოგია - „ თითოეული - თითოეულთან “;
- მეზონინური მოდულების მიერთების ინტერფეისი PCI 2.1, 33 MHz, 32 ბიტი.

შესასვლელ-გამოსასვლელი არხების რაოდენობა :

- 3 ხაზი Ethernet 100 Mb/s;
- 100 base TX სტანდარტი, STP გადაცემის გარემოს საშუალებით;
- მულტიპლექსის გაცვლის 4 არხი დარეზერვებით; კონტროლერის, ბოლო მოწყობილობისა და მონიტორის ფუნქციების შესრულებით;
- 16 შესასვლელი მიმდევრობითი გაცვლის დამოუკიდებელი არხები.

შესასვლელი ერთჯერადი ბრძანებები:

- 12 შესასვლელი ელექტრულად იზოლირებული საერთო გამტარის ხაზიდან „27 ვ ბორტი-ქსელი“;
- 8 შესასვლელი ხაზი „საერთო - გაწყვეტა“;

გამოსავლელი ერთჯერადი ბრძანებები;

- 12 გამოსავლელი ელექტრულად იზოლირებული საერთო გამტარის ხაზიდან „27ვ ბორტი-ქსელი“;
- 8 გამოსავლელი ხაზი „საერთო-გაწყვეტა“;
- 1 გამოსასვლელი სიგნალი კომპიუტერის „გამართულობა“.

პროგრამული უზრუნველყოფა:

- რეალური დროის ოპერაციული სისტემა OS RV Baguette 3.0;
- მოწყობილობების მომსახურების პროგრამა (დრაივერები, მოდულები);
- ჩაშენებული კონტროლის ტესტი;
- ავტონომიური კონტროლის ტესტი;
- ფუნქციური პროგრამული უზრუნველყოფა;

- პროგრამირების ენები: Assembler, C/C++;

კვების ძაბვა: 27 V DC GOST R 54073-2010;

მოხმარებული სიმძლავრე: არაუმეტეს 250 ვტ;

ნაკეთობის წონა: არაუმეტეს 8 კგ.

კონსტრუქციული შესრულება: მონობლოკი 2K GOST 26765.16-87-ის მიხედვით;

ექსპლუატაციის პირობები GOST V 20.39.304-76-ის მიხედვით;

ფრენისას საშუალო მტყუნებამდე ნამუშევარი დრო: 7000 სთ.

13.3.4. საბორტო კომპიუტერი ბაგეტი 53-31

ბაგეტი 53-31 (სურ.13.6) არის საბორტო კომპიუტერი, რომელიც ასრულებს გამოთვლით პროცესებს და ამ ყველაფერში ეხმარება PCI სალტე. ბაგეტი-53-31 თავსებადია დამატებით ელექტრონულ მოდულებთან, რაც განპირობებულია მისი ღია სტრუქტურით, ეს საშუალებას იძლევა კომპიუტერი დამაგრდეს სპეციალურ კორპუსში.



სურ.13.6.

ბაგეტი 53-31 საბორტო კომპიუტერის ტექნიკური მახასიათებლები:

მიკროპროცესორი:

- მიკროპროცესორის ტიპი: KOMDIV-64SMP (1890VM5F), MIPS IV არქიტექტურა;
- ბირთვის ტაქტური სიხშირე : 260 MHz;
- პროცესორის მოდული: MPON-1;
- პროცესორის მოდულის კონსტრუქცია: euromechanics 6U;
- პროცესორის მოდულების რაოდენობა: 7 ც;
- პროცესორთშორისი ურთიერთქმედების მექანიზმი: კომპუტაციური ტექნოლოგია RapidIO;
- ტოპოლოგია - ორმაგი "ვარსკვლავი";
- მეზონინური მოდულების დასაკავშირებლად გამოყენებულია: PCI 2.1 33 MHz, 32 ბიტი;
- MPON-A BTsVM-ზე მოდულების რაოდენობა: 2 მოდული, ტიპი PMC.

შესასვლელ-გამოსასვლელი არხები კონტროლის გათვალისწინებით :

- ბოჭკოვანი 2 არხი FC-AE-ASM (FC-RV-PSNA) ინფორმაციის გაცვლისთვის, მონაცემთა გადაცემის სიჩქარე -1062.5 Mb/s;
- სიგნალების 2 შესასვლელი და 2 გამოსასვლელი არხი, ბოჭკოვანი არხის ინტერფეისი პროფილის მიხედვით ARINC 818, მონაცემთა სიჩქარე - 1062,5 Mb/s;
- 16 შესასვლელი და 8 გამოსასვლელი დამოუკიდებელი არხი;
- მულტიპლექსის გაცვლის 4 არხი სიჩქარით GOST R 52070-2003 შესაბამისად
- 12 შემავალი და 12 გამომავალი არხი "27 V ბორტი / ქსელი" GOST 18977-79 მიხედვით;
- Ethernet 100Base-TX ინტერფეისის ტექნოლოგიური არხები და RS-232.

პროგრამული უზრუნველყოფა :

- რეალური დროის ოპერაციული სისტემა OS RV Baguette 2.0;

- ინიციალიზაციის პროგრამა;
- ჩაშენებული საკონტროლო ტესტები;
- ავტონომიური კონტროლის ტესტები.
- კვების ძაბვა 27 ვ;
- მოხმარებული სიმძლავრე: არაუმეტეს 160 ვტ;
- ნაკეთობის წონა: არაუმეტეს 18 კგ;
- კონსტრუქციული შესრულება: მონობლოკი 4K GOST 26765.16-87 მიხედვით;
- ექსპლუატაციის პირობები GOST RV 20.39.304-98 მიხედვით;
- ფრენისას საშუალო მტყუნებამდე ნამუშევარი 7000 სთ.

13.3.5. საბორტო კომპიუტერი ბაგეტი 55

საბორტო კომპიუტერი ბაგეტი 55 წარმოადგენს სპეციალიზებულ კომბინირებულ მრავალპროცესორიან კომპიუტერს რადიოლოკაციური, აკუსტიკური და ოპტიკური სიგნალების ციფრული დამუშავებისათვის, ასევე მართვის ამოცანების გადაწყვეტისათვის დროის რეალურ მასშტაბში. მის შემადგენლობაში შედის სიგნალების პროგრამირებადი პროცესორი და მმართველი სპეციალიზებული ელექტრონული გამოთვლითი მანქანა, რომლებიც გაერთიანებული არიან გაცვლის საერთო მაგისტრალით (VME).

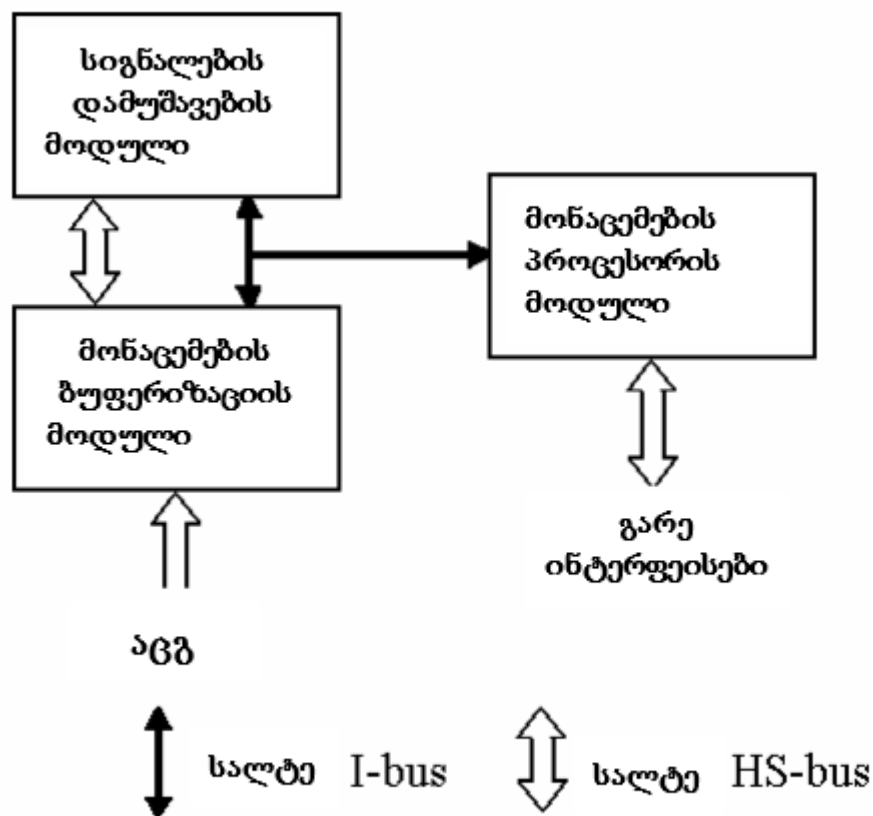
ბაგეტი 55 გამიზნულია, როგორც გამომთვლელი სხვადასხვა დანიშნულების საავიაციო საბორტო სისტემებში სიგნალების მრავალარხიანი ციფრული დამუშავებისათვის დროის რეალურ მასშტაბში, ისე სხვადასხვა მართვის სისტემების შემადგენლობაში გამოყენებისათვის.

ის გათვალისწინებულია ექსპლუატაციის რთულ (ხისტ) პირობებში მუშაობისთვის:

- ვიბრაცია ნაკლებია ან ტოლია 6გ-ზე 20-2000 ჰერცი სიხშირის დიაპაზონში;
- ერთეული დარტყმები 100გ-მდე;
- მრავალჯერადი დარტყმები ≤ 20 გ-მდე;
- ხაზოვანი აჩქარება 34 გ-მდე;
- მუშა ტემპერატურა -50-დან+60°C-მდე;
- ტენიანობა 100%-მდე 35°C-ის დროს;
- მუშა წნევა 5მმ ვერცხლისწყლის სვეტზე მეტი.

ბაგეტი 55-ის სტრუქტურული სქემა მოყვანილია ნახ.13.7-ზე. ის აგებულია სამი ტიპის მოდულის საფუძველზე:

- სიგნალების დამუშავების მოდული (სდმ);
- მონაცემების პროცესორის მოდული(მპმ);
- მონაცემების მიღებისა და ბუფერიზაციის მოდული (მმმ).



ნახ.13.7

ბაგეტი 55-ს შეუძლია ფუნქციონირება ორ რეჟიმში - საშტატო და ტექნოლოგიურ რეჟიმებში.

მუშაობის საშტატო რეჟიმში უზრუნველყოფილია:

- საბორტო კომპიუტერის საწყისი ინიციალიზაცია ელექტრული კვების ჩართვის დროს;
- მომხმარებლის ბრძანების მიხედვით საწყისი ტესტირების რეჟიმის განსაზღვრა;
- მეხსიერებაში ტესტირების შედეგების გაცემა;
- მომხმარებლის პროგრამაზე მართვის გადაცემა, რომელიც განლაგებულია

გადაპროგრამირებად მუდმივ დამხსომებელ მოწყობილობაში.

მუშაობის ტექნოლოგიურ რეჟიმში უზრუნველყოფილია:

- საბორტო კომპიუტერის საწყისი ინიციალიზაცია ელექტრული კვების ჩართვის დროს;
- მომხმარებლის ბრძანების მიხედვით საწყისი ტესტირების რეჟიმის განსაზღვრა;
- საბორტო კომპიუტერის ტესტირება არჩეული რეჟიმის შესაბამისად;
- მეხსიერებაში ტესტირების შედეგების გაცემა;
- მომხმარებლის პროგრამების დისტანციური ჩატვირთვა ინსტრუმენტული პერსონალური კომპიუტერიდან;
- მომხმარებლის პროგრამის გაშვება შესრულებაზე;
- გამართული მუშა პროგრამული უზრუნველყოფის ჩაწერა რეპროგრამირებად მუდმივ დამხსომებელ მოწყობილობაში.

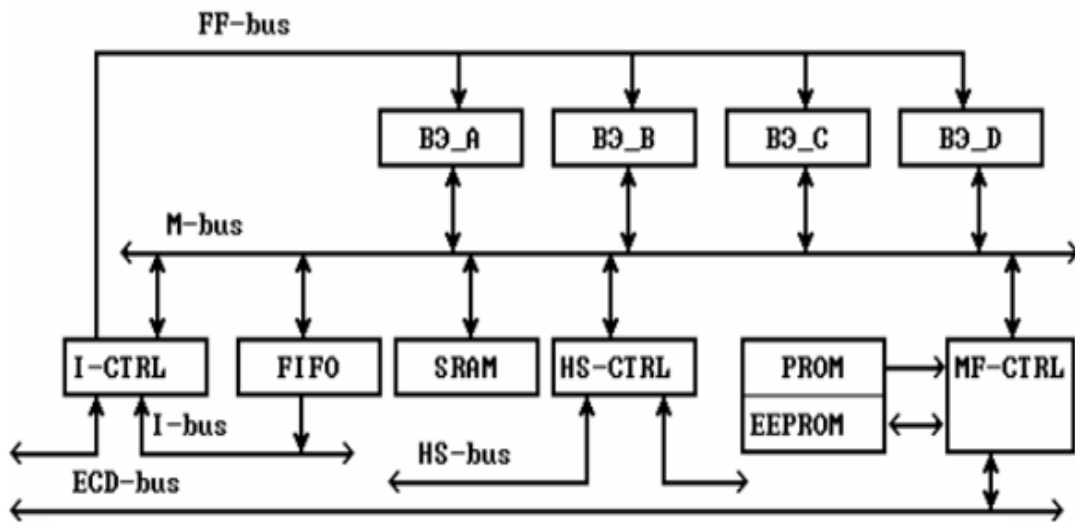
სიგნალების დამუშავების მოდული

სიგნალების დამუშავების მოდული წარმოადგენს გამოთვლით მოწყობილობას სიმეტრიული შიგა არქიტექტურით, რომელიც შეიცავს ოთხ სიგნალურ მიკროპროცესორს და წარმოადგენს ძირითად გამოთვლით ერთეულს სიგნალების პროგრამირებადი პროცესორებისთვის.

სიგნალების დამუშავების მოდულის დანიშნულებაა ინფორმაციის იმ ნაკადების დამუშავება, რომლებიც მიეწოდება I სალტის მიხედვით მონაცემების ბუფერიდან ან სხვა სიგნალების დამუშავების მოდულებიდან დროის რეალურ მასშტაბში მონაცემების პროცესორის მოდულის მართვის შედეგად.

სიგნალების დამუშავების მოდული შეიცავს 4 სასიგნალო პროცესორს IB577 (DSP-პროცესორს), რომლებიც მუშაობენ 40 მეგაჰერც სიხშირემდე და გააჩნია პიკური მწარმოებლურობა 160 MIPS ან 240 MFLOPS.

მოდულის სტრუქტურა ნაჩვენებია ნახ.13.8-ზე.



ნახ. 13.8. სიგნალების დამუშავების მოდულის სტრუქტურული სქემა

სიგნალების დამუშავების მოდულში შედის შემდეგი ფუნქციური ბლოკები:

- გამოთვლითი ელემენტები BE-A-BE-D, რომლებიც შეიცავენ პროცესორს IB577, ლოკალურ სტატიკურ მეხსიერებას (SRAM) მოულობით 2 მეგაბაიტი, ლოკალურ დამხსომებელ მოწყობილობას (სტეკური მეხსიერება) „პირველი მოვიდა -პირველი გავიდა“ ტიპის რიგების ორგანიზაციისთვის (first in-first out fifo) 8K X 32 - თანრიგიანი სიტყვების ტევადობით და მართვის ლოგიკით;
- საერთო სტატიკური მეხსიერება 2 მეგაბაიტი ტევადობით;
- ჩამტვირთავი მუდმივი დამხსომებელი მოწყობილობა (მდმ) არანაკლები 32 კილობაიტისა (PROM);
- გამოსასვლელი FIFO, რომელიც განლაგებულია პროცესორების საერთო სალტეზე (M-bus);
- HS სალტის კონტროლერი (HS-CTRL);
- მრავალფუნქციური კონტროლერი (MF-CTRL);

- ენერგოდამოუკიდებელი მიმდევრობითი მუდმივი დამხსომებელი მოწყობილობა 2 კილობიტი ტევადობით (EEPROM);
- I სალტის კონტროლერი (I-CTRL).
მოდულის შიგა სალტებს მიეკუთვნება:
 - M სალტე - სიგნალური პროცესორების საერთო სალტე;
 - LX სალტე - ლოკალური სალტეები B Θ -A...B Θ -D;
 - FF სალტე - ლოკალური FIFO -ების მონაცემების შესავლელი სალტე.
 გარე სალტებს მიეკუთვნება:
 - I სალტე - ანალოგურ-ციფრული გარდამქმნელებიდან მონაცემების მიღება-გადაცემის სალტე, გამოიყენება მონაცემების ლოკალური მოდელებშორისი გაცვლისათვის;
 - HS სალტე - მონაცემების მოდულებშორისი გაცვლის სალტე;
 - ECD სალტე - სისტემაში პროცესების მართვის სალტე.

მონაცემების პროცესორის მოდული

მონაცემების პროცესორის მოდული წარმოადგენს კომბინირებულ მოწყობილობას და ასრულებს ბაგეტ 55-ში მმართველი მოწყობილობის ფუნქციას და ახორციელებს მონაცემების გაცვლას კომპიუტერსა და გარე სისტემებს შორის.

პირობითად შესაძლებელია მოდული დაიყოს ორ ნაწილად: სუბმოდული A და სუბმოდული B. სუბმოდული A რეალიზებულია სასიგნალო IB577 მიკროპროცესორზე კომპიუტერის მართვისათვის (მოვლენების მართვა, რეჟიმების შეცვლის პროცედურები, გამართვა და ა.შ.). სუბმოდული B რეალიზებულია ერთკრისტალური RISC IB578 მიკროპროცესორზე და უზრუნველყოფს საბორტო კომპიუტერის შეუღლებას გარე სისტემებთან. კავშირი ამ მოდულს შორის ხორციელდება ორპორტიანი მეხსიერების საშუალებით.

მონაცემების პროცესორის მოდულის შემადგენლობაში შედის შემდეგი ფუნქციური კვანძები:

სუბმოდული A:

- IB577 მიკროპროცესორი, რომელიც მუშაობს 40 მეგაჰერცის სიხშირემდე;
- ლოკალური სტატიკური მეხსიერება (Local SRAM) 1 მეგაბაიტი მოცულობით;
- ჩასატვირთი მიმდევრობითი გადაპროგრამირებადი მუდმივი დამხსომეხელი მოწყობილობა 256 კილობიტი მოცულობით;
- გლობალური სტატიკური მეხსიერება (Global SRAM) 1 მეგაბაიტი მოცულობით;
- HS სალტის კონტროლერი (HS-CTRL);
- მრავალფუნქციური კონტროლერი (Multy-CTRL).

სუბმოდული B:

- RISK IB578 მიკროპროცესორი, რომელიც მუშაობს 50 მეგაჰერც სიხშირემდე;
- მიკროპროცესორის სტატიკური მეხსიერება (SRAM) 8 მეგაბაიტი მოცულობით;
- ჩასატვირთი პროგრამირებადი მუდმივი დამხსომეხელი მოწყობილობა 4 მეგაბიტი მოცულობით;
- ფლეშ-მეხსიერება 16 მეგაბაიტი მოცულობით;
- ბუფერულ სტატიკური მეხსიერება (Buffer SRAM) 1 მეგაბიტი მოცულობით, რომელიც დაშვებადია როგორც პროცესორის მხრიდან, ისე VME მაგისტრალის მხრიდანაც.
- მონაცემების გაცვლის თანმიმდევრობითი არხების კონტროლერი RS232 ტიპის;
- უნივერსალური კონტროლერი (Uni-CTRL);
- PCI სალტის კონტროლერი (PCI- CTRL);
- VME სალტის კონტროლერი (VME -CTRL);
- ორპორტიანი მეხსიერება 64 კილო 32 თანრიგიანი სიტყვის მოცულობამდე.

მოდულს გააჩნია ინტერფეისების ნაკრები, რომელებიც უზრუნველყოფს მის შეუღლებას, როგორც საბორტო კომპიუტერის მოდულებთან ისე გარე მოწყობილობებთან. მონაცემების გაცვლა საბორტო კომპიუტერის მოდულებიდან ხორციელდება HD და ECD სალტეების საშუალებით. გარე მოწყობილობებთან შეუღლებისათვის მოდულს გააჩნია ორი სლოტი PCI სალტის ინტერფეისით,

რომლებშიც შესაძლებელია დაყენდეს სტანდარტული PMC - მეზონინები აუცილებელი ინტერფეისებით. ჩამონათვალის გარდა, მონაცემების პროცესორის მოდული შეიცავს მონაცემების გაცვლის ორ თანმიმდევრობით არხს, რომლებსაც გააჩნიათ ელექტრული ინტერფეისები RS-232, ასევე დისკრეტული სიგნალების ნაკრები, რომლებიც გამოიყენებიან როგორც მართვის ბრძანებები.

მონაცემების მიღებისა და ბუფერიზაციის მოდული

მონაცემების მიღებისა და ბუფერიზაციის მოდულის დანიშნულებაა ანალოგურ-ციფრული გარდამქმნელებიდან მიღებული ინფორმაციის დამუშავება.

მოდული ასრულებს შემდეგ ფუნქციებს:

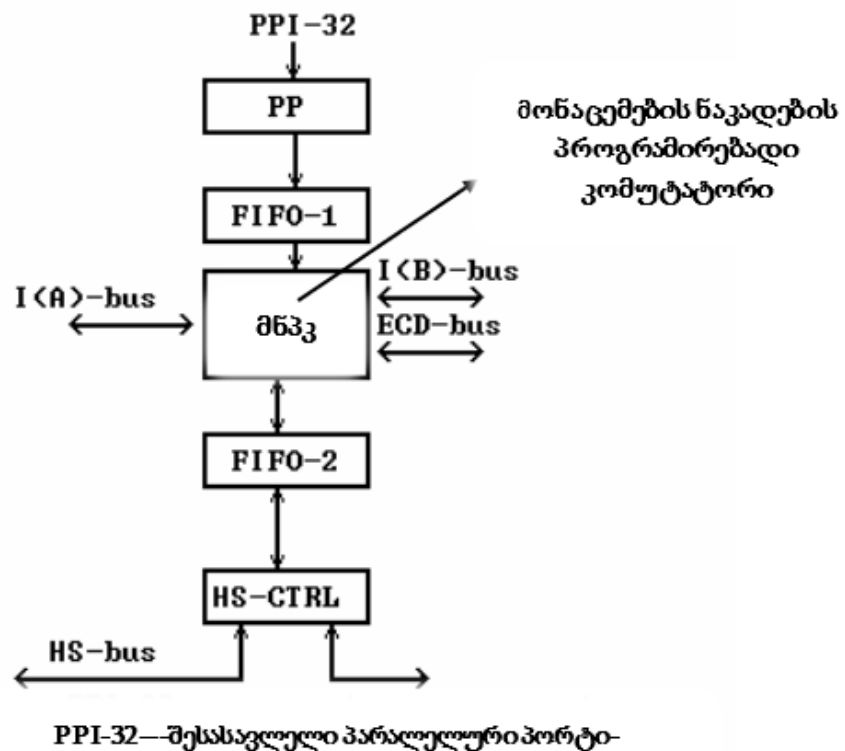
- გარე წყაროებიდან ინფორმაციის მიღება და ბუფერიზაცია;
- მონაცემების შემავალ ნაკადისთვის იდენტიფიკატორის მინიჭება;
- სხვადასხვა იდენტიფიკატორების მქონე ნაკადების პროგრამული კომუტაცია;
- მოცემული მონაცემების ბუფერიდან ინფორმაციის გაცემა მომდევნოში (მეზონინური ფირფიტა);
- შემავალი არხების მუშაობაში უწყესივრობების გამოვლენა და მათ მიხედვით მოვლენის ფორმირება ECD სალტეზე;
- ECD სალტეზე „საწყისი კადრის“ სიგნალის მიწოდებისას მოვლენის პროგრამულად შენიღბული ფორმირება.

მონაცემების ბუფერის მოდულის სტრუქტურული სქემა მოყვანილია ნახ.13.9-ზე.

მონაცემების ბუფერის მოდული შეიცავს ერთ 32 თანრიგიან პარალელურ პორტს (PP-32), რომელიც მუშაობს მხოლოდ ანალოგურ-ციფრული გარდამქმნელებიდან მონაცემების მიღებაზე. მოდული შეიცავს შემდეგ ფუნქციურ ბლოკებს:

- შესასვლელი ინტერფეისი, პარალელური პორტი;
- გამოსასვლელი ინტერფეისი გამოსასვლელი პორტის მეზონინური ფირფიტისათვის;
- ბუფერული მეხსიერება FIFO-1 შესასვლელზე ანალოგურ-ციფრული გარდამქმნელებიდან 8 კილო 32 თანრიგიანი სიტყვის სიღრმით;

- მონაცემების ნაკადის პროგრამირებადი კომპუტატორი (მნკკ) SWITCH;
- ბუფერულ მეხსიერება FIFO-2 HS - სალტიდან შესასავლელზე 8 კილო 32 თანრიგიანი სიტყვის სიღრმით;
- HS სალტის კონტროლერი (HS- CTRL).



ნახ. 13.9. მონაცემების მიღებისა და ბუფერიზაციის მოდულის სტრუქტურული სქემა

მოდულს გააჩნია მრავალსაღებო (მრავალარხიანი) შიგა არქიტექტურა, რომელიც იძლევა საშუალებას მოახდინოს სალტებს შორის კომპუტირება, რითაც უზრუნველყოფს სწორედ მონაცემთა ნაკადების სხვადასხვაგვარ განაწილებას.

გარე გამოსასვლელი არხის მაქსიმალური გატარების უნარი შეადგენს 20 მილიონ 32 თანრიგიან სიტყვას წამში (40 მეგაპერცი სიხშირის დროს).

გამოსასვლელი არხის მონაცემების წყაროებად შეიძლება იყოს ნებისმიერი არხი, რომლებიც შეერთებული იქნება პროგრამირებად კომპუტატორთან.

ბაგეტ 55-ის ყველა ეს მახასიათებლები უზრუნველყოფს ერთ და ორგანოზომილებიანი სიგნალების ეფექტურ ციფრულ დამუშავებას.

13.3.6. მმართველი საბორტო კომპიუტერი ბაგეტი-83

კომპიუტერი ბაგეტი-83 უკორპუსოა და ის თავსდება სხვა მოწყობილობებთან ერთად ერთ საერთო კორპუსში.

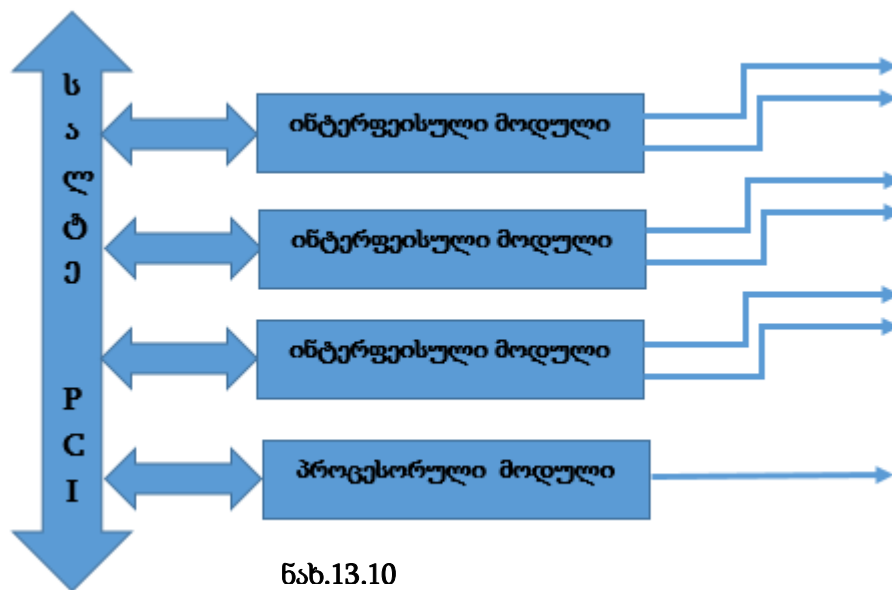
ჩაშენებადი საბორტო კომპიუტერი ბაგეტი-83 უზრუნველყოფს სხვადასხვა კონფიგურაციის გამოთვლითი სისტემების აგებას მოდულთშორისი გაცვლის PCI სალტის უნიფიცირებული მაგისტრალის საფუძველზე, ასევე უნიფიცირებული პროგრამული საშუალებებისა და ელექტრონული მოდულების ერთიანი ტიპზომების (103 x 96 x 16,5)მმ გამოყენების საფუძველზე.

ის გამოირჩევა მცირე გაზომვებით და ენერგომოხმარებით, საკმარისად მაღალი მწარმოებლურობისა და უნივერსალურობის თვისებებით. კომპიუტერს გააჩნია ღია არქიტექტურა და შეუძლია ადაპტირება მოახდინოს კონკრეტული ამოცანების მიმართ დამატებითი ელექტრონული მოდულების გამოყენების ხარჯზე.

კონსტრუქცია PC/104-Plus PCI Only, რომელიც გამოიყენება ბაგეტი-83 ტიპის კომპიუტერში, საშუალებას იძლევა დაემატოს მას ოთხი დამატებითი ინტერფეისული მოდული PCI სალტზე.

ჩაშენებული მცირეგაზომიანი კომპიუტერი წარმოადგენს აპარატულ-პროგრამულ კომპლექსს, რომელიც შედგება ერთი პროცესორული მოდულისგან (BT83-201) და დამატებითი ელექტრონული მოდულებისგან PC/104-Plus კონსტრუქტივში და აღჭურვილია საბაზო პროგრამული უზრუნველყოფით. კონსტრუქციულად კომპიუტერი შესრულებულია უკორპუსოდ და გათვალისწინებულია საბოლოო ნაკეთობის კორპუსზე დამაგრებისთვის, ან სპეციალურ კორპუსში დამაგრებისთვის, რომელიც დაპროექტებულია კონკრეტული გამოყენებისთვის. მოდულების შეერთებისთვის

არსებობს სამონტაჟო კომპლექტი. მცირეგაბარიტიანი ჩაშენებადი კომპიუტერის ბაგეტი 83-ის ტიპური სტრუქტურა მოყვანილია ნახ.13.10-ზე.



კომპიუტერი ბაგეტი 83 გამოიყენება ექსპლუატაციის შემდეგ პირობებში:

- ვიბრაცია 6g (20-2000 ჰერცი);
- ერთეული დარტყმა 500g;
- მრავალჯერადი დარტყმები 20g;
- ხაზოვანი აჩქარება 20g;
- მუშა ტემპერატურა -55+80° C – მდე;
- ტენიანობა 100% - მდე + 35° C - ის დროს;
- დადაბლებული მუშა წნევა 5 მილიმეტრი ვერცხლის სვეტისა.

13.4. საავიაციო საბორტო კომპიუტერი ტაიბერი

საავიაციო საბორტო კომპიუტერი ტაიბერი (სურ. 13.11) განკუთვნილია შემდეგი ფუნქციების შესასრულებლად:

- საჰაერო ხომალდის ბორტზე მართვის სტანდარტული პანელის შეცვლა;
- საჰაერო ხომალდის მართვაზე დაშვების შეზღუდვის განხორციელების შესაძლებლობა ავტორიზაციის (პაროლის) საშუალებით;
- საჰაერო ხომალდის ძალური ავტომატიკის სრული კონტროლის უზრუნველყოფა;
- საჰაერო ხომალდზე არსებული ყველა სისტემის დიაგნოსტიკის შესაძლებლობა;
- ხელსაწყოებისა და სისტემების არჩევის საშუალება და მათთან მუშაობა;
- მართვის მიწისზედა სადგურის გარეშე საფრენოსნო დავალების შეტანა;
- ავტოპილოტისა და სხვა ავარიული სისტემების კვების წყაროების დარეზერვების უზრუნველყოფა;
- ფრენისას სასარგებლო დატვირთვის მუშაობის უზრუნველყოფა, მონაცემების ჩაწერა და დამუშავება;
- საბორტო კომპიუტერის, როგორც დამხმარე სანავიგაციო მოწყობილობის გამოყენების შესაძლებლობა, სამოქალაქო ავიაციის ხომალდებისთვის ავტოპილოტის როლში გამოყენება.

ძირითადი პარამეტრები

- კვება 11-36 ვოლტი;
- მოხმარებული სიმძლავრე 250 ვატი/ 24 ვოლტი.

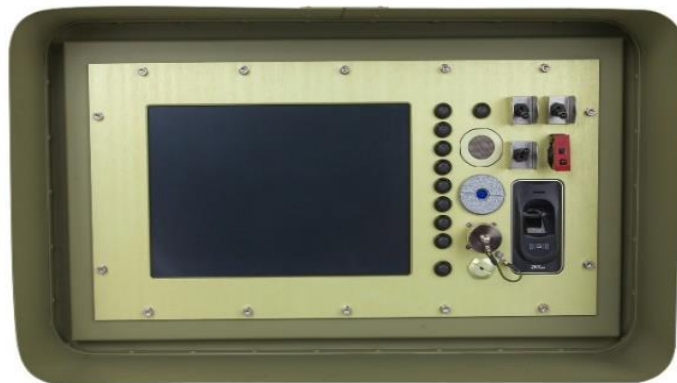
ინტერფეისები

- ინტერფეისი RS 232;

- ინტერფეისი RS 485;
- CAN;
- Ethernet.

საავიაციო საბორტო კომპიუტერი ტაიბერი უზრუნველყოფს:

- CAN ინტერფეისის მიხედვით ყველა სენსორის მიერთებას, დიაგნოსტიკას და მდგომარეობის ვიზუალიზაციას;
- CAN ინტერფეისის მიხედვით საჰაერო ხომალდის ყველა შემსრულებელი მექანიზმების მიერთება, დიაგნოსტიკა, მართვა და მდგომარეობის ვიზუალიზაცია;
- CAN ინტერფეისის მიხედვით მიმღებ-გადამცემი აპარატების მიერთება, დიაგნოსტიკა და მდგომარეობის ვიზუალიზაცია;
- გადარჩენის ბალისტიკური სისტემის დიაგნოსტიკა.



სურ.13.11

ლექცია 14

საბორტო კომპიუტერების და სისტემების საიმედოობა

14.1. საიმედოობის ძირითადი ტერმინები და განსაზღვრებები

საიმედოობა არის საბორტო კომპიუტერების და სისტემების თვისება შეასრულოს დაკისრებული ფუნქციები ექსპლუატაციის მოცემულ პირობებსა და განსაზღვრული დროის განმავლობაში ისე, რომ ძირითადი პარამეტრების მნიშვნელობები შენარჩუნებულ იქნეს დადგენილ ფარგლებში.

საბორტო კომპიუტერების და სისტემების საიმედოობა არ შეიძლება გაზომილი იქნეს უშუალოდ ისე, როგორც ეს შესაძლებელია ნებისმიერი ფიზიკური სიდიდისათვის. ის შესაძლებელია მხოლოდ შეფასებული ან ნავარაუდევო იქნეს რაოდენობრივად. საიმედოობის ძირითადი მაჩვენებლების შესაფასებლად იყენებენ ალბათობის თეორიის მათემატიკურ აპარატს.

საიმედოობის მახასიათებლების (მაჩვენებლების) რაოდენობრივი შეფასებისათვის აუცილებელია სტატისტიკური მასალა და ანალიზი, რომელიც მიიღება ცდის (დაკვირვების, გამოცდის) შედეგად.

საბორტო კომპიუტერების და სისტემების საიმედოობის შეფასების დროს გამოკვლევის ძირითად ობიექტებს წარმოადგენენ შემთხვევითი სიდიდეები და შემთხვევითი მოვლენები.

შემთხვევითი სიდიდე შეიძლება იყოს დისკრეტული ან მუდმივი. სიდიდეს, რომლის მნიშვნელობა ყოველ ცალკეულ ცდაში შემთხვევით იცვლება შემთხვევითი, ანუ სტოქასტური სიდიდე ეწოდება. შემთხვევითი სიდიდისათვის არ შეიძლება ზუსტად

იქნეს მითითებული, თუ რა კონკრეტულ მნიშვნელობას მიიღებს იგი ამა თუ იმ პირობებში. შემთხვევითი სიდიდის ამომწურავ მახასიათებელს მისი განაწილების კანონი წარმოადგენს. იგი მოცემულად ითვლება, თუ:

ა) მითითებულია შემთხვევითი სიდიდის შესაძლო მნიშვნელობათა სიმრავლე;

ბ) დადგენლია ამ სიმრავლის ნებისმიერ უბანზე შემთხვევითი სიდიდის მოხვედრის ალბათობათა განსაზღვრის ხერხი.

როგორც წესი, აღნიშნული გამოთვლები შემთხვევითი სიდიდის განაწილების ფუნქციის საშუალებით წარმოებს. შემთხვევითი სიდიდის განაწილების ფუნქცია ეწოდება $F(t)$ ფუნქციას, რომელიც $\tau < t$ ხდომილობის ალბათობას იძლევა

$$F(t) = P\{\tau < t\}.$$

მისი ძირითადი თვისებებიდან შესაძლებელია გამოიყოს შემდეგი ორი თვისება: ა) $F(t)$ ფუნქცია არაკლებადია, ე.ი. თუ $t_2 > t_1$, მაშინ

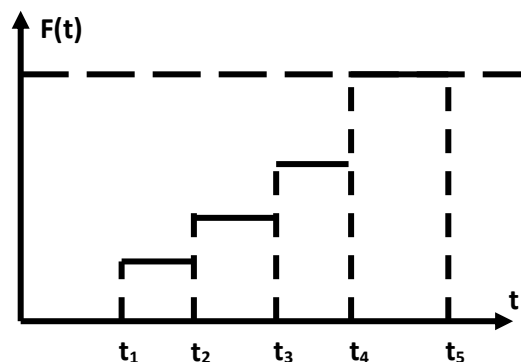
$$F(t_2) \geq F(t_1);$$

ბ) ნებისმიერი t -სათვის განაწილების $F(t)$ ფუნქცია აყმაყოფილებს შემდეგ უტოლობას

$$0 \leq F(t) \leq 1.$$

შემთხვევით სიდიდეს ეწოდება წყვეტილი ანუ დისკრეტული, თუ მას შეუძლია მიიღოს მნიშვნელობათა მხოლოდ სასრული ან თვლადი $t_1, t_2, t_3, \dots$ სიმრავლე.

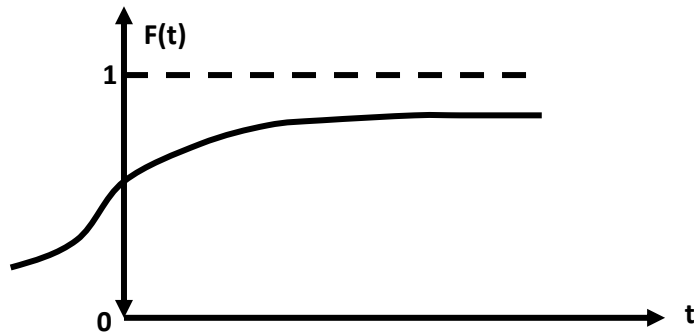
დისკრეტული შემთხვევითი სიდიდის განაწილების ფუნქცია ხშირ შემთხვევაში კიბისებურ ფუნქციას წარმოადგენს ნახტომებით $t_1, t_2, t_3, \dots$ წერტილებში (იხ. ნახ 14.1).



ნახ.14.1. დისკრეტული შემთხვევითი

სიდიდის განაწილების ფუნქციის გრაფიკი

შემთხვევით სიდიდეს ეწოდება უწყვეტი, თუ მას რომელიმე ღია ან ჩაკეტილი ინტერვალთან შეუძლია ნებისმიერი მნიშვნელობის მიღება. უწყვეტი შემთხვევითი სიდიდის განაწილების ფუნქცია უწყვეტ ფუნქციას წარმოადგენს (ნახ.14.2).



ნახ. 14.2. უწყვეტი შემთხვევითი სიდიდის განაწილების ფუნქცია

საბოლოო კომპიუტერების და სისტემების ტექნიკური მოწყობილობის ერთი და იგივე პარამეტრის მრავალჯერადი გაზომვის შედეგად მიიღება დისკრეტული შემთხვევითი სიდიდეების გარკვეული რიგი. ხოლო საბოლოო კომპიუტერის მუშაობის დრო მტყუნებამდე ან დრო, რომელიც საჭიროა მისი მუშაობისუნარიანობის აღსადგენად, წარმოადგენს უწყვეტი შემთხვევითი სიდიდის მაგალითს.

ყველა მნიშვნელობათა ალბათობას, რომელიც შესაძლებელია მიიღოს შემთხვევითმა სიდიდემ უწოდებენ ალბათობის განაწილებას. შემთხვევითი სიდიდეების განაწილების ალბათობებში შეინიშნება განსაზღვრული კანონზომიერებანი, რომლებიც შემთხვევითი სიდიდეების განაწილების კანონებად იწოდებიან.

შემთხვევითი მოვლენა ეწოდება ისეთ მოვლენას, რომელიც ცდის შედეგად შეიძლება მოხდეს ან არ მოხდეს. მაგალითად: მტყუნება, აღდგენა, ელექტრული წრედების მოკლედ ჩართვა, დიულექტრიკების გარღვევა წარმოადგენს შემთხვევით მოვლენებს.

შემთხვევითი მოვლენები, რომლებსაც ადგილი აქვთ დროში და ქმნიან გარკვეულ თანმიმდევრობას, იწოდებიან მოვლენათა ნაკადად (მაგალითად აღდგენადი მოწყობილობების მტყუნება).

საბორტო კომპიუტერების და სისტემების ერთი მდგომარეობიდან მეორე მდგომარეობაში გადასვლა საიმედოობის თეორიაში განიხილება, როგორც პროცესი (მაგალითად: მოწყობილობის მტყუნებელი მდგომარეობიდან ფუნქციონირების რეჟიმში გადასვლა).

საიმედოობის თეორიაში დიდი გავრცელება ჰპოვა შემთხვევითი სიდიდეების განაწილების შემდეგმა კანონებმა:

- 1) ბინომალური და პუასონის განაწილების კანონებმა დისკრეტული შემთხვევითი სიდიდეების დროს;
- 2) ექსპონენციალური, ვეიბულის, ნორმალური და ლოგარითმული განაწილების კანონებმა უწყვეტი შემთხვევითი სიდიდეების შემთხვევაში.

საბორტო კომპიუტერების და სისტემების საიმედოობის არსი ძირითადად უმტყუნობის, ხანგამძლეობის, სარემონტოდ ვარგისობის და შენახვისუნარიანობის ცნებათა ერთობლიობაში მჟღავნდება.

საბორტო კომპიუტერების და საბორტო სისტემების უმტყუნობა ეწოდება მის თვისებას შეინარჩუნოს მუშაობის უნარი (მუშაობისუნარიანობა) დროის გარკვეულ პერიოდში ექსპლუატაციის განსაზღვრულ პირობებში.

საბორტო კომპიუტერების მტყუნება კი წარმოადგენს მისი მუშაობისუნარიანობის (ფუნქციონირების) ნაწილობრივ ან სრულ დაკარგვას.

საბორტო კომპიუტერების და საერთოდ მრავალ მოწყობილობათა მიმართ მტყუნების ცნება ფარდობითია. მიუხედავად ამისა, იგი საშუალებას იძლევა შემოვიღოთ საიმედოობის სხვადასხვა რიცხობრივი მაჩასიათებლები და მათ საფუძველზე საბორტო კომპიუტერის და სისტემების საიმედოობის გარკვეული შედარებები ვაწარმოოთ.

არსებული კლასიფიკაციის მიხედვით მტყუნება შეიძლება იყოს უეცარი (კატასტროფული) და თანდათანობითი.

უეცარი (კატასტროფული) მყუნებები საბორტო კომპიუტერში (სისტემებში) აღიძვრებიან მასში შემავალი ელემენტების ერთი ან რამდენიმე პარამეტრის მყისურად შეცვლის შედეგად (წრედის მოკლედ ჩართვის ან მისი გაწყვეტის დროს). ასეთი სახის მტყუნების აღმოფხვრა ხორციელდება მწყობრიდან გამოსული ელემენტის (ბლოკის, მოწყობილობის) შესაბამისი ვარგისიანი ელემენტით შეცვლის ხარჯზე ან საბორტო კომპიუტერის ბლოკის (მოწყობილობის) შეკეთების ხარჯზე.

თანდათანობითი მტყუნებები აღიძვრება მხოლოდ მაშინ, როცა მოხდება საბორტო კომპიუტერში შემავალი ელემენტების რომელიმე პარამეტრის (ან პარამეტრების) თანდათანობითი ცვლილება იმ ზღვრამდე, როცა დაირღვევა ამ ელემენტის (ელემენტების) ნორმალური მუშაობის პირობები. ამ პარამეტრების დადგენილი საზღვრების გადაცდენა შეიძლება გამოწვეული იყოს ელემენტების დაძველებით, გარემომცველი გარემოს ტემპერატურის, წნევის, ტენიანობის ზემოქმედებით, რადიაციის დონით და მექანიკური ზემოქმედებით - ვიბრაციით, დარტყმებით, გადატვირთვებით. ასეთი სახის მტყუნების აღმოფხვრა შესაძლებელია ელემენტის შეცვლით, რემონტით (შეკეთებით) ან კიდევ ამ ელემენტის პარამეტრების რეგულირებით.

გარეგანი გამოვლინებების მიხედვით მტყუნებები იყოფიან აშკარა და არააშკარა მტყუნებებად.

აშკარა (ცხადი) მტყუნებები გამოიქვანდებიან საბორტო კომპიუტერების გარეგანი დათვალიერების დროს.

არაცხადი მტყუნების გამოვლენისათვის საჭიროა კონტროლის სპეციალური მეთოდების გამოყენება.

ხანგამძლეობა - საბორტო კომპიუტერების (საბორტო სისტემების) თვისებაა შეინარჩუნოს მუშაობისუნარიანობა ზღვრულ მდგომარეობამდე, ტექნიკური მომსახურებისა და შეკეთებისათვის აუცილებელი შესვენებების გათვალისწინებით, ე.ი. ეს არის მისი თვისება ხანგრძლივად შეინარჩუნოს მუშაობის უნარი ექსპლუატაციის გარკვეული რეჟიმისა და საჭირო ტექნიკური მომსახურების პირობებში. იმ ვადის მიწურულში, რომელიც ხანგამძლეობას განსაზღვრავს საბორტო კომპიუტერებში წარმოიშობა დაძველებასთან („სიბერესთან“) და გაცვეთასთან დაკავშირებული პროცესები. მათი აცილება ან შეუძლებელია ან კიდევ ეკონომიურად გაუმართლებელი. ხანგამძლეობის რაოდენობრივი მაჩვენებლებია დრო, ციკლთა რიცხვი, შესრულებული სამუშაოს მოცულობა.

საბორტო კომპიუტერების სარემონტოდ ვარგისობა ეწოდება მის მოხერხებულობას აღდგენისადმი, მტყუნების აცილებისა და გამოვლინებისადმი. უფრო ფართო განმარტებით სარემონტოდ ვარგისობა ეს არის საბორტო კომპიუტერების თვისება, რომელიც გულისხმობს მტყუნებების აცილების, აღმოჩენისა და აღმოფხვრისადმი, ასევე

სხვა სახის შეფერხებისადმი გამოვლენის მოხერხებულობას ჩატარებული ტექნიკური მომსახურებისა და შეკეთებების საშუალებით.

შენახულობა - ეს არის საბორტო კომპიუტერების (სისტემის) თვისება შეინარჩუნოს საექსპლუატაციო მაჩვენებლები როგორც მოცემული შესანახი ვადის განმავლობაში, ისე მისი გასვლის შემდეგაც. ეს თვისება ახასიათებს საბორტო კომპიუტერს უსაფრთხოებას შენახვის რეჟიმში.

14.2. არააღდგენადი საბორტო კომპიუტერების საიმედოობის ძირითადი მაჩვენებლები

არააღდგენად საბორტო კომპიუტერებს მიეკუთვნებიან ისეთი კომპიუტერები, რომლებიც მუშაობენ მხოლოდ პირველ მტყუნებამდე. ასეთ კომპიუტერებში არ ხდება მტყუნებული ელემენტის ან მოწყობილობის შეცვლა ახალი ელემენტით ან მოწყობილობით, რაც იწვევს კომპიუტერების ფუნქციონირების (მუშაობის) შეწყვეტას.

საბორტო არააღდგენადი კომპიუტერების საიმედოობის ძირითადი მაჩვენებლებია: უმტყუნო მუშაობის ალბათობა $P(t)$, მტყუნების ალბათობა $q(t)$, მტყუნებამდე ნამუშევარის დროითი განაწილების სიმკვრივე $f(t)$, მტყუნებათა ინტენსიურობა (საფრთხე) $\lambda(t)$, საშუალო ნამუშევარი პირველ მტყუნებამდე $T_{სშ}$.

უმტყუნო მუშაობის ალბათობა $P(t)$ წარმოადგენს არააღდგენადი საბორტო კომპიუტერის საიმედოობის ძირითად რაოდენობრივ მახასიათებელს. ის გამოხატავს იმის ალბათობას, რომ კომპიუტერში მოცემული t დროის მონაკვეთზე არ მოხდება მტყუნება. ეს იმას ნიშნავს, რომ საბორტო კომპიუტერის პირველ მტყუნებამდე ნამუშევარის განმსაზღვრელი შემთხვევითი τ დრო ტოლი ან მეტი იქნება წინასწარ მოცემულ t დროის ინტერვალზე:

$$P(t) = P\{\tau \geq t\}. \quad (14.1)$$

ამ ფუნქციას ხშირად საიმედოობის ფუნქციასაც უწოდებენ.

დავუშვათ, რომ $t = 0$ დროის მომენტში არააღდგენადი საბორტო კომპიუტერი მუშაობას იწყებს, ხოლო $t = \tau$ მომენტისათვის ის მტყუნებას განიცდის. თუ τ ნაკლები იქნება წინასწარ მოცემული დროის ინტერვალზე t -ზე, მაშინ მისი τ -ს, როგორც შემთხვევითი

სიდიდის ყველაზე სრულ დახასიათებას იძლევა განაწილების შემდეგი ფუნქცია:

$$q(t) = P\{\tau < t\}. \quad (14.2)$$

(14.2) ფორმულა გამოსახავს იმის ალბათობას, რომ საბოლოო კომპიუტერი განიცდის მტყუნებას t მომენტამდე. $q(t)$ -ს მტყუნების ალბათობა ეწოდება.

მოცემული t დროის ინტერვალის განმავლობაში შესაძლებელია მოხდეს ორი ურთიერთსაპირისპირო მოვლენიდან მხოლოდ ერთი, კერძოდ, კომპიუტერი იმუშავებს უმტყუნებოდ ან ადგილი იქნება მის მტყუნებას. ამიტომ ცხადია, რომ სამართლიანია დამოკიდებულება:

$$P(t) + q(t) = 1, \quad (14.3)$$

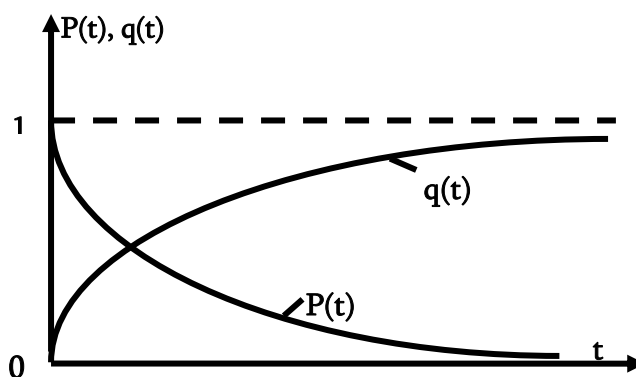
საიდანაც

$$P(t) = 1 - q(t). \quad (14.4)$$

უმტყუნებო მუშაობის ალბათობის $P(t)$ და მტყუნების ალბათობის $q(t)$ ფუნქციების ზოგადი სახე ნახ. 14.3-ზეა ნაჩვენები. ნახაზიდან ჩანს, რომ $q(t)$ მონოტონურად იზრდება. ამასთან $q(0) = 0$ და $q(t) \rightarrow 1$, თუ $t \rightarrow \infty$. $P(t)$ კი პირიქით, მონოტონურად მცირდება, $P(0) = 1$ და $P(t) \rightarrow 0$, როცა $t \rightarrow \infty$.

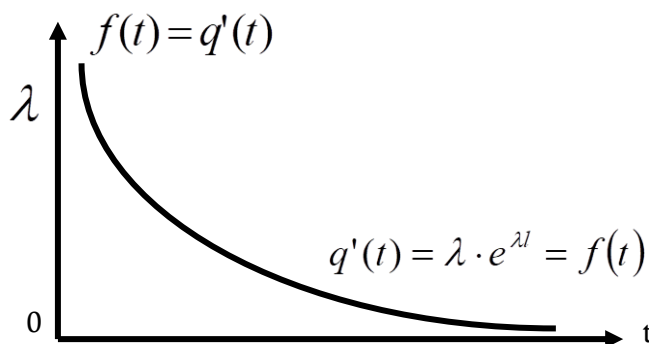
მტყუნებამდე ნამუშევარის დროითი განაწილების სიმკვრივე

საიმედოობის თეორიაში ხშირად უშვებენ, რომ $q(t)$ ფუნქცია უწყვეტია, ისევე როგორც მისი პირველი წარმოებული:



ნახ.14.3. უმტყუნო მუშაობის ალბათობისა და მტყუნების ალბათობის გრაფიკები

$f(t) = q'(t)$, რომელიც მტყუნებამდე ნამუშევარის (მტყუნების ალბათობის) სიმკვრივეს წარმოადგენს.



ნახ.14.4. მტყუნებამდე ნამუშევარის დროითი განაწილების სიმკვრივის გრაფიკული ილუსტრაცია

უმტყუნო მუშაობის ალბათობის ექსპერიმენტული განსაზღვრა

უმტყუნო მუშაობის ალბათობის ექსპერიმენტული განსაზღვრა შემდეგნაირად ხდება: ერთნაირ პირობებში გამოსაცდელად აყენებენ N ერთნაირ ელემენტს (ელემენტის ცნების ქვეშ იგულისხმება ნებისმიერი ნაკეთობა, რომლის საიმედოობა შეისწავლება მისი შემადგენელი ნაწილების საიმედოობისაგან დამოუკიდებლად) და აღნიშნავენ ყოველი ელემენტის მტყუნების მომენტს. ამის შედეგად პოულობენ $N(t)$ ფუნქციას, რომელიც დროის t მომენტისათვის უმტყუნოდ მომუშავე ელემენტების რაოდენობას წარმოადგენს. ცხადია, რომ საწყისი მომენტისათვის $N(0) = N$, ხოლო შემდეგ ყოველი მტყუნების მომენტში იგი ერთი ერთეულით მცირდება.

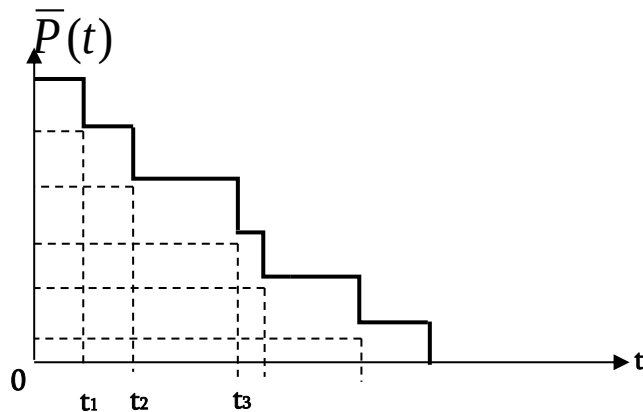
ფარდობა:

$$\bar{P}(t) = \frac{N(t)}{N} \quad (14.5)$$

ელემენტის საიმედოობის $P(t)$ ფუნქციის სტატისტიკურ შეფასებას წარმოადგენს. მას საიმედოობის ემპირიული ფუნქცია ეწოდება. როცა $N \rightarrow \infty$,

$$\bar{P}(t) = \frac{N(t)}{N} = P(t) \quad (14.6)$$

საიმედოობის ემპირიული ფუნქციის ზოგადი სახე ნახ.14.5-ზეა წარმოდგენილი.



ნახ.14.5. საიმედოობის ემპირიული ფუნქციის ზოგადი სახე

მტყუნების ინტენსიურობა (საფრთხე) ეწოდება $\lambda(t)$ ფუნქციას, რომელიც განისაზღვრება მტყუნების ალბათობის სიმკვრივის $f(t)$ -ს ფარდობით ელემენტის უმტყუნო მუშაობის ალბათობასთან $P(t)$ -სთან

$$\lambda(t) = \frac{f(t)}{P(t)} = \frac{q'(t)}{P(t)}. \quad (14.7)$$

თუ ცნობილი იქნება მტყუნებათა ინტენსიურობა λ_t , მაშინ მისი საშუალებით შეიძლება გამოსახულ იქნეს უმტყუნო მუშაობის ალბათობა $P(t)$.

$$\text{რადგან } f(t) = q'(t) = (1 - P(t))' = -P'(t),$$

ე.ი. $f(t) = -P'(t)$, ამიტომ (14.7) ფორმულა შეგვიძლია ჩავწეროთ ასეთი სახით:

$$\lambda(t) = -\frac{P'(t)}{P(t)} . \quad (14.8)$$

რადგან $\lambda(t) = -\frac{P'(t)}{P(t)} = -[\ln P(t)]'$, ამ გამოსახულების ინტეგრირების შემდეგ

მივიღებთ:

$$\ln P(t) = -\int_0^t \lambda(t) dt ,$$

საიდანაც

$$p(t) = e^{-\int_0^t \lambda(t) dt} . \quad (14.9)$$

ამრიგად, უმტყუნო მუშაობის ალბათობა გამოისახება მტყუნების ინტენსიურობის საშუალებით.

მტყუნების ინტენსიურობის ექსპერიმენტული განსაზღვრა

$\lambda(t)$ -ს ექსპერიმენტული განსაზღვრისათვის გამოსაცდელად აყენებენ N ელემენტს და მტყუნებებზე დაკვირვებებს აწარმოებენ. თუ Δt საკმაოდ მცირეა, ხოლო N საკმაოდ დიდი, სტატისტიკური შეფასება $\lambda(t)$ -თვის ასეთი სახით შეიძლება ჩაიწეროს

$$\lambda(t) = \frac{P(t) - P(t + \Delta t)}{P(t)\Delta t} \approx \frac{N(t) - N(t + \Delta t)}{N} : \frac{N(t)}{N} \cdot \Delta t \approx \frac{n(t, t + \Delta t)}{N(t) \cdot \Delta t} ,$$

$$\text{ე.ი. } \lambda(t) = \frac{n(t, t + \Delta t)}{N(t) \cdot \Delta t} . \quad (14.10)$$

(14.10) ფორმულაში $N(t)$ იმ ელემენტთა რიცხვია, რომლებიც დროის t მომენტისათვის განაგრძობენ მუშაობას, $n(t, t + \Delta t)$ კი მტყუნებათა რაოდენობაა დროის t მომენტიდან $t + \Delta t$ მომენტამდე.

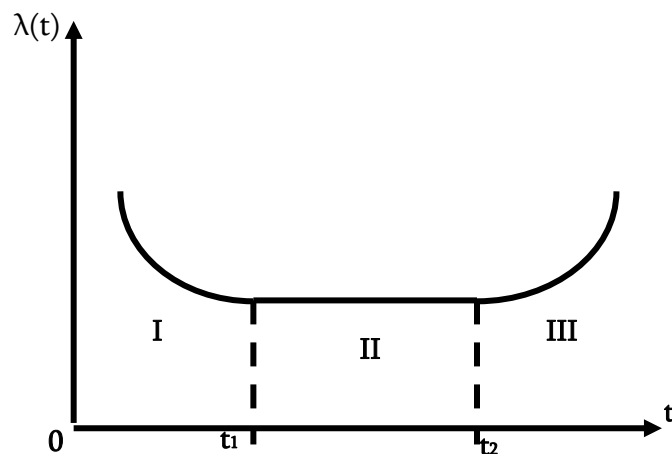
(14.10) ფორმულა შეიძლება ასეთნაირადაც ჩაიწეროს

$$\lambda(t) = \frac{\Delta n_i}{(N - n_i) \cdot \Delta t_i} , \quad (14.11)$$

სადაც Δn_i - Δt დროის მონაკვეთში მტყუნებული ელემენტების რაოდენობაა. N - ელემენტების საწყისი რაოდენობაა, n_i - მტყუნებული ელემენტების საერთო რიცხვია განსახილველი დროითი Δt მონაკვეთის დასაწყისში.

λ -ს განზომილების ერთეულია სთ^{-1} .

ექსპერიმენტული მონაცემებით ელემენტთა უმრავლესობისათვის $\lambda(t)$ ფუნქციას ნახ.14.6-ზე წარმოდგენილი მრუდის სახე აქვს. გრაფიკზე მკვეთრად გამოირჩევა დროითი ინტერვალების სამი უბანი. დროითი ინტერვალის პირველ უბანზე $(0 \div t_1)$ $\lambda(t)$ ფუნქციის მნიშვნელობები საკმაოდ დიდია, რაც იმით არის გამოწვეული, რომ მრავალ ელემენტს ფარული ტექნოლოგიური და საწარმოო დეფექტები გააჩნიათ და ისინი მათი მუშაობის დაწყებისთანავე ვლინდებიან. დროითი ინტერვალის $(0 \div t_1)$ უბანს დამუშავების პერიოდს უწოდებენ.



ნახ. 14.6. მტყუნების ინტენსიურობის ცვლილება დროში. I - დამუშავების პერიოდი; II - ნორმალური ექსპლუატაციის პერიოდი; III - დაძველების პერიოდი

ინტერვალის II უბანი $(t_1 \div t_2)$ ნორმალური მუშაობის პერიოდს შეესაბამება. იგი ხასიათდება მტყუნების ინტენსიურობის მუდმივობით, ე.ი. $\lambda(t) = \lambda = \text{const.}$ დროითი ინტერვალის III უბანი $(t_2 \div \infty)$ კი დაძველების („დაბერების“) პერიოდს წარმოადგენს. ამ უბანზე მტყუნების ინტენსიურობა დროის ზრდად ფუნქციას წარმოადგენს.

პირველი უზნისათვის $(0 \div t_1)$, როგორც არ უნდა იყოს Δt , სამართლიანია უტოლობა

$$\lambda(t + \Delta t) < \lambda(t),$$

ამიტომ შეგვიძლია გავაკეთოთ შემდეგი დასკვნა: რაც უფრო მეტი დრო აქვს ნამუშევარი ელემენტს წარსულში (ე.ი. რაც უფრო მეტია t), მით უფრო მაღალია მისი საიმედოობა დამუშავების პერიოდში.

დროითი ინტერვალის მეორე უზნისათვის $(t_1 \div t_2)$ $\lambda(t + \Delta t) = \lambda(t)$, ამ შემთხვევაში ელემენტის საიმედოობა მომავალში სრულებით არ არის დამოკიდებული იმაზე, თუ რამდენ ხანს იმუშავა მან წარსულში, ე.ი. $\lambda(t + \Delta t) = \lambda(t) = \lambda = const$.

ჩვენთვის ყველაზე მეტად საინტერესოა საბორტო კომპიუტერების (სისტემების, ელემენტების) ფუნქციონირების პროცესი სწორედ ნორმალური მუშაობის პერიოდში (II უბანი). განვსაზღვროთ ამ შემთხვევისათვის საიმედოობის მაჩვენებლები.

როგორც უკვე ვნახეთ, არააღდგენადი საბორტო კომპიუტერის უმტყუნო მუშაობის ალბათობასა $p(t)$ და მტყუნების ინტენსიურობას λ -ს შორის არსებობს შემდეგი დამოკიდებულება:

$$p(t) = e^{-\int_0^t \lambda(t) dt}, \quad (14.12)$$

როცა $\lambda(t) = \lambda = const$, მაშინ აღნიშნული დამოკიდებულება მიიღებს სახეს

$$p(t) = e^{-\lambda t}, \quad (14.13)$$

და იგი გამოხატავს უმტყუნო მუშაობის ალბათობის $P(t)$ -ს ექსპონენციალური კანონით ცვლილებას მტყუნების ინტენსიურობისაგან. (14.13) დამოკიდებულებას ადგილი აქვს უეცარი მტყუნების დროს.

მტყუნების ალბათობა t დროის მონაკვეთზე $q(t)$ ტოლი იქნება

$$q(t) = 1 + p(t) = 1 - e^{-\lambda t},$$

$$\text{ე.ი. } q(t) = 1 - e^{-\lambda t} . \quad (14.14)$$

მტყუნების ალბათობის სიმკვრივე $f(t)$ ექსპონენციალური განაწილების კანონისათვის შემდეგი სახით ჩაიწერება:

$$f(t) = q'(t) = [1 - P(t)]' = -P'(t) = (-e^{-\lambda t})' = \lambda e^{-\lambda t}; \text{ ე.ი.}$$

$$q'(t) = \lambda \cdot e^{-\lambda t} . \quad (14.15)$$

(14.15)-ში t არის პირველ მტყუნებამდე ნამუშევარი დრო, λ კი სისტემის მტყუნების ინტენსიურობა დროის ერთეულში.

უმტყუნო მუშაობის საშუალო დრო (საშუალო ნამუშევარი პირველ მტყუნებამდე)

ელემენტის უმტყუნო მუშაობის საშუალო დრო გამოითვლება ფორმულით

$$T_{\text{საშ}} = \int_0^{\infty} t \cdot f(t) dt = \int_0^{\infty} t \cdot q'(t) dt = - \int_0^{\infty} t \cdot P'(t) dt$$

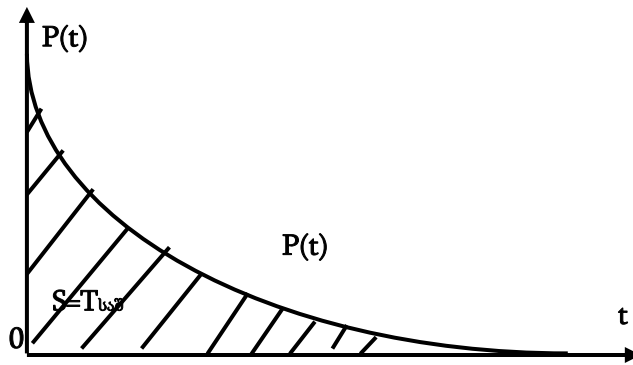
ამ გამოსახულების ინტეგრირება ნაწილების მიხედვით მოგვცემს

$$T_{\text{საშ}} = -t \cdot P(t) \Big|_0^{\infty} + \int_0^{\infty} P(t) dt$$

ამ გამოსახულების მარჯვენა ნაწილის პირველი შესაკრები ტოლია ნულის, რადგან t შემთხვევითი სიდიდისათვის, რომლისთვისაც არსებობს მათემატიკური ლოდინი $M(t)$, სხვაობა $1 - q(t) = P(t)$, როცა $t \rightarrow \infty$ უფრო სწრაფად მცირდება (ეცემა), ვიდრე იზრდება t . ამიტომ გვექნება

$$T_{\text{საშ}} = \int_0^{\infty} P(t) dt$$

ეს ფორმულა $T_{\text{საშ}}$ სიდიდის ნათელ გეომეტრიულ ინტერპრეტაციას იძლევა, რაც იმის მაჩვენებელია, რომ ელემენტის უმტყუნო მუშაობის საშუალო დრო რიცხობრივად უდრის იმ ფართობს, რომელიც კოორდინატთა ღერძებითა და $P(t)$ მრუდით არის შემოფარგლული (ნახ. 14.7).



ნახ. 14.7. უმტყუნო მუშაობის საშუალო დროის გეომეტრიული ინტერპრეტაცია

ექსპონენციალური კანონის დროს $P(t) = e^{-\lambda t}$ -ს, ამიტომ ამ შემთხვევაში მივიღებთ

$$T_{\text{საშ}} = \int_0^{\infty} P(t) dt = \int_0^{\infty} e^{-\lambda t} dt = -\frac{1}{\lambda} \cdot e^{-\lambda t} \Big|_0^{\infty} = \lim \left[\left(-\frac{1}{\lambda} e^{-\lambda(t)} \right) \right] - \left(-\frac{1}{\lambda} e^{-\lambda \cdot 0} \right) = 0 + \frac{1}{\lambda} \cdot e^0 = \frac{1}{\lambda}$$

ვიციტ, რომ $\lambda = \frac{1}{T_{\text{საშ}}}$, ამიტომ

$$p(t) = e^{-\lambda t} = e^{-\frac{t}{T_{\text{საშ}}}}$$

$$p(t) = e^{-\frac{t}{T_{\text{საშ}}}} \quad (7.2) \quad (14.16)$$

ჩვენ აღვნიშნეთ, რომ უმტყუნო მუშაობის ალბათობის ექსპონენციალური განაწილების შემთხვევაში მტყუნების ინტენსიურობა მუდმივია, ე.ი. $\lambda(t) = \lambda$. დავამტკიცოთ, რომ ეს ასეა:

$$\lambda(t) = -\frac{P'(t)}{P(t)} = \frac{q'(t)}{1-q(t)} = \frac{\lambda \cdot e^{-\lambda t}}{1-(1-e^{-\lambda t})} = \lambda,$$

$$\text{ე.ი. } \lambda(t) = \lambda = \text{const}.$$

მაგალითი 14.1. საბორტო გამოთვლითი სისტემა შედგება 5 ელემენტისგან. მათ მტყუნებათა ინტენსიურობები მოყვანილია ცხრილში 14.1.

ცხრილი 14.1

ელემენტის ნომერი	1	2	3	4	5
λ_i სთ ⁻¹	0,00007	0,00005	0,00004	0,00006	0,00004

განისაზღვროს საბორტო გამოთვლითი სისტემის საიმედოობის მაჩვენებლები: მტყუნების ინტენსიურობა, უმტყუნო მუშაობის საშუალო დრო, უმტყუნო მუშაობის ალბათობა, უმტყუნო მუშაობის დროის განაწილების სიმკვრივე. საიმედოობის მაჩვენებლები უმტყუნო მუშაობის ალბათობა $P(t)$ და უმტყუნო მუშაობის დროითი განაწილების სიმკვრივე მიღებულ იქნეს ინტერვალში 0-დან 1000 საათამდე 100 საათის ბიჯით.

ამოხსნა

საბორტო გამოვითვალთ სისტემის მტყუნების ინტენსიურობა λ_s და უმტყუნო მუშაობის საშუალო დრო $T_{სშ}$.

$$\lambda_b = \sum_{i=1}^n \lambda_i = 0,00007 + 0,00005 + 0,00004 + 0,00006 + 0,00004 = 0,00026 \text{ სთ}^{-1},$$

$$T_{სშ} = \frac{1}{\lambda_c} = \frac{1}{0,00026} = 3846 \text{ სთ}$$

საბორტო გამოთვლითი სისტემის უმტყუნო მუშაობის ალბათობა $P_s(t)$ და მტყუნებამდე დროითი განაწილების სიმკვრივე T_s დროით ინტერვალში 0-დან 1000მდე იქნება:

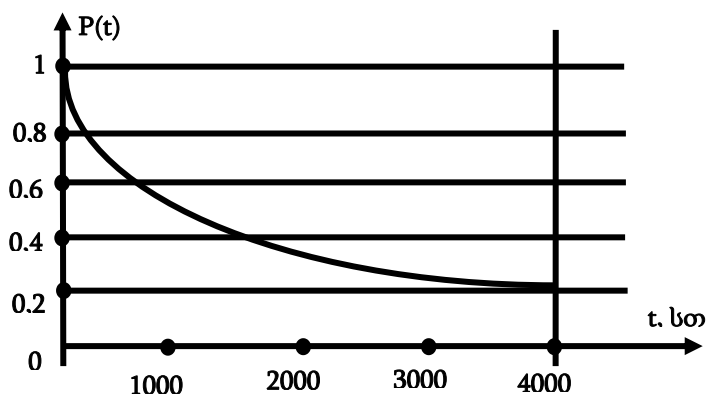
$$P_b(t) = e^{-\lambda_b t} = e^{-0,00026 t}$$

$$\tau_b(t) = \lambda_b \cdot e^{-\lambda_b t} = 0,00026 \cdot e^{-0,00026 \cdot 0,00026}$$

ტაბულირების შედეგები წარმოდგენილია ცხრილში 14.2

ცხრილი 14.2

$t, \text{სT}$	$P_s(t)$	$f_s(t)$
0	1	0,00026
100	0,974335	0,000253
200	0,949329	0,000247
300	0,924964	0,000240
400	0,901225	0,000234
500	0,878095	0,000228
600	0,855559	0,000222
	0,833601	0,000217
800	0,812207	0,000211
900	0,791362	0,000206
1000	0,771052	0,000200



ნახ. 14.8. სისტემის უმტყუნო მუშაობის ალბათობა

მაგალითი 14.2. ცნობილია, რომ არააღდგენად სისტემას გააჩნია მტყუნებამდე ნამუშევარის ექსპონენციალური განაწილება. საჭიროა განსაზღვრულ იქნეს სისტემის საშუალო ნამუშევარი $t_{საშ}$ მტყუნებამდე, იმ პირობით, რომ მისი უმტყუნო მუშაობის ალბათობა იყოს არანაკლები 0,995-ისა 500 საათის განმავლობაში.

ამოხსნა

საბოლოო გამოთვლითი სისტემის უმტყუნო მუშაობის ალბათობის გამოთვლისათვის ვისარგებლოთ ფორმულებით:

$$P(t) = e^{-t/t_{საშ}} \text{ და } t_{საშ} = \frac{1}{\lambda}$$

$$\text{მივიღებთ } p(t) = e^{-t/t_{საშ}} = e^{-500/t_{საშ}} \geq 0,995 \text{ ან}$$

$$- 500/t_{საშ} \geq \ln 0,995 = -0,0049,$$

საიდანაც მივიღებთ

$$t_{საშ} = \frac{500}{0,0049} \approx 1,02 \cdot 10^5 \text{ სთ.}$$

ახლა განვსაზღვროთ მიახლოებითი ფორმულით $p(t) = 1 - \lambda t$,

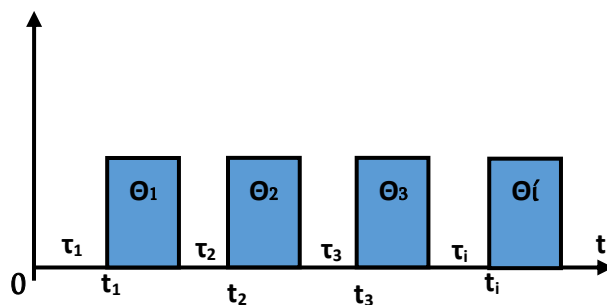
$$t_{საშ} = \frac{500}{0,005} = 10^5 \text{ სთ.}$$

მიღებული ამონახსნების შედეგების შედარებიდან ჩანს, რომ $t_{საშ}$ განსაზღვრისას მიახლოებითი ფორმულის გამოყენებით არსებული შეცდომა არ აღემატება 0,98%-ს.

14. 3. აღდგენადი საბორტო კომპიუტერების საიმედოობის ძირითადი მაჩვენებლები

საბორტო კომპიუტერები აღდგენადია, თუ მტყუნების შემდეგ აუცილებელი სარემონტო ღონისძიებათა ჩატარების შემდეგ მას შეუძლია განაახლოს მუშაობა.

როგორც წესი, საბორტო კომპიუტერების მტყუნების შემდეგ ხდება მისი ნორმალური ფუნქციონირების შეწყვეტა. მისი ფუნქციონირების აღსადგენად ხდება მისი რემონტი ან შეცვლა. აღდგენადი საბორტო კომპიუტერების მუშაობა ხასიათდება დროითი დიაგრამით, რომელიც მოცემულია ნახ.14.9 - ზე.



ნახ. 14.9. აღდგენადი საბორტო კომპიუტერების მუშაობის დროითი დიაგრამა.

$\tau_1, \tau_2 \dots \tau_i$ - საბორტო კომპიუტერის ფუნქციონირების ინტერვალები;

$t_1, t_2 \dots t_i$ - საბორტო კომპიუტერის მწყობრიდან გამოსვლის მომენტები;

$\theta_1, \theta_2 \dots \theta_i$ - საბორტო კომპიუტერის შეკეთებისა და აღდგენის ინტერვალები.

ბუნებრივია, რომ, $\tau_1, \tau_2 \dots \tau_i$, $\theta_1, \theta_2 \dots \theta_i$ შემთხვევითი სიდიდეებია.

აღდგენადი საბორტო კომპიუტერების სისტემები ხასიათდება საიმედოობის შემდეგი რაოდენობრივი მაჩვენებლებით: მტყუნებების ნაკადის პარამეტრით $\omega(t)$; აღდგენების ნაკადის პარამეტრით $\mu(t)$; ორ მტყუნებას შორის მუშაობის საშუალო

დროით $T_{სშ}$; აღდგენის საშუალო დროით $T_{აღ}$; მზადყოფნის კოეფიციენტით $K_{მზ}$; იძულებითი გაცდენის კოეფიციენტით $K_{გაც}$.

მტყუნებათა ნაკადის პარამეტრი $\omega(t)$

დავუშვათ, რომ რაღაც t დროის განმავლობაში თვალყურს ადევნებენ N ნიმუშების საბორტო კომპიუტერების მუშაობას (ნორმალური ექსპლუატაციის პირობებში) და აფიქსირებენ თითოეული ნიმუშის საბორტო კომპიუტერის მტყუნებების $m_i(t)$ რიცხვს. t_N დროის განმავლობაში მტყუნებების საშუალო რიცხვი გამოითვლება ფორმულით:

$$m_{საშ}(t) = \frac{1}{N} \sum_{i=1}^N m_i(t)$$

სადაც მტყუნებების რიცხვში შედის არა მარტო პირველ-საწყისი მტყუნებები, არამედ ასევე ის მტყუნებებიც, რომლებიც აღიძვრება აღდგენის შემდეგ ან მტყუნებული ელემენტების შეცვლის შემდეგაც. მტყუნებების წარმოშობა შეიძლება განხილულ იქნეს, როგორც მოთხოვნათა ნაკადი მომსახურებაზე (ამ შემთხვევაში აღდგენაზე). ამ ნაკადის მახასიათებელს განსაზღვრავენ როგორც

$$H(t) = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{i=1}^N m_i(t). \quad (14.17)$$

$H(t)$ ფუნქციის მნიშვნელობის მიხედვით გამოითვლება მტყუნებათა ნაკადის პარამეტრი. ამ გამოთვლებისათვის საჭირო განტოლებას აქვს სახე

$$\omega(t) = \frac{dH(t)}{dt}. \quad (14.18)$$

პრაქტიკაში იყენებენ სხვა განტოლებას, რომელიც საშუალებას იძლევა განსაზღვრულ იქნეს მტყუნებათა ნაკადის პარამეტრის მიახლოებითი მნიშვნელობა

$$\omega(t) = \frac{1}{N\Delta t} = \left[\sum_1^N m_i(t + \Delta t) - \sum_1^N m_i(t) \right], \quad (14.19)$$

სადაც $m_i(t)$ და $m_i(t + \Delta t)$ თითოეული ნიმუშის საბორტო კომპიუტერის მტყუნებათა რიცხვია, შესაბამისად t და $(t + \Delta t)$ დროის მომენტში. Δt - საკმარისად მცირე დროის $H(t)$ შუალედია.

საბორტო კომპიუტერებისთვის დამახასიათებელია ე.წ. დამუშავების პერიოდი, რომელიც მთავრდება დროის რაღაც t_0 მომენტისათვის. ამ შემთხვევაში მტყუნებათა ნაკადის მახასიათებელი ხდება ხაზოვანი და მრუდის განტოლება შეიძლება ჩაწერილ იქნეს შემდეგნაირად:

$$H(t) = H(t_0) + (t - t_0) \cdot \omega, \quad (14.20)$$

სადაც ω - მუდმივი სიდიდეა.

(14.18) ფორმულის საფუძველზე შეიძლება განსაზღვრულ იქნეს მტყუნებათა ნაკადის პარამეტრი

$$\omega(t) = d[H(t_0) + (t - t_0) \cdot \omega] / dt = \omega = const. \quad (14.21)$$

საბორტო კომპიუტერების მტყუნებათა ნაკადი ტოლია საბორტო კომპიუტერში შემავალი ცალკეული მოწყობილობების მტყუნებათა ნაკადების ჯამისა. თუ თითოეული ნაკადი ცალ-ცალკე ჯამურ ნაკადზე ახდენს თანაბარზომიერ და მცირე გავლენას, ასეთ შემთხვევაში ჯამური ნაკადი იქნება უმარტივესი.

უმარტივესი ნაკადი უნდა აკმაყოფილებდეს სტაციონარობის, მერმექმედების და ორდინარობის პირობებს.

ნაკადის სტაციონარობა ნიშნავს, რომ ზუსტად K რაოდენობის მტყუნების წარმოშობის ალბათობა დროის $t_0 \div (t + t_0)$ შუალედში არ არის დამოკიდებული t_0 მომენტზე და წარმოადგენს t და K ცვლადების ფუნქციას.

ნაკადის მერმექმედების არარსებობის თვისების ქვეშ იგულისხმება ის, რომ $t_0 \div (t + t_0)$ დროის შუალედის განმავლობაში K რაოდენობის მტყუნებების წარმოშობის ალბათობა არ არის დამოკიდებული იმაზე თუ რამდენი იყო მტყუნებები ამ ინტერვალამდე და რა სიხშირით წარმოიშობოდნენ ისინი.

ნაკადის ორდინარობა გამოხატავს იმის პირობას, რომ პრაქტიკულად შეუძლებელია ორი ან მეტი მტყუნების ერთდროულად წარმოშობა ე.ი. დროის ერთსა და იმავე მომენტში.

საბოროტო კომპიუტერებში, როცა ის მუშაობს ექსპლუატაციის სტაბილურ პირობებში მტყუნებათა ნაკადის ძირითადი სახეა - უმარტივესი ნაკადი.

აღდგენების ნაკადების პარამეტრი $\mu(t)$

აღდგენების ნაკადი - ეს არის აღდგენადი საბოროტო კომპიუტერის აღდგენების მიმდევრობა, რომელიც გამოჩნდება დროის შემთხვევით მომენტში. თუ შემოვიღებთ აღდგენების ნაკადის პარამეტრად - აღდგენის ინტენსიურობის $\mu(t)$ -ს ცნებას (მტყუნებათა ინტენსიურობის ცნების $\lambda(t)$ -ს ანალოგიურად), მაშინ აღდგენადი საბოროტო კომპიუტერის აღდგენის ალბათობა იქნება

$$P_{\text{ად}}(t) = 1 - e^{-\mu t}.$$

აღდგენის საშუალო დრო კი

$$T_{\text{ად}} = \int_0^{\infty} (1 - p_{\text{ად}}(t)) dt.$$

ხოლო როცა $\mu(t) = \mu = \text{const}$, ფორმულა მიიღებს სახეს

$$T_{\text{ად}} = 1/\mu.$$

მტყუნებათაშორისი ნამუშევარი $T_{\text{საშ}}$

აღდგენადი საბოროტო კომპიუტერების საიმედოობის ერთ-ერთი ძირითადი მახასიათებელია მტყუნებათაშორისი ნამუშევარი, რომელიც განისაზღვრება როგორც საბოროტო კომპიუტერების ნამუშევარის საშუალო მნიშვნელობა მტყუნებებს შორის. იმ შემთხვევაში, როცა მტყუნებათაშორისი ნამუშევარი გამოხატულია დროის ერთეულში, იხმარება მეორე ტერმინიც: უმტყუნო მუშაობის საშუალო დრო. დროითი

ინტერვალისათვის t_1 ნამუშევრიდან t_2 ნამუშევრამდე მტყუნებათაშორისი $T_{საშ}$ ნამუშევრის გამოთვლისათვის საჭირო ზუსტ განტოლებას აქვს შემდეგი სახე:

$$T_{საშ} = (t_2 - t_1) / [F(t_2) - H(t_1)],$$

სადაც $H(t)$ მტყუნებათა ნაკადის მახასიათებელია.

პრაქტიკული გამოთვლებისათვის ჩვეულებრივ იყენებენ მიახლოებით განტოლებას

$$T_{საშ} \approx (t_2 - t_1) / [m_{საშ}(t_2) - m_{საშ}(t_1)] .$$

ახლა დავრწმუნდეთ იმაში, რომ დამუშავების პერიოდის დამთავრებისას, როცა მტყუნების ნაკადის მახასიათებელი ხდება წრფივი საბორტო კომპიუტერის მტყუნებათაშორისი ნამუშევარი არ არის დამოკიდებული t_1 და t_2 ნამუშევრების მნიშვნელობების არჩევაზე. თუ $t_1 = t_0 + \Delta t_1$ და $t_2 = t_0 + \Delta t_2$, და გამოვიყენებთ (14.20) და (14.21) ფორმულებს მივიღებთ:

$$T_{საშ} = \frac{1}{\omega} \quad (14.22)$$

სადაც ω - მტყუნებათა ნაკადის პარამეტრია.

მტყუნებათაშორისი ნამუშევარი ეწოდება საბორტო კომპიუტერების ფუნქციონირების დროის მათემატიკურ ლოდინს (საშუალო მნიშვნელობას) ორ მეზობელ მტყუნებათა შორის.

სტატისტიკურად მტყუნებათაშორისი ნამუშევარი განისაზღვრება ფორმულით:

$$T_{საშ} = \frac{1}{n} \sum_{i=1}^n t_i,$$

სადაც t_i - აღდგენადი საბორტო კომპიუტერების ნორმალური ფუნქციონირების დროითი ინტერვალია $(i-1)$ და i -ურ მტყუნებათა შორის, n კი მტყუნებათა რიცხვია ექსპლუატაციის დაკვირვების პერიოდში.

აღდგენადი საბორტო კომპიუტერების საიმედო მუშაობის შეფასებისათვის იშვიათად, მაგრამ მაინც იყენებენ უმტყუნო მუშაობის ალბათობას. ასეთ შემთხვევაში უმტყუნო მუშაობის ალბათობა t_1 და t_2 ნამუშევრებს შორის პერიოდში გამოითვლება შემდეგი ფორმულით:

$$p(t_2 - t_1) = e^{-[H(t_2) - H(t_1)]} \quad (14.23)$$

თუ მტყუნებების ნაკადის $H(t)$ მახასიათებელი ხაზოვანია, მაშინ (14.23)-ე ფორმულა გამარტივდება და მიიღებს სახეს:

$$P(t) = e^{-\omega t} = e^{-\frac{t}{T}}, \text{ სადაც } t = t_2 - t_1$$

აღდგენის საშუალო დრო $T_{\text{აღ}}$

აღდგენის საშუალო დრო $T_{\text{აღ}}$ განისაზღვრება სტატისტიკური თანაფარდობით:

$$T_{\text{აღ}} = \frac{1}{m} \sum_{i=1}^m \tau_i,$$

სადაც τ_i - საბორტო კომპიუტერის i -ური მტყუნების აღსადგენად დახარჯული დროა, m კი აღდგენების საერთო რიცხვია დაკვირვების დროის განმავლობაში.

საბორტო კომპიუტერების აღდგენის მომენტები წარმოქმნიან მოთხოვნათა ნაკადს, რომელსაც აღდგენის ნაკადს უწოდებენ. აღდგენის ნაკადის ძირითადი მახასიათებელია - ნაკადის პარამეტრი $\mu(t)$, რომელსაც როგორც ვიცით აღდგენის ინტენსიურობას უწოდებენ და იგი სტატისტიკურად განისაზღვრება როგორც საბორტო კომპიუტერების აღდგენათა საერთო რიცხვის შეფარება აღდგენის ჯამურ დროსთან:

$$\mu = \frac{m}{\sum_{i=1}^m \tau_i}. \quad (14.24)$$

ამ თანაფარდობიდან ჩანს, რომ აღდგენის ინტენსიურობა μ , აღდგენის საშუალო დროის $T_{აღ}$ -ის შებრუნებული სიდიდეა

$$\mu = 1 / T_{აღ}. \quad (14.25)$$

აღდგენადი საბორტო კომპიუტერების დროს მტყუნებათშორისი ნამუშევარი $T_{საშ}$ და $T_{აღ}$ ერთმანეთთან დაკავშირებული არიან მზადყოფნის კოეფიციენტით $K_{აშ}$

$$K_{აშ} = \sum_{i=1}^n t_i / (\sum_{i=1}^n t_i + \sum_{i=1}^m t_i) = \frac{T_{საშ}}{T_{საშ} + T_{აღ}}.$$

ე.ი.

$$K_{აშ} = \frac{T_{საშ}}{T_{საშ} + T_{აღ}}.$$

ეს კოეფიციენტი საშუალებას იძლევა დროის ნებისმიერ მომენტში მონახულ იქნეს საბორტო კომპიუტერების გამართული მდგომარეობის ალბათობა.

საბორტო კომპიუტერების იძულებითი გაცდენის კოეფიციენტი $K_{გაც}$

იძულებითი გაცდენის კოეფიციენტს $K_{გაც}$ უწოდებენ იძულებითი გაცდენის დროის შეფარდებას საბორტო კომპიუტერების გამართული მუშაობისა და იძულებითი გაცდენების დენების დროების ჯამთან:

$$K_{გაც} = t_{გაც} / t_{გაც} + t_{გამ}. \quad (14.25)$$

იძულებითი გაცდენის კოეფიციენტი $K_{გაც}$ წარმოადგენს იმის ალბათობას, რომ საბორტო კომპიუტერი დროის ნებისმიერ მომენტში არამუშა მდგომარეობაშია. (14.25)

ფორმულაში $t_{გაც}$ -იძულებითი გაცდენის დროა, $t_{გამ}$ - საბორტო კომპიუტერების გამართული მუშაობის დრო.

მზადყოფნისა და მოცდენის (გაცდენის) კოეფიციენტები

ისინი ერთმანეთთან დაკავშირებულნი არიან შემდეგი დამოკიდებულებით:

$$K_{გაც} = 1 - K_{შ}. \quad (14.26)$$

$K_{გაც}$ და $K_{შ}$ კოეფიციენტები გამოიყენებიან აღდგენადი საბორტო კომპიუტერების ექსპლუატაციის ხარისხის დასახასიათებლად და მოიცავენ რა ისინი, ერთდროულად, საბორტო კომპიუტერის ორ ან მეტ თვისებას, საიმედოობის კომპლექსურ მაჩვენებლებსაც უწოდებენ.

(14.26) ფორმულა შეიძლება ჩაიწეროს ასეთნაირად:

$$V_{გ} = \frac{T_{აღ}}{T_{აღ} + T_{საშ}} \quad (7.3.21)$$

მაგალითი 14.3. აღდგენადი საბორტო საბორტო კომპიუტერების გამოთვლითი სისტემა შედგება 8 ელემენტისგან. მათი მტყუნებათა ინტენსიურობები მოცემულია ცხრილში 14.3.

ცხრილი 14.3

ელემენტის ნომერი	1	2	3	4	5	6	7	8
λ_i სთ ⁻¹	0,0003	0,0002	0,0009	0,0006	0,0004	0,0003	0,0005	0,0007

ელემენტების აღდგენის ინტენსიურობები ერთნაირია და ტოლია $M=0,4$ სთ⁻¹. გამოვთვალოთ სისტემის საიმედოობის ნაჩვენებლები.

ამოხსნა

განვსაზღვროთ საბორტო სისტემის მტყუნების ინტენსიურობა.

$$\lambda_{\Sigma} = \sum_{i=1}^n \lambda_i = 0,0003 + 0,0002 + 0,0009 + 0,0006 + 0,0004 + 0,0003 + 0,0005 + 0,0007 = 0,0039.$$

მაშინ აღდგენადი საბორტო გამოთვლითი სისტემის მტყუნებამდე ნამუშევარი $T_{საშ}$, აღდგენის საშუალო დრო $T_{აღ}$ და მზადყოფნის კოეფიციენტი შესაბამისად

$$T_{\text{ბაზ}} = \frac{1}{\lambda_b} = \frac{1}{0,0039} = 256,4 \text{ სთ},$$

$$T_{\text{აგ}} = \frac{1}{N} = \frac{1}{0,4} = 2,5 \text{ სთ},$$

$$K_{\text{მბ}} = \frac{T_{\text{ბაზ}}}{T_{\text{ბაზ}} + T_{\text{აგ}}} = \frac{256,4}{256,4 + 2,5} = 0,9903.$$

ტოლი იქნება:

რადგან ელემენტების აღდგენის ინტენსიურობები ერთნაირია, ამიტომ სისტემა შეგვიძლია განვიხილოთ როგორც ერთი ელემენტი მტყუნებების ინტენსიურობით λ და აღდგენის ინტენსიურობით μ .

შემდეგი ფორმულის გამოყენებით მივიღებთ:

$$T_{\text{მბ}}(t) = \frac{\mu}{\lambda_b + N} = \frac{\lambda_b}{\lambda_b + \mu} \cdot e^{-(\lambda_b + \mu)t}$$

$$T_{\text{მბ}}(t) = \frac{0,4}{0,4039} + \frac{0,0039}{0,4039} \cdot e^{-0,4039t}$$

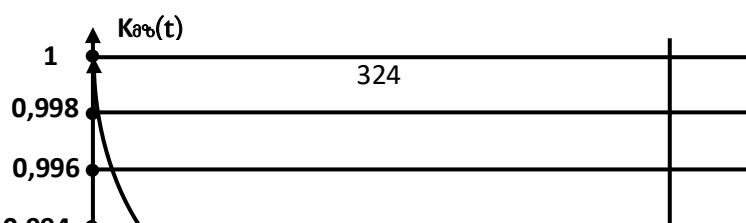
მზადყოფნის ფუნქციის ტაბულირებით 0-დან 40 საათამდე 2 საათის ბიჯით მივიღებთ მნიშვნელობებს, რომლებიც მოყვანილია ცხრილში 14.4.

მზადყოფნის ფუნქციის გრაფიკი გამოსახულია ნახ.14.10-ზე.

ცხრილი

14.4

t სთ	K _{მბ} (t)	t სთ	K _{მბ} (t)	t სთ	K _{მბ} (t)
0	1	14	0,990378	28	0,990344
2	0,994649	16	0,990359	30	0,990344
4	0,992263	18	0,990351	32	0,990344
6	0,9912	20	0,990347	34	0,990344
8	0,990726	22	0,990345	36	0,990344
10	0,990514	24	0,990345	38	0,990344
12	0,990042	26	0,990344	40	0,990344



ნახ. 14.10-დან ჩანს, რომ გარდამავალი პროცესის დრო მცირეა და შეადგენს დაახლოებით 10 საათს. ეს ნიშნავს, რომ მაღალსაიმედო სისტემის შემთხვევაში ($K_{\text{ფ}} = 0,99$) და მისი მუშაობის დიდი ხანგრძლივობისას მიზანშეწონილია სისტემის მზადყოფნა შეფასდეს მზადყოფნის კოეფიციენტით.

მაგალითი 14.4. აღდგენადი საბორტო გამოთვლითი სისტემა შედგება 8 ელემენტისგან. მათი მტყუნებათა ინტენსიურობა იგივეა, რაც წინა მაგალითში (ცხრილი 14.3). ხოლო აღდგენის ინტენსიურობები სხვადასხვაა და მოყვანილია ცხრილში 14.5.

ცხრილი 14.5

ელემენტის ნომერი	1	2	3	4	5	6	7	8
μ_i სთ ⁻¹	0,5	0,2	0,1	0,4	0,9	0,7	0,6	0,8

განისაზღვროს საბორტო სისტემის საიმედოობის მაჩვენებლები.

ამოხსნა

ვიპოვოთ საიმედოობის მაჩვენებლები: მტყუნებამდე ნამუშევარი $T_{საშ}$, აღდგენის საშუალო დრო $T_{აღ}$ და მზადყოფნის კოეფიციენტი $K_{აზ}$ შემდეგი ფორმულების გამოყენებით:

$$T_{საშ} = \frac{1}{\lambda_0}, \quad T_{აღ} = \frac{\sum_{i=1}^n \frac{\lambda_i}{\mu_i}}{\lambda_0}, \quad K_{აზ} = \frac{1}{1 + \sum_{i=1}^n \frac{\lambda_i}{\mu_i}}.$$

ისე, როგორც წინა მაგალითში, სისტემის მტყუნების ინტენსიურობა $\lambda_0 = 0,0039 \text{ სთ}^{-1}$.

განვსაზღვროთ ჯამი:

$$\sum_{i=1}^n \frac{\lambda_i}{\mu_i} = \frac{0,0003}{0,5} + \frac{0,0002}{0,2} + \frac{0,0009}{0,1} + \frac{0,0006}{0,4} + \frac{0,0004}{0,9} + \frac{0,0003}{0,7} + \frac{0,0005}{0,6} + \frac{0,0007}{0,8} = 0,0147$$

მივიღებთ:

$$T_{საშ} = \frac{1}{0,0039} = 256,4 \text{ სთ}$$

$$T_{აღ} = \frac{0,0147}{0,0039} = 3,77 \text{ სთ}$$

$$K_{აზ} = \frac{1}{1 + 0,0147} = 0,985$$

საბოლოო სისტემის უმტყუნო მუშაობის ალბათობისა და მტყუნების ინტენსიურობის განსაზღვრა

საბოლოო სისტემის უმტყუნო მუშაობის ალბათობა განისაზღვრება მისი შემადგენელი ელემენტების უმტყუნო მუშაობის ალბათობების ნამრავლით:

$$P(t) = P_1(t) \cdot P_2(t) \dots \cdot P_n(t) = \prod_{i=1}^n P_i(t) \quad (14.27)$$

სისტემის მტყუნების ინტენსიურობა ტოლია მისი შემადგენელი ელემენტების ელემენტების მტყუნებების ინტენსიურობების ჯამისა:

$$\Lambda(t) = \lambda_1(t) + \lambda_2(t) + \dots + \lambda_n(t) = \sum_{i=1}^n \lambda_i(t). \quad (14.28)$$

14.4. საბორტო კომპიუტერების სტრუქტურული საიმედოობა და მისი ამაღლების მეთოდები

ნებისმიერი რადიოელექტრონული აპარატურის სტრუქტურული საიმედოობა, რა თქმა უნდა, მათ შორის, საბორტო კომპიუტერის საიმედოობაც, ეს არის მისი რეზულტირებული საიმედოობა, როცა ცნობილია მისი სტრუქტურული სქემა და ყველა იმ ელემენტების საიმედოობის მნიშვნელობები, რომლებიც შეადგენენ სწორედ საბორტო კომპიუტერის სტრუქტურულ სქემას. ამასთან ერთად, ელემენტის ცნება გულისხმობს საბორტო კომპიუტერის საერთო ელექტრულ პრინციპულ სქემაში ჩართულ როგორც ყველა ელემენტის (ინტეგრალური მიკროსქემები, რეზისტორები, კონდენსატორები და ა.შ.) ერთობლიობას, რომლებიც ასრულებენ განსაზღვრულ კონკრეტულ ფუნქციას, ისე იმ ელემენტების ერთობლიობასაც, რომლებიც არ შედიან საბორტო კომპიუტერების სტრუქტურულ სქემაში და იწოდებიან დამხმარე ელემენტებად, მაგრამ საბორტო კომპიუტერების საიმედოობის გაანგარიშებისათვის აუცილებელ კომპონენტებს წარმოადგენენ. ასეთებია: ნარჩილავი შეერთებები, გასართი შეერთებები, დამაგრების ელემენტები და სხვა.

რაოდენობრივი მახასიათებლების განსაზღვრისათვის ახდენენ საბორტო კომპიუტერების სტრუქტურული სქემის შედგენას, მიუთითებენ მოწყობილობის ელემენტებს (ბლოკები, კვანძები) და მათ შორის კავშირებს. შემდეგ ახდენენ სქემის ანალიზს და გამოყოფენ მის იმ ელემენტებს და კავშირებს, რომლებიც განაპირობებს მოცემული მოწყობილობის ძირითადი ფუნქციის შესრულებას. ამის შემდეგ ხდება გამოყოფილი ძირითადი ელემენტებისა და კავშირების მიხედვით ფუნქციური სქემის შედგენა, რომელშიც გამოყოფენ ელემენტებს არა კონსტრუქციული, არამედ ფუნქციური ნიშნის მიხედვით და ის, რომ თითოეულ ფუნქციურ ელემენტს უნდა შეესაბამებოდეს

დამოუკიდებლობა, რაც იმას ნიშნავს, რომ ერთი ფუნქციური ელემენტის მტყუნებამ არ უნდა გამოიწვიოს მეორე ელემენტის მტყუნების წარმოშობის ალბათობის ცვლილება. ამიტომ ცალკეული ფუნქციური სქემების (მოწყობილობათა კვანძები, ბლოკები) შედგენისას, ხანდახან საჭირო ხდება გაერთიანდეს ის კონსტრუქციული ელემენტები, რომელთა მტყუნებები ურთიერთდაკავშირებულია, თუმცა არ ახდენენ სხვა ელემენტების მტყუნებებზე ზეგავლენას.

საბორტო კომპიუტერის საიმედოობის რაოდენობრივი მახასიათებლების განსაზღვრა შესაძლებლობას იძლევა სტრუქტურული სქემის დახმარებით გადაწყდეს საბორტო კომპიუტერის შემადგენელი ნაწილების (ფუნქციური ელემენტების, კვანძების, ტიპური შეცვლის ელემენტების, ბლოკების, პანელების, დგარების, რამების) ყველაზე უფრო ოპტიმალური ვარიანტის შერჩევა საიმედოობისა და კონსტრუქციის თვალსაზრისით.

სტრუქტურული სქემის აგებისას იყენებენ ელემენტების მიმდევრობით, პარალელურ და მიმდევრობით-პარალელურ შეერთებებს.

ელემენტების მიმდევრობითი შეერთების დროს (ნახ.14.11) სქემის საიმედო მუშაობისათვის აუცილებელია ყველა ფუნქციური ელემენტის მუშაობა.

ასეთი სქემის უმტყუნო მუშაობის ალბათობა ტოლია ყველა ფუნქციური ელემენტის უმტყუნო მუშაობის ალბათობების ნამრავლისა

$$P(t) = P_1(t) \cdot P_2(t) \cdot \dots \cdot P_n(t) = \prod_{i=1}^n P_i(t) \quad (14.29)$$

თუ ყველა ელემენტის უმტყუნო მუშაობის ალბათობა ერთნაირია, ე.ი.

$$P_1(t) = P_2(t) = \dots P_n(t) = P_0(t), \quad \text{მაშინ} \quad P(t) = P_0^n(t).$$

თუ

$$P_i(t) = e^{-\lambda_i t}, \quad \text{მაშინ} \quad P(t) = \prod_{i=1}^n P_i(t) = e^{-\sum_{i=1}^n \lambda_i t} = e^{-\lambda_{\Sigma} t},$$

$$\text{სადაც } \lambda_{\Sigma} = \lambda_1 + \lambda_2 + \lambda_1 + \dots + \lambda_1.$$

ამ შემთხვევაში მტყუნებამდე ნამუშევარის საშუალო დრო ტოლია

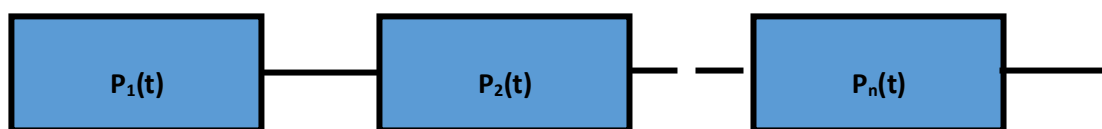
$$T = \frac{1}{\sum_{i=1}^n \lambda_i} = \frac{1}{\lambda_{\Sigma}}.$$

ყველა ელემენტის ტოლი საიმედოობის შემთხვევაში

$$(\lambda_1 = \lambda_2 = \lambda_3 = \dots = \lambda_n = \lambda).$$

გვეყენება:

$$\lambda_{\Sigma} = n\lambda_i \quad P(t) = e^{-n\lambda t}; \quad T_{საშ} = \frac{1}{n\lambda}.$$



ნახ. 14.11. სტრუქტურულ სქემაში ელემენტების მიმდევრობითი ჩართვის სქემა

პარალელური შეერთების შემთხვევაში სტრუქტურული სქემის აგებისას ელემენტების (ნახ.14.12) სქემაში შემავალი ელემენტების მტყუნების ინტენსიურობა ტოლია

$$q_j(t) = 1 - P_j(t).$$

მთელი სქემის მტყუნებას ადგილი ექნება მხოლოდ მაშინ, როცა ადგილი ექნება ყველა ელემენტების მტყუნებას, ე.ი.

$$Q(t) = q_1(t) \cdot q_2(t) \cdot \dots \cdot q_m(t) = \prod_{j=1}^m [1 - P_j(t)]$$

სადაც - m პარალელურად შეერთებული ელემენტების რიცხვია. მთელი სქემის უმტყუნო მუშაობის ალბათობა იქნება:

$$P(t) = 1 - Q(t) = 1 - \prod_{j=1}^m [1 - P_j(t)] \quad (14.30)$$

ტოლსაიმედოობიანი ელემენტების გამოყენების შემთხვევაში, თუ

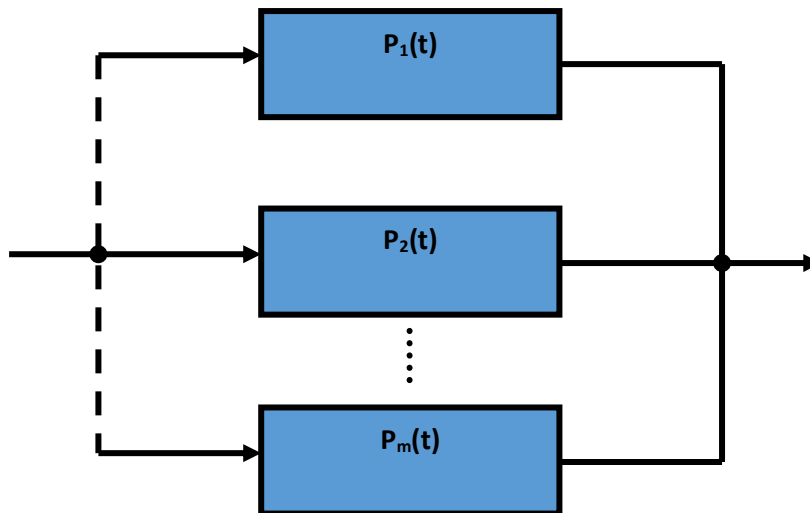
$$P_1(t) = P_2(t) = \dots = P_m(t) = P_0(t),$$

მაშინ

$$Q(t) = q_0^m(t) = [1 - P_0(t)]^m \quad \text{და} \quad P(t) = 1 - [1 - P_0(t)]^m.$$

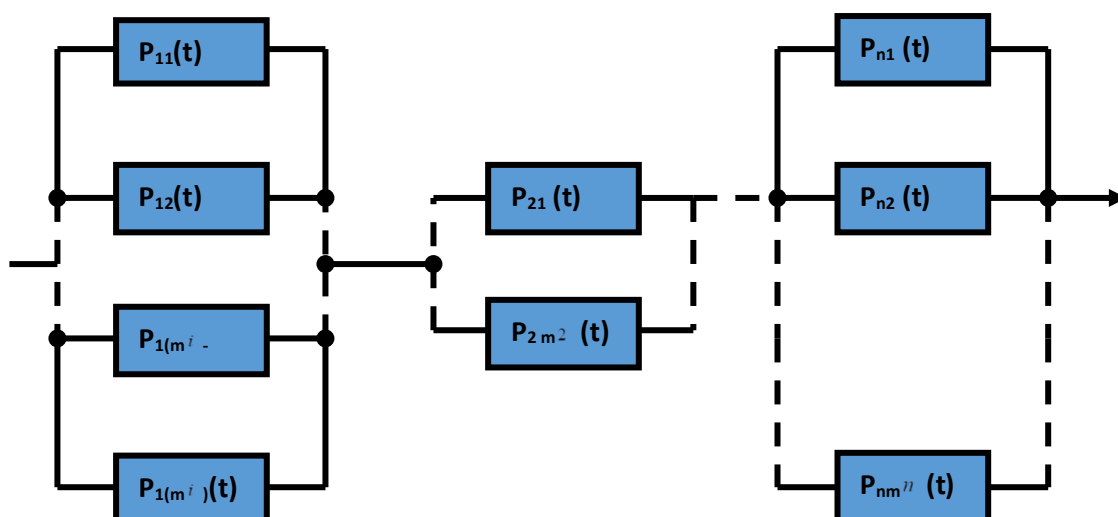
თუ $P_0(t) = e^{-\lambda t}$, მაშინ

$$T_{საშ} = 1/\lambda + 1/2\lambda + \dots 1/m\lambda.$$



ნახ. 14.12. სტრუქტურულ სქემაში ელემენტების პარალელური ჩართვის სქემა

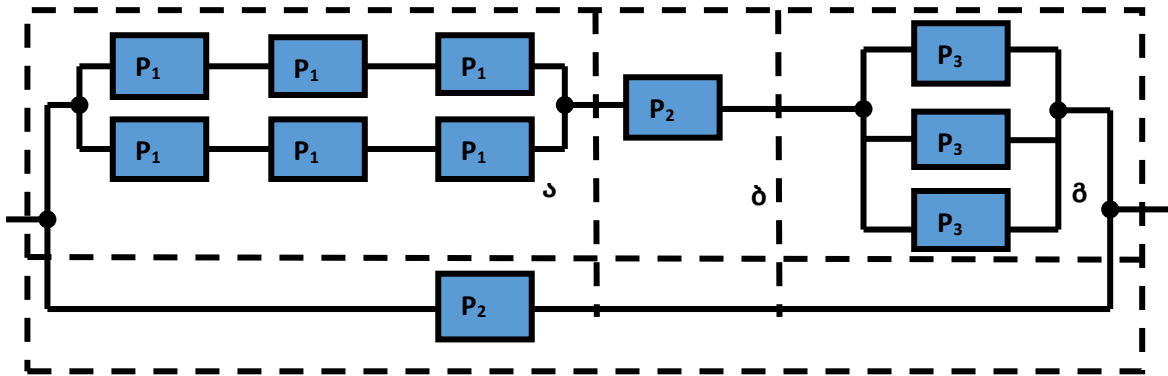
ელემენტების პარალელურ-მიმდევრობითი შეერთებისას (ნახ.14.13), ჯერ უნდა მოინახოს თითოეული პარალელურად შეერთებული ელემენტების ჯგუფისთვის უმტყუნო მუშაობის ალბათობა, ხოლო შემდეგ კი მთელი სქემისთვის.



ნახ.14.13. სტრუქტურულ სქემაში ელემენტების პარალელურ-მიმდევრობითი ჩართვის სქემა

მაგალითი 14.5. გამოვთვალოთ ნახ.14.14.-ზე მოცემული სქემის უმტყუნო მუშაობის ალბათობა, მისი შემადგენელი ელემენტების უმტყუნო მუშაობის ცნობილი ალბათობების მიხედვით: $P_1=0,9$; $P_2=0,97$; $P_3=0,88$.

ამოხსნა



ნახ.14.14. სქემის უმტყუნო მუშაობის ალბათობის გაანგარიშების მაგალითისთვის

გამოვყოთ ა, ბ, და გ უბნები. მაშინ (14.28)-ის საფუძველზე მიღებული აღნიშვნების გათვალისწინებით შეგვიძლია ჩავწეროთ

$$P_I = P_a \cdot P_b \cdot P_g,$$

ხოლო ამ უკანასკნელის გათვალისწინების საფუძველზე გვექნება

$$P = 1 - (1 - P_I)(1 - P_{II}) = 1 - (P_a \cdot P_b \cdot P_g)(1 - P_{II}).$$

ვღებულობთ P_a და P_b -ის გვექნება

$$P_a = 1 - (1 - P_1^3)^2, \quad P_b = 1 - (1 - P_3^3)^3.$$

რაც შეეხება P_g -ს იგი ტოლია

$$P_g = P_{II} = P_2$$

აქედან

$$P = 1 - \left[1 - \left[1 - (1 - P_1^3)^2 \right] \cdot P_2 \left[1 - (1 - P_3^3)^3 \right] \right] \cdot (1 - P_2).$$

ამ გამოსახულებაში P_1, P_2, P_3 -ის მნიშვნელობის ჩასმა მოგვცემს, რომ $P = 0,996$.

მაგალითი 14.6. განვსაზღვროთ იმ საბორტო კომპიუტერის საშუალო მტყუნებამდე ნამუშევარი $T_{საშ}$ და უმტყუნო მუშაობის ალბათობა $P(t)$, რომელიც განკუთვნილია საჰაერო ხომალდზე ექსპლუატაციისათვის თუ იგი შედგება $\lambda = 10^{-6}1/$

სთ. $T_{საშ}$ და $P(t)$ საიმედოობის მაჩვენებლები განსაზღვრულ იქნეს იმ შემთხვევისათვის როცა $t=20$ სთ და მოცემულია დამატებითი უგანზომილებო კოეფიციენტის K_1 -ს მნიშვნელობა, რომელიც ითვალისწინებს საჰაერო ხომალდზე საბორტო კომპიუტერის ექსპლუატაციის პირობებს. საჰაერო ხომალდისათვის $K_1 = 150$.

ამოხსნა

ექსპლუატაციის პირობების გათვალისწინებით გვექნება:

$$\lambda_z = N \cdot K_1 \cdot \lambda = 200 \cdot 150 \cdot 10^{-6} \text{ 1/სთ} = 0,03 \text{ 1/სთ}$$

$$T_{საშ} = \frac{1}{\lambda_z} = \frac{1}{0,03} = 33,3 \text{ სთ.}$$

$$P(t) = e^{-N\lambda K_1 t} = e^{-200 \cdot 10^{-6} \cdot 150 \cdot 20} = e^{-0,6} = 0,553.$$

ამ მაგალითში მიღებული საიმედოობის მაჩვენებლები $T_{საშ} = 33,3$ სთ და $P(20) = 0,553$. უმრავლეს შემთხვევაში არ აკმაყოფილებენ დამმუშავებლების მოთხოვნებს საიმედოობის დონის თვალსაზრისით, ამიტომ დროის აუცილებელი მოთხოვნა საბორტო კომპიუტერების საიმედოობის ამაღლებისათვის გამოყენებულ იქნეს სხვადასხვა მეთოდები, რომელთა შორის ძირითადად იყენებენ საბორტო კომპიუტერების საიმედოობის ამაღლების სტრუქტურულ ან ინფორმაციულ მეთოდებს.

გამოთვლითი სისტემების და საბორტო კომპიუტერების საიმედოობის გაზრდის ერთ-ერთი საშუალებაა მათ სტრუქტურაში სქემური სიჭარბის შემოტანა.

სქემური სიჭარბის ერთ-ერთ სახეს წარმოადგენს სტრუქტურული დარეზერვირება, რომელიც გულისხმობს მოწყობილობის სქემაში დამატებითი ელემენტების ჩართვას და ამის საფუძველზე მოწყობილობის ცალკეული ნაწილების მტყუნებების კომპენსირებას და მისი საიმედო მუშაობის უზრუნველყოფას. დარეზერვება ეფექტურია მხოლოდ იმ შემთხვევაში, როცა უწყისივრობები სტატისტიკურად დამოუკიდებელი არიან.

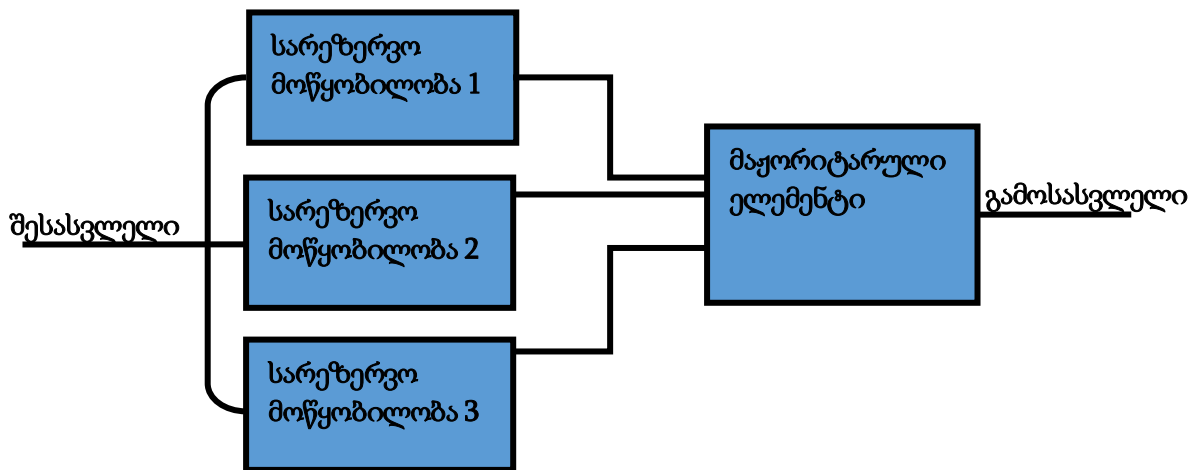
ზოგადად დარეზერვირება არის საიმედოობის გაზრდის მეთოდი ნაკეთობაში ელემენტთა სასარგებლო სიჭარბის შემოტანის ხარჯზე. დარეზერვების ნაირსახეობათა კლასიფიკაციის მიხედვით განასხვავებენ დარეზერვების შემდეგ სახეებს: საერთო

დარეზერვება, განცალკავებული დარეზერვება, მუდმივი დარეზერვება, დარეზერვება შენაცვლებით, დარეზერვება მთელი და წილადი ჯერადობებით და სხვა.

სტრუქტურული სიჭარბის შემოტანის შემთხვევაში საბორტო გამოთვლითი სისტემებისა და საბორტო კომპიუტერების საიმედოობის ამაღლების თვალსაზრისით შედარებით ყველაზე უფრო რაციონალურ სქემად ითვლება ის სქემა, რომლის აგებისა და ფუნქციონირების დროს გამოიყენება მაჟორიტარული ლოგიკის პრინციპი და რომელსაც ახორციელებს მაჟორიტარული ელემენტი ე.წ. „უმრავლესობის ლოგიკით“. მაჟორიტარული ელემენტის მუშაობა მდგომარეობს შემდეგში: ელემენტის შესასვლელზე მიეწოდება ორობითი სიგნალები კენტი რაოდენობის იდენტური ელემენტებისგან. ელემენტის გამოსასვლელი სიგნალი დებულობს მნიშვნელობას, რომელიც ტოლია შესასვლელი სიგნალების უმრავლესობის მიერ მიღებული მნიშვნელობისა. ყველაზე უფრო ფართო გამოყენება ჰპოვა მაჟორიტარულმა ელემენტებმა, რომლებიც მუშაობენ კანონით „ორი სამიდან“ („2 3“-დან). ასეთ ელემენტებში გამოსასვლელი სიგნალის მნიშვნელობა ტოლია ორი ერთნაირი შესასვლელი სიგნალების მნიშვნელობისა. გარდა ამისა ცნობილია მაჟორიტარული ელემენტები, რომლებიც მუშაობენ ლოგიკით „სამი ხუთიდან“, „ოთხი შვიდიდან“ და ა.შ., თუმცა ისინი გარკვეული მოსაზრებების გამო შედარებით უფრო იშვიათად გამოიყენებიან პრაქტიკაში.

მაჟორიტარული ელემენტის სქემა, რომელიც მუშაობს ლოგიკით „ორი სამიდან“ წარმოდგენილია ნახ.14.15-ზე. მის მიერ განხორციელებული ლოგიკური ფუნქცია $Z = X_1X_2 + X_2X_3 + X_1X_3$ იწოდება მაჟორიტარულ ფუნქციად.

დარეზერვების სქემას, სადაც სარეზერვო მოწყობილობების გამოსასვლელები დატვირთვას უერთდება არა უშუალოდ, არამედ მაჟორიტარული ელემენტის საშუალებით მაჟორიტარული დარეზერვების სქემა ეწოდება (ნახ.14.5)



ნახ.14.15 .მაჟორიტარული დარეზერვების სქემა

ნახ. 14.15-ზე წარმოდგენილ მაჟორიტარული დარეზერვების სქემაზე ნაჩვენებია სამივე სარეზერვო მოწყობილობა (სარეზერვო მოწყობილობა 1, სარეზერვო მოწყობილობა 2, სარეზერვო მოწყობილობა 3) იდენტური არიან, ხოლო მტყუნებები მათში კი დამოუკიდებელი.

მაჟორიტარული დარეზერვების ხერხი უმეტესად ხშირად გამოიყენება საბორტო მმართველ გამოთვლით სისტემებში და მმართველ საბორტო კომპიუტერებში. მმართველ გამოთვლით სისტემებში სარეზერვო მოწყობილობების სახით გამოიყენება მმართველი საბორტო კომპიუტერები, ხოლო მმართველ საბორტო კომპიუტერებში კი მიკროპროცესორები, რომლებიც მუშაობენ ერთი და იგივე პროგრამებით. ასეთ დარეზერვების სქემაში, რომლის გამოსასვლელი სიგნალი მაჟორიტარული ელემენტის გამოსასვლელი სიგნალია, ასევე წარმოადგენს მთელი სქემის გამოსასვლელ სიგნალს, რომელიც ყოველთვის ემთხვევა იდენტურად მომუშავე უმრავლესი მოწყობილობების გამოსასვლელი სიგნალების სახეს.

იმდენად რამდენადაც მაჟორიტარული ელემენტის (მე) ტიპი სრულად განსაზღვრავს დარეზერვების სქემას, ამიტომ ზოგადად სქემას რომელიც შედგება m რაოდენობის დარეზერვებული მოწყობილობებისგან, რომლებთაგან სწორად მომუშავე მოწყობილობების მინიმალურ რაოდენობას, რომელთა დროსაც გამოსასვლელი სიგნალი კვლავ რჩება სწორი აღვნიშნავთ k -თი და იმავდროულად გავითვალისწინებთ

მაჟორიტარული ელემენტის არსებობასაც, რომელიც ასრულებს მაჟორიტარულ ფუნქციას „ k m “-დან, ვუწოდებთ C_m^K სქემას.

ახლა უფრო დაწვრილებით განვიხილოთ ნახ.14.15-ზე წარმოდგენილი მაჟორიტარული სტრუქტურის უმტყუნო მუშაობის ალბათობის განსაზღვრის მეთოდიკა. უპირველეს ყოვლისა ის გულისხმობს მისი შემადგენელი სარეზერვო მოწყობილობების შესაძლო მდგომარეობათა ალბათობების კომბინაციათა ჯამს მაჟორიტარული ელემენტის უმტყუნო მუშაობის ალბათობის გათვალისწინებით. ასეთი მაჟორიტარული სტრუქტურის სქემის შემთხვევაში მისი უმტყუნო მუშაობა მიღწეულ იქნება ორ შემთხვევაში:

1. როცა ყველა სამივე სარეზერვო მოწყობილობა და მაჟორიტარული ელემენტი გამართულია (ერთი მდგომარეობა). ამ დროს მაჟორიტარული სტრუქტურის უმტყუნო მუშაობის ალბათობა P_1 -ის ტოლია

$$P_1 = P^3 \cdot P_{\partial\eta},$$

სადაც P - ცალკეული სარეზერვო მოწყობილობების უმტყუნო მუშაობის ალბათობაა, და ისინი ერთმანეთის ტოლია, $P_{\partial\eta}$ - მაჟორიტარული ელემენტის უმტყუნო მუშაობის ალბათობაა.

2. როცა სამი სარეზერვო მოწყობილობიდან მხოლოდ ერთ-ერთია მწყობრიდან გამოსული (ან პირველი, ან მეორე, ან მესამე სარეზერვო მოწყობილობა) და გამართულია მაჟორიტარული ელემენტი (სამი მდგომარეობა).

ამ დროს უმტყუნო მუშაობის ალბათობა P_2 ტოლია

$$P_2 = 3P^2(1-P)P_{\partial\eta}.$$

აქედან გამომდინარე მაჟორიტარული სტრუქტურის უმტყუნო მუშაობის ალბათობა იქნება

$$\begin{aligned} P' &= P_1 + P_2 = P^3 P_{\partial\eta} + 3P^2(1-P) \cdot P_{\partial\eta} = P_{\partial\eta} [P^3 + 3P^2(1-P)] = P_{\partial\eta} (P^3 + 3P^2 - 3P^3) = \\ &= P_{\partial\eta} (3P^2 - 2P^3) \end{aligned}$$

იდეალური მაჟორიტარული ელემენტისთვის ($P_{\text{მე}}=1$) და ეს გამოსახულება ღებულობს სახეს $P' = 3P^2 - 2P^3$.

მაჟორიტარული დარეზერვების შედეგად მიღებული მოგების კოეფიციენტი საიმედოობის მიხედვით დაურეზერვებელ მოწყობილობასთან შედარებით გამოითვლება შემდეგი ფარდობით:

$$\alpha = \frac{P'}{P} = \frac{P_{\text{მე}}(3P^2 - 2P^3)}{P} = P_{\text{მე}}(3P - 2P^2) \quad (7.7.6)$$

α -ს მნიშვნელობა იცვლება დიაპაზონში $1 < \alpha < 1,125$. მისი მაქსიმალური მნიშვნელობა $\alpha = 1,125$ მიიღება მაშინ, როცა $P = 0,75$.

ზოგადად მაჟორიტარული სქემის უმტყუნო მუშაობის ალბათობა, როცა $K = \frac{(m-1)}{2}$ -ს და გამოყენებულია მხოლოდ ერთი მაჟორიტარული ელემენტი განისაზღვრება შემდეგი გამოსახულებით:

$$P' = P_{\text{მე}} \sum_{i=0}^{\frac{(m-1)}{2}} (i^m) \cdot P^{(m-i)} (1 - P) \quad (7.7.7)$$

მაჟორიტარული დარეზერვების სქემის საიმედოობის ამაღლებისთვის შესაძლებელია შემოტანილ იქნეს ჭარბი მაჟორიტარული ელემენტები მის გამოსასვლელზე, ამ შემთხვევაში მაჟორიტარული ელემენტების საერთო რიცხვიდან ერთი ან რამდენიმე მაჟორიტარული ელემენტის მწყობრიდან გამოსვლა არ გამოიწვევს მთლიანი სქემის მტყუნებას უმტყუნებოდ მომუშავე უმრავლესი მაჟორიტარული ელემენტების მუშაობის გამო. ასეთი სქემის უმტყუნო მუშაობის ალბათობა განისაზღვრება როგორც

$$P' = \sum_{i=0}^{\frac{(m-1)}{2}} (i^m) \cdot P^{(m-i)} (1 - P)^i \sum_{j=0}^{\frac{(m-1)}{2}} (j^m) \cdot P_{\text{მე}}^{(m-j)} (1 - P_{\text{მე}})^j \quad (7.7.8)$$

მაგალითი 14.7. გამოვთვალოთ იმ მაჟორიტარული დარეზერვების სქემის უმტყუნო მუშაობის ალბათობა (P') და მოგება საიმედოობის მიხედვით (α), რომელიც ახორციელებს მაჟორიტარულ ფუნქციას „2-3“-დან, როცა დაურეზერვებელი

მოწყობილობების უმტყუნო მუშაობის ალბათობები ტოლია 0,8-ის და მაქორიტარული ელემენტი წარმოადგენს უმტყუნებოდ მომუშავე ელემენტს ($P_{\theta\theta}=1$).

ამოხსნა

უმტყუნო მუშაობის ალბათობა

$$P' = 3P^2 - 2P^3 = 3 \cdot 0,8^2 - 2 \cdot 0,8^3 = 3 \cdot 0,64 - 2 \cdot 0,512 = 1,92 - 1,024 = 0,896.$$

მოგება საიმედოობის მიხედვით იქნება

$$\alpha = \frac{P'}{P} = \frac{0,896}{0,8} = 0,12.$$

14.5. საბორტო გამოთვლითი სისტემების საიმედოობის ამაღლება ავტომატური რეკონფიგურაციის საშუალებით

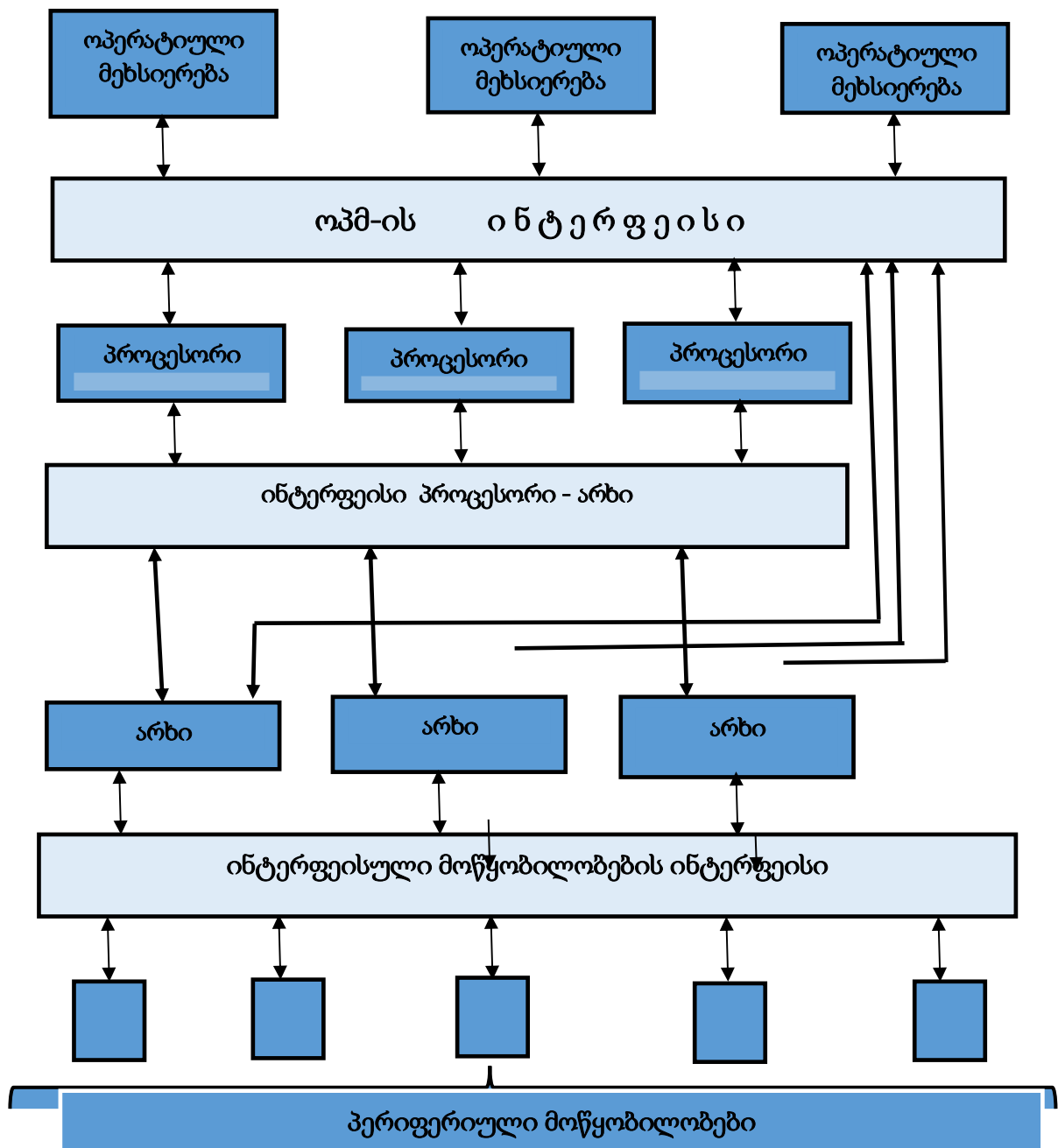
საბორტო გამოთვლითმა სისტემებმა რომლებიც მუშაობენ რეალურ დროში საავიაციო რადიოელექტრონული კომპლექსის შემადგენლობაში, უნდა უზრუნველყონ უწყვეტი ხანგრძლივი ფუნქციონირება, ისე რომ არ დაუშვან რადიოელექტრონული მოწყობილობებში ავარიული სიტუაციების ან მტყუნებების წარმოშობა, როცა გამოთვლითი სისტემის მოწყობილობებში ადგილი აქვს უწყვირობებს ან მტყუნებებს. ჩვეულებრივ ფუნქციები (პროგრამები), რომლებსაც ახორციელებენ საბორტო ციფრული გამოთვლითი სისტემები, არაერთმნიშვნელოვანია. ამიტომ საბორტო გამოთვლითი სისტემისთვის საიმედოობის ძირითად მახასიათებლად მტყუნებამდე საშუალო ნამუშევარი კი არ უნდა ჩაითვალოს, როგორც ეს ხდება საბორტო კომპიუტერების შემთხვევაში, არამედ დროის ამ ინტერვალზე სისტემის ნებისმიერი შემადგენელი ელემენტების მხრიდან დაკისრებული ძირითადი ფუნქციების შესრულების უნარის დაკარგვის ალბათობა. ეს ალბათობა შესაძლებელია მიღებულ იქნეს სისტემის იმედუნარიანობის ზომად.

სისტემის საიმედოობის ამაღლება მიიღწევა სისტემაში ჭარბი მოწყობილობების და სპეციალური ლოგიკური ორგანიზაციის შემოტანით, რომლის საშუალებით მიიღწევა სისტემის ისეთი რეკონფიგურაცია, რომ მტყუნებების წარმოშობის დროსაც კი სისტემა შეინარჩუნებს თავისი ძირითადი ფუნქციების შესრულებას.

მაღალი საიმედოობის სისტემების აგებისათვის ავტომატური რეკონფიგურაციის საფუძველზე ძირითად პრინციპებს წარმოადგენს: მრავალმოწყობილობითობა (მათ რიცხვში მრავალპროცესორულობა; ოპერატიული მეხსიერების, პროცესორების, არხების პერიფერიული მოწყობილობების საერთო ველები); ერთტიპურ მოწყობილობებს შორის ფუნქციების დინამიკური განაწილება.

მრავალმოწყობილობადობის პრინციპი გულისხმობს რომ სისტემა უნდა შეიცავდეს ერთტიპური მოწყობილობების (პროცესორების, ოპერატიული მეხსიერების მოდულების, არხების და სხვ.) რამდენიმე ეგზემპლიარს, ამასთან უზრუნველყოფილ უნდა იქნეს ყველა ტიპის მოწყობილობების სიჭარბე რომლითაც მიღწეულ იქნება საიმედოობის ამაღლება და სასიცოცხლოდ მნიშვნელოვანი ფუნქციების შესრულება (ნახ.14.16).

საერთო ველის ცნების ქვეშ იგულისხმება მოწყობილობების ტოლდაშვებადობის პრინციპი, რომელიც თავის მხრივ იძლევა იმის შემოთავაზებას, რომ: ოპერატიული მეხსიერების ყველა მოდული დაშვებადი იყოს ყველა პროცესორისა და არხების მხრიდან; ყველა არხი იდენტური და დაშვებადი იყოს სისტემის ნებისმიერი პროცესორისთვის; ყველა პერიფერიული მოწყობილობა იყოს დაშვებადი ნებისმიერი არხისთვის. მეხსიერების ყველა მოდული დაშვებადი იყოს ყველა პროცესორისა და არხების მხრიდან; ყველა არხი იდენტური და დაშვებადი იყოს სისტემის ნებისმიერი პროცესორისთვის; ყველა პერიფერიული მოწყობილობა იყოს დაშვებადი ნებისმიერი არხისთვის.



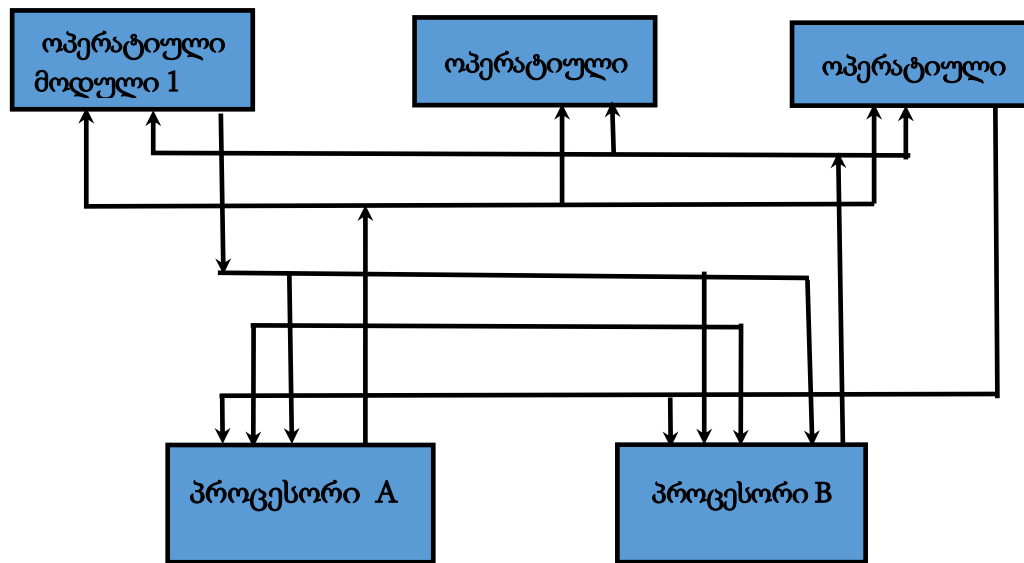
ნახ.14.16

ფუნქციის დინამიკური განაწილება ნიშნავს, რომ პროგრამები პროცესორთან ხისტად არ არის მიბმული და მათი შესრულება შესაძლებელია კომპლექსის ნებისმიერ პროცესორზე, შეტანა-გამოტანის ოპერაციებიც არაა ასევე ხისტად მიბმული არხებთან და შეუძლიათ ნებისმიერი არხით სარგებლობა. ფუნქციების დინამიკური განაწილების

შემთხვევაში წინასწარ არ არის ცნობილი ერთტიპური მოწყობილობებიდან რომელი შეასრულებს მოცემულ ფუნქციას. უფრო მეტიც, მუშაობა შეიძლება დაიწყოს ერთ მოწყობილობაზე, გაგრძელდეს მეორეზე და დამთავრდეს მესამე მოწყობილობაზე.

საერთო ველების და ფუნქციების დინამიკური განაწილება ერთტიპურ მოწყობილობებს შორის, სისტემას საშუალებას აძლევს შეინარჩუნოს მუშაობისუნარიანობა, მანამ, სანამ იარსებებს თითოეული ტიპის მოწყობილობებიდან თუნდაც ერთი მაინც, რომელიც იქნება გამართული. გამოთვლითი სისტემების პრაქტიკული რელიზაციისათვის აუცილებელია შემდეგი ლოგიკური თვისებების უზრუნველყოფა: მრავალკავშირიანი ინტერფეისი, არაცხადი დამისამართება და პროგრამების დინამიკური განაწილება. ეს თვისებები მთლიანობაში უზრუნველყოფს რეკონფიგურაციის ავტომატურ შესრულებას მინიმალურ დროში, ყველაზე უფრო საჭირო ფუნქციების შენახვით და პროგრამების ავტომატური აღდგენით, რომლებიც შეწყვეტილი იყო დაბრკოლებებით და მტყუნებებით.

მრავალკავშირიანი ინტერფეისი (ნახ.14.17) საშუალებას იძლევა მოწყობილობები დუკავშირდნენ ერთმანეთს დამოუკიდებელი სალტების საშუალებით განსხვავებით საბორტო კომპიუტერებისგან რომლებისთვისაც დამახასიათებელია ერთკავშირიანი ინტერფეისები და რომლებშიც (ერთკავშირიან ინტერფეისებში) საერთო სალტები ნაწილდება დროში. მრავალკავშირიან ინტერფეისში გამორიცხულია მაქანებსშორისი ინტერფეისის მწყობრიდან გამოსვლა სალტებიდან ერთი სალტის გაუმართაობის შემთხვევაში (ეს მოითხოვს დამატებით მოწყობილობებს). მიზანშეწონილია ინტერფეისი აიგოს ისეთნაირად, რომ თითოეულ მოწყობილობას გააჩნდეს ერთი გამოსასვლელი და სალტების რამდენიმე დამოუკიდებელი მოწყობილობა.



ნახ.14.17

არაცხადი დამისამართება. ერთტიპური მოწყობილობები, რომლებიც ემსახურებიან შეტანა-გამოტანის ოპერაციას (არხები, ინტერფეისული მოწყობილობების მართვის ბლოკები) გააჩნიათ ერთიანი ლოგიკური ნომრები და ამიტომ ფაქტობრივად შეტანა-გამოტანის ბრძანებით არ მისამართდებიან. მიმართვა შეტანა-გამოტანის ოპერაციის შესრულებაზე მიიღება მიმართვის გავლის გზაზე მოცემული ტიპის პირველ გამართულ და თავისუფალი მოწყობილობის მიერ. არაცხადი დამისამართება მოქმედებს არა მარტო საწყისი ამორჩევის ოპერაციის დროს, არამედ ასევე პერფერული მოწყობილობის სელექტორული და მულტიპლექსური მომსახურების დროსაც.

შეტანა-გამოტანის ერთი და იგივე ოპერაციით მულტიპლექსური მომსახურების სხვადასხვა სეანსებში შესაძლებელია გამოყენებულ იქნეს სხვადასხვა არხები და მართვის ბლოკები. არაცხადი დამისამართება და ინტერფეისების მრავალკავშირობა უზრუნველყოფს ოპერატიული მეხსიერებიდან თავისუფალი და გამართული გზის ავტომატურ არჩევას პერიფერიული მოწყობილობისაკენ.

პროგრამების დინამიკური განაწილება პროცესორებს შორის. ოპერატიულ მეხსიერებაში ორგანიზდება რიგები პროგრამებისთვის რომელთაც გააჩნიათ სხვადასხვა პრიორიტეტი. რიგი უფრო მაღალი ნომრით შეესაბამება უფრო მაღალ პრიორიტეტს. პროგრამები შეიტანება რიგში წყვეტის თავისი პრიორიტეტის მიხედვით. აპარატურა

პროცესორებს შორის ირჩევს კანდიდატს წყვეტაზე რომელსაც წარმოადგენს პროცესორი, რომელიც ამუშავებს პროგრამას უმცირესი პრიორიტეტით. აპარატურა უწყვეტად ადარებს პროცესორ-კანდიდატის პროგრამის პრიორიტეტს წყვეტაზე არაცარიელი რიგის პრიორიტეტიდან უდიდესი ნომრით. თუ რიგში გამოჩნდება პროგრამა მეტი პრიორიტეტით, პროცესორ-კანდიდატი წყვეტს თავის პროგრამას, შეაქვს ის შესაბამისი პრიორიტეტის რიგში და იწყებს იმ პროგრამის შესრულებას რომელსაც გააჩნია ყველაზე დიდი პრიორიტეტი. შეწყვეტილი პროგრამის დამუშავება გაგრძელებული იქნება როცა მისი პრიორიტეტი წყვეტაზე გახდება კანდიდატის პროგრამის პრიორიტეტზე უფრო მაღალი.

აღნიშნული მექანიზმი საშუალებას იძლევა მიღწეულ იქნეს პროცესორებისა და პროგრამების ურთიერთდამოუკიდებლობა და უზრუნველყოფს დროის თითოეულ მომენტში ყველაზე უფრო პრიორიტეტული პროგრამების შესრულებას არსებულ გამართულ პროცესორებზე.

ლექცია 15

15. საბორტო კომპიუტერების განვითარების პერსპექტივები

საბორტო კომპიუტერების განვითარების პერსპექტიული მიმართულებები შესაძლებელია დაიყოს შემდეგ ჯგუფებად:

- კლასიკური არქიტექტურის კომპიუტერების ელემენტური ბაზის სრულყოფა;
- კომპიუტერების აგებისათვის ახალი არქიტექტურული მიდგომების დამუშავება;
- საბორტო კომპიუტერების გამოყენების სფეროს გაფართოება ინფორმაციის ციფრული დამუშავების ახალი ალგორითმებისა და მეთოდების საფუძველზე;
- ახალი ფიზიკური პრინციპების გამოყენება მონაცემების დამუშავებისა და შენახვის შესრულებისათვის.

საბორტო კომპიუტერების განვითარების პერსპექტივები დაკავშირებულია მეხუთე თაობის კომპიუტერების დამუშავებასთან. მეხუთე თაობის კომპიუტერების დამუშავების ერთ-ერთ უმთავრეს და ძირითად ამოცანას წარმოადგენს ისეთი კომპიუტერის შექმნა, რომელიც გათვლილია ოპერატორთან (ეკიპაჟთან) დიალოგის რეჟიმში მოხერხებულ ურთიერთქმედებაზე და სიმარტივეზე ბუნებრივ ენაზე. მეხუთე თაობის კომპიუტერების სხვა განსაკუთრებულობას წარმოადგენს ამოცანების ლოგიკური გადაწყვეტის სისტემის არსებობა. ის უზრუნველყოფს კომპიუტერი აღჭურვოს უნარით მოახდინოს ინფორმაციის ასოციაციური მოძიება, შეძლოს ლოგიკური დასკვნისა და თვითსწავლების უზრუნველყოფა.

იდეალურ შემთხვევაში ამ კომპიუტერმა უნდა შეძლოს დასკვნების საერთო მასივიდან ამოიღოს ინფორმაციის ის მონაცემები, რომლებიც ერთმანეთთან დაკავშირებული არიან სხვადასხვა ნიშან-თვისებებით და აუცილებელია ლოგიკური დასკვნების ფორმირებისათვის. მეხუთე თაობის კომპიუტერები რეაგირებას მოახდენენ არამკაფიოდ დასმულ ამოცანებსა და საკითხებზე, იმავდროულად მოითხოვენ ადამიანებისგან (სპეციალისტებისგან) მათ უფრო მკაფიოდ ფორმულირებას ან თვითონ

დამოუკიდებლად შეძლებენ მასში ინფორმაციის უდიდესი მოცულობის ანალიზის საფუძველზე უტყუარი ლოგიკური დასკვნის გაკეთებას. მეხუთე თაობის კომპიუტერის შემადგენლობაში უნდა არსებობდეს კიდევ ერთი ფუნქციური სისტემა - ცოდნის ბაზა შესაბამისი მართვის სისტემით. ცოდნის ბაზიდან მიღებული ინფორმაციის საფუძველზე ლოგიკური გადაწყვეტის მანქანაში მოხდება ოპერატორის მიერ დასმული პრობლემის გადაწყვეტა, რომელიც შემდეგში დაექვემდებარება აზრობრივ ანალიზს. ასეთი ანალიზის მიზანია მიღებულ გადაწყვეტაში არარსებითი ინფორმაციის უგულებელყოფა და მასში იმ ფაქტებისა და წესების გამოყოფა, რომლებიც შემდგომში იქნება გამოყენებული სხვა ამოცანების გადაწყვეტისათვის. შედეგები, რომლებსაც ექნებათ საერთო ინტერესი კონკრეტულ სფეროში კონკრეტული ამოცანების გადაწყვეტისათვის დაგროვდებიან კომპიუტერის ცოდნის ბაზაში.

ივარადლება, რომ მეხუთე თაობის კომპიუტერები მწარმოებულობისა და ფუნქციური შესაძლებლობების თვალსაზრისით მნიშვნელოვნად გადააჭარბებს თანამედროვე კომპიუტერების შესაბამის მახასიათებლებს. გამოთვლითი სისტემა, რომელიც გამოიყენებს მეხუთე თაობის კომპიუტერს, წარმოდგენილი იქნება გამოთვლითი კომპიუტერების არაერთგვაროვანი ქსელის საშუალებით, რომელთანაც მიერთებული იქნება საერთო დანიშნულების უნივერსალური მაღალმწარმოებლური კომპიუტერები, კომპიუტერები მცირე მწარმოებლურობით, ასევე სპეციალიზებული გამომთვლელები და ყველა ისინი გამოიყენებენ ინფორმაციის პარალელურ დამუშავებას. მეხუთე თაობის კომპიუტერებისათვის ნიშანდობლივი იქნება პარალელური არქიტექტურის არსებობა.

საბოროტო კომპიუტერების სრულყოფას სხვა პერსპექტიულ მიმართულებას წარმოადგენს ახალი მასალებისა და ინფორმაციის დამუშავების ახალი პრინციპების გამოყენება კომპიუტერის აპარატული ნაწილის აგებისათვის. საბოროტო კომპიუტერების პერსპექტიული განვითარების თვალსაზრისით ყურადღებას იმსახურებს და შეგვიძლია გამოვყოთ ოპტიკური კომპიუტერები, კომპიუტერები ბიოტექნოლოგიების საფუძველზე (ნეიროკომპიუტერები) და კვანტური კომპიუტერები.

15.1. ოპტიკური კომპიუტერები

ოპტიკური პროცესორების შექმნისათვის მუშაობა დაიწყო გასული საუკუნის 80-იან წლებში. მუშაობის დაწყებას წინ უსწრებდა გარკვეული წარმატებები ოპტიკური კვანტური გენერატორების - ლაზერების შექმნის საქმეში. სუფთა ოპტიკური კომპიუტერის შექმნა დღესაც არაა მიღწეული გარკვეული მიზეზების გამო და ამჟამად ყველა რეალურად მოქმედი (არსებული) ოპტიკური კომპიუტერები წარმოადგენენ ოპტოელექტრონულ კომპიუტერებს.

ელექტრონული სქემების წარმოების ტექნოლოგიის სრულყოფის გარდა სამომავლო გამოთვლით ტექნიკაში ფეხს იკიდებს ახალი სამეცნიერო-ტექნიკური მიმართულება-ციფრული ოპტიკური გამოთვლითი ტექნიკა. ითვლება, რომ ის საშუალებას მოგვცემს მოხდეს გადასვლა ელექტრონული კომპიუტერებიდან ოპტიკურ კომპიუტერზე, რომლებიც აღჭურვილი იქნებიან განუზომლად დიდი სწრაფქმედებით, რაც განპირობებული იქნება ინფორმაციის მატარებლის სახით ფოტონების გამოყენების შედეგად. ჩვეულებრივ კლასიკურ კომპიუტერში ინფორმაციის ბიტის რანგში გამოიყენება ელექტრონი, უფრო სწორად ელექტრონის მუხტი. თუ მოხდება ელექტრონის შეცვლა ფოტონით, ხოლო მუხტისა ელექტრომაგნიტური ველის იტენსიურობით მიიღება ეგრეთწოდებული კლასიკური ოპტიკური კომპიუტერი.

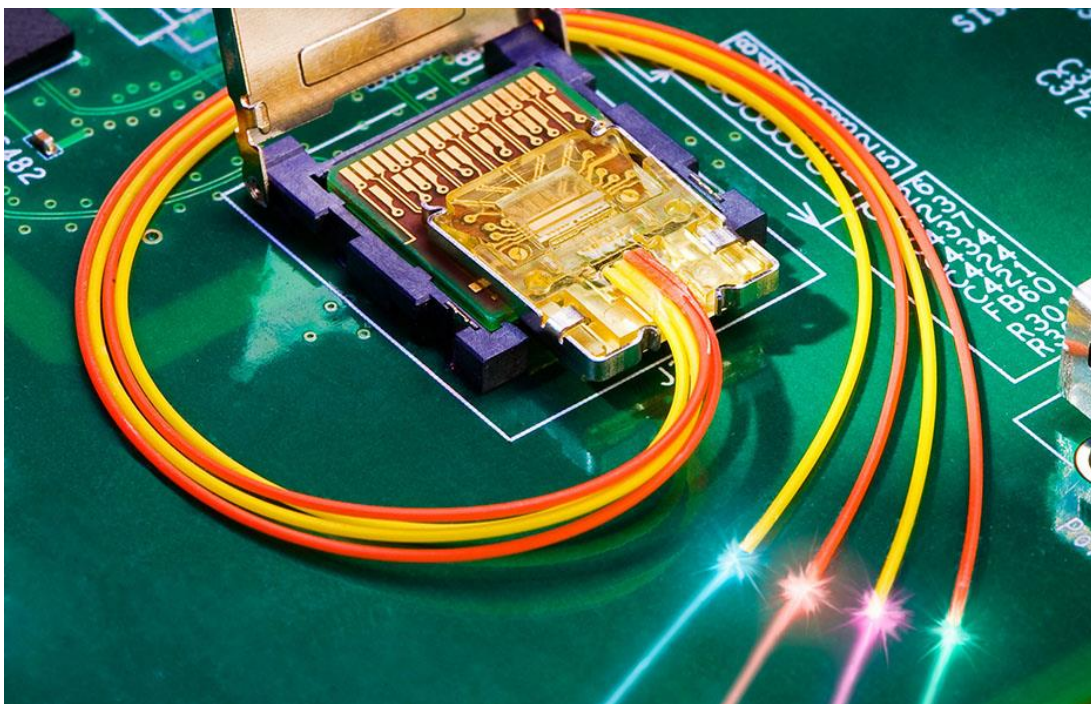
ოპტიკური კომპიუტერი - ეს არის კომპიუტერი, რომელიც დაფუძნებულია ოპტიკური პროცესორის გამოყენებაზე (სურ.15.1). ჩვეულებრივი ელექტრონული კომპიუტერებისგან განსხვავებით, რომლებიც დაფუძნებული არიან ელექტრონულ ტექნოლოგიებზე, ოპტიკურ კომპიუტერში ოპერაციები სრულდება ოპტიკური გამოსხივების ნაკადებზე მანიპულაციის გზით, რაც საშუალებას იძლევა მიღწეულ იქნეს გამოთვლების დიდი სისწრაფე და მწარმოებლურობა. ოპტიკურ კომპიუტერებში გამოთვლები წარმოებს ფოტონების დახმარებით, რომლებსაც ასხივებენ ლაზერები და შუქდიოდები.

ოპტიკური ტექნოლოგიის უპირატესობებია:

- მწარმოებლურობის პრინციპული ამაღლება;
- სქემის ელემენტების ზომების შესაძლო შემცირება;
- მოხმარებული სიმძლავრის შემცირება.

ოპტიკური კომპიუტერი - ეს არის ინფორმაციის დამუშავების მოწყობილობა, რომელიც იყენებს სინათლეს. სინათლე როგორც ელექტრომაგნიტური ტალღის

განსხვავებული განსაკუთრებული სახე ხასიათდება იმით, რომ სინათლის ტალღის სიხშირე რამდენიმე რიგით მაღალია თანამედროვე კომპიუტერულ ტექნიკაში გამოყენებულ ელექტრული სიგნალების და ტალღების სიხშირეზე. ასე მაგალითად, ელექტრული ტალღა, რომელიც გამოიყენება დღეს რადიოტექნიკაში ასრულებს დაახლოებით 100 ათას რხევას წამში, მაშინ როცა სინათლის ტალღას გააჩნია სიხშირე, რომელიც 10-100 მილიონჯერ აღემატება ამ სიხშირეს. ამიტომ მისი დახმარებით დროის ფიქსირებულ ინტერვალში შესაძლებელია გადაიცეს სიგნალების უფრო დიდი რიცხვი, ე.ი. შესაბამისად ინფორმაცია. გარდა ამისა, იმდენად რამდენადაც სინათლის ტალღის სიგრძე მეტისმეტად მცირეა, ამდენად შესაძლებელია ინფორმაციის დამუშავება მოხდეს უაღრესად მაღალი სიჩქარით. უნდა აღინიშნოს, რომ ოპტიკური კომპიუტერი „სუფთა“ სახით ჯერ კიდევ არ არსებობს. მოცემულ მომენტში არსებობს მხოლოდ ელექტრონულ-ოპტიკური კომპიუტერი



სურათი 15.1

ოპტიკურ კომპიუტერში ინფორმაციის შეტანის ბლოკში გამოიყენება ოპტიკური გადამწოდები. ესენი არის მოწყობილობები რომლებშიც სინათლის დახმარებით განისაზღვრება ინფორმაციის რაოდენობრივი მახასიათებლები, მაგალითად საგნის არსებობა/არ არსებობა, მისი ფორმის განსაკუთრებულობა, სიჩქარე, ტემპერატურა და ა. შ. ოპტიკურ გადამწოდებს, განსხვავებით სხვა გადამწოდებისგან არ ჰქირდებათ უშუალო

კონტაქტი დასაკვირებელ ობიექტთან. გარდა ამისა ოპტიკური გადამწოდები გამოირჩევიან მაღალი მგრძნობიარობით და სწრაფქმედებით.

სინათლის დახმარებით ინფორმაციის ჩაწერის ტექნიკამ (ოპტიკური მეხსიერების შექმნა) უკანასკნელ წლებში მიიპყრო ძალზე დიდი ყურადღება. **დიდი ყურადღება ექცევა ასევე ჰოლოგრაფიული მეხსიერების დამუშავებას.** ყველაზე უფრო გავრცელებულ დამუშავებას ამ სფეროში წარმოადგენს ოპტიკური დისკები.

ოპტიკური დისკიდან ინფორმაციის ამოკითხვის პრინციპი მდგომარეობს დისკის ზედაპირის დასხივებაში ლაზერული სხივით და მის ზედაპირიდან არეკლილი სხივის დახმარებით ინფორმაციის მოხსნაში. მომავალში დიდი ალბათობით, ოპტიკური მეხსიერება გამოდევნის მაგნიტურ მეხსიერებას, რომელიც გამოიყენება კლასიკურ კომპიუტერებში.

ოპტიკური კომპიუტერის ინფორმაციის კავშირისა და გადაცემის ბლოკში ინფორმაციის გადაცემისთვის გამოიყენება ოპტიკური ბოჭკო. ინფორმაციის გადაცემა ოპტიკური ბოჭკოს საშუალებით მდგომარეობს მასში სინათლის გავრცელებაში ფოტონების საშუალებით, რომლებიც წარმოადგენენ ინფორმაციის იდეალურ გადამცემებს. ოპტიკური სიგნალი ბოჭკოში შეიძლება გადატანილი იქნეს ათობით კილომეტრებზე პრაქტიკულად მილევისა და დამახინჯების გარეშე. სინათლეს არა მარტო გააჩნია უნარი გადასცეს ინფორმაცია სიჩქარით, რომელიც მთელი რიგით აღემატება ელექტრული სიგნალის გადაცემის სიჩქარეს, არამედ ასევე უზრუნველყოფს დროის ერთეულში სინათლის დახმარებით გადაცემული ინფორმაციის გაზრდილ მოცულობასაც. რადგან ბოჭკოვან-ოპტიკური კაბელები არ ასხივებენ რადიოდიაპაზონში, ამიტომ შეუძლებელია მოსმენილ იქნეს მათ მიერ გადაცემული ინფორმაცია. ინფორმაციის გაცვლის ოპტიკური საშუალებების გამოყენება იძლევა უფრო მეტ უპირატესობას, ვიდრე მონაცემების ჩვეულებრივი ელექტრული გადაცემა.

ოპტიკური გამოსხივების გამოყენებას ინფორმაციის მატარებლის სახით გააჩნია მთელი რიგი პოტენციური უპირატესობები ელექტრულ სიგნალებთან შედარებით, სახელდობრ:

1. სინათლის ნაკადებს, განსხვავებით ელექტრულისგან, შეუძლიათ ერთმანეთის გადაკვეთა;

2. სინათლის ნაკადები შესაძლებელია ლოკალიზებულ იქნეს ნანომეტრულ ზომამდე და გადაცემულ იქნეს თავისუფალ სივრცეში;

3. სინათლის გავრცელების სიჩქარე მაღალია ელექტრული სიგნალის გავრცელების სიჩქარეზე;

4. სინათლის ნაკადების ურთიერთქმედება არახაზოვან გარემოში გადანაწილებულია მთელ გარემოში, რაც იძლევა თავისუფლების ახალ ხარისხებს (ელექტრონულ სისტემებთან შედარებით) კავშირის ორგანიზაციაში და პარალელური არქიტექტურის შექმნაში.

ზოგადად, პარალელური არქიტექტურის დიდი რაოდენობის შექმნა ნახევარგამტარული კომპიუტერებისგან განსხვავებით წარმოადგენს ოპტიკური კომპიუტერის ძირითად ღირსებას, ის საშუალებას იძლევა გადალახულ იქნეს ინფორმაციის შეზღუდვები სწრაფქმედებისა და პარალელური დამუშავების თვალსაზრისით, რაც ესოდენ ნიშანდობლივია დღეს თანამედროვე კომპიუტერებისთვის. ოპტიკური ტექნოლოგიის განვითარება მნიშვნელოვანია არა მარტო ოპტიკური კომპიუტერების შექმნისათვის, არამედ ის ასევე მნიშვნელოვანია ოპტიკური კომუნიკაციებისა და Internet ქსელის განვითარებისათვის.

ოპტიკური კომპიუტერის ძირითადი ღირებულებებია: უნარი, შეასრულოს ოპერაციები ორგანოზომილებიან ოპერანდებთან; ოპერაციების შესრულების მაღალი სიჩქარე, რომელიც განისაზღვრება სინათლის სხვისი გავლის დროით პროცესორის შესასვლელიდან გამოსასვლელამდე; ინფორმაციის დამუშავების საშუალებების შემადგენლობაში დიდი ტევადობის და სწრაფქმედების ჰოლოგრაფიული (ოპტიკური) დამხსომებელი მოწყობილობების გამოყენება; ინტერფეისის სახით ბოჭკოვან-ოპტიკური არხების გამოყენება ინფორმაციის შეტანა-გამოტანის და შიგა მანქანური კავშირების რეალიზაციისათვის.

ოპტიკური კომპიუტერის მთავარ ბლოკს წარმოადგენს ოპტიკური ლოგიკური მატრიცული პროცესორი, რომელიც ასრულებს არითმეტიკულ და ლოგიკურ ოპერაციებს. ოპტიკურ პროცესორთან შერწყმულია ოპერატიული დამხსომებელი მოწყობილობა და შეტანა-გამოტანის არხები. კონსტრუქციულად ყველა ისინი განლაგებულია ერთ დიდ ოპტიკურ ორობით ვენტილურ მატრიცაზე (კრისტალზე), რომელიც აღჭურვილია სივრცითობის და ორგანოზომილებიანი ინფორმაციის დამუშავების

პარალელურობის თვისებებით, რომლის (ორგანოზომილებიანი ინფორმაციის) შეტანა და გამოტანა ხორციელდება პარალელურად ინფორმაციული გვერდების საშუალებით.

შემაერთებული არხების (კავშირების) სახით დაგეგმილია ოპტიკური ბოჭკოების, ინტეგრალურ-ოპტიკური ტალღის და სინთეზირებული ჰოლოგრამების გამოყენება.

Enlight 256 – ეს არის Lenslet კომპანიის მიერ შექმნილი პირველი ოპტიკური პროცესორი DSP (Digital Signal Processor) (სურ.15.2), რომელიც 3 - ჯერ უფრო უკეთესია ვიდრე ელექტრონული DSP პროცესორები. უფრო ზუსტად რომ ვთქვათ Enlight 256 – ეს ჰიბრიდული ოპტიკური პროცესორია, რომელშიც შეცვლილია მხოლოდ ბირთვი, ხოლო შესასვლელი და გამოსასვლელი ინფორმაცია წარმოდგება ელექტრონული სახით. ბირთვი შედგება 256 VCSEL ლაზერებისგან, სინათლის სივცითი მოდულატორისგან , ლინზების ნაკრებისგან და მიმღებებისგან. პროცესორის მწარმოებლურობა შეადგენს 8 ტრილიონ ოპერაციას წამში. ერთ ტაქტში (8 ნანოწამი) პროცესორი ამრავლებს 256 ბაიტს რიცხვს 256 x 256 მატრიცაზე.



სურ.15.2

ოპტიკურ კავშირებს ელექტრულ კავშირებთან შედარებით გააჩნია მთელი რიგი უპირატესობანი: დინამიკური ოპტიკური კომპონენტების დახმარებით გადაპროგრამირების უნარის შესაძლებლობა, ტევადური დატვირთვებისა და ურთიერთდაბრკოლებათა გავლენის არ არსებობა, კავშირების მოქნილობა და სხვ.

სისტემის გადაპროგრამირება გადაწყობადი კავშირების სტრუქტურით საშუალებას იძლევა ორგანიზება გაუკეთდეს კომპიუტერული არქიტექტურის ახალ ტიპებს, რაც არსებითად აამაღლებს ინფორმაციის დამუშავების ეფექტურობას.

კიდევ ერთხელ ხაზგასმით უნდა ავლნიშნოთ, რომ ოპტიკური კომპიუტერი ციფრული გამოთვლების შესასრულებლად იყენებს ფოტონებს ხილულ შუქზე ან ინფრაწითელ სხივებში, და არა ელექტრულ დენს. ელექტრული დენის გადინების

სიჩქარე სინათლის სიჩქარის მხოლოდ 10%-ს შეადგენს. ეს ზღუდავს მონაცემთა გაცვლას დიდ დისტანციებზე და ეს არის ერთ-ერთი ფაქტორი, რამაც გამოიწვია ოპტიკური ბოჭკოების ევოლუცია. ხილული და ინფრაწითელი სხივების ზოგიერთი უპირატესობის გამოყენებით მოწყობილობისა და კომპონენტის მასშტაბით, მომავალში შეიძლება განვითარდეს კომპიუტერი, რომელსაც შეეძლება შეასრულოს გამოთვლითი ოპერაციები ასეულჯერ უფრო სწრაფად, ვიდრე ამას აკეთებენ დღეს ჩვეულებრივი, კლასიკური ელექტრონული კომპიუტერები.

თუმცა უნდა აღინიშნოს, რომ პრობლემები, რომლებიც დაკავშირებულია ოპტიკური კომპიუტერების შექმნასთან ჯერ კიდევ ბევრია. ერთის მხრივ, თუ საჭიროა არაწრფივი ოპერაციების შესრულება, მაგალითად, ლოგიკური ოპერაციებისა „და“ ან „ან“ (რომლებიც ხშირად ვხვდებიან კომპიუტერულ პროგრამებში), მაშინ უზრუნველყოფილ უნდა იქნეს ფოტონების ურთიერთქმედება ერთმანეთთან. საერთოდ დაუმუხტავი ნაწილაკები ურთიერთქმედებენ ცუდად, ამისათვის ჩვეულებრივ საჭიროა დიდი ინტენსიურობები, რომლებიც თავის მხრივ იწვევენ გახურებას და ამცირებენ ენერგოეფექტურობას. მეორეს მხრივ, ფოტონები მუდმივად ცდილობენ სადღაც გაფრინდნენ, ამიტომ ძალზე ძნელია შეიქმნას მოწყობილობა, რომელსაც შეეძლება შეინახოს ოპტიკური ინფორმაცია, თუნდაც რაღაც მინიმალური დროის განმავლობაში (თუნდაც მიკროწამები იქნებოდა ძალიან კარგი, მაგრამ ასეთი მოწყობილობები სამწუხაროდ პრაქტიკულად არ არსებობენ). აუცილებელია შეიქმნას ფოტონური სისტემის ინტერფეისები დამხსომებელი მოწყობილობებით (მაგნიტური ან ელექტრონული), რომლებიც იქნებოდნენ სწრაფი და ენერგოეფექტურები. ასევე ერთ-ერთ პრობლემას წარმოადგენს მინიატურიზაცია. საქმე იმაშია, რომ ერთი ელექტრონული ტრანზისტორის ზომა დაახლოებით 10 ნანომეტრია. ამიტომ ჩიპის ზედაპირის სანტიმეტრზე შესაძლებელია „ჩაეწყოს“ ასობით მილიონი ტრანზისტორი, რომელიც უზრუნველყოფს ელექტრონული კომპიუტერების სწრაფქმედებას, ოპტიკური ელემენტების ზომები განისაზღვრება ფოტონის ტლდის სიგრძით - რომელიც ერთი მიკრონის რიგისაა. ოპტიკური ტრანზისტორები არ იქნებიან ისეთივე კომპაქტურები, როგორც ელექტრონულები, რაც რა თქმა უნდა ზღუდავს მათი გამოყენების სფეროებს. მიუხედავად ამისა, ოპტიკური ჩიპები დღეს უკვე აქტიურად გამოიყენება ოპტიკურბოჭკოვან ქსელებში ინფორმაციის დამუშავებისათვის და მრავალ სხვა

კონკრეტულ ამოცანებში, რომლებშიც მათი გამოყენება ელექტრონულ ანალოგებთან შედარებით უპირატესია.

ოპტიკური კომპიუტერები ერთ-ერთი სამომავლო ტენდენციაა. როგორც ვიცით დღეს არსებული კომპიუტერებში ბიტის შემნახველი არის ტრანზისტორი, ტრანზისტორის რაოდენობა კი ასეულ მილიონებს აღწევს. თანამედროვე კომპიუტერების ზრდა ამ თემის ირგვლივ ეყრდნობა მურის კანონს, რომელიც გულისხმობს, რომ ტრანზისტორების რაოდენობა ყოველ ორ წელიწადში ორმაგდება მიკროჩიპებში, ამის ფონზე კი ღირებულება ნახევრდება. ამის მიუხედავად არის ვარაუდები რომ ტრანზისტორებს დიდი განვითარების პერსპექტივა აღარ აქვთ. მურის კანონის თანახმად ორჯერ გაზრდილი ტრანზისტორები საბოლოოდ გვამღევს ექსპონენციალურ ზრდას, ამ ზრდის გზის ბოლოს კი შეიძლება დიდი რაოდენობით ტრანზისტორებმა საერთოდ შეაფერხონ კიდევ პროცესორის სიჩქარე. გარდა ამისა ტრანზისტორს აქვს ფიზიკური ზომის შეზღუდვა. ასევე რაც მეტია ტრანზისტორი, მით მეტი ენერგიაა საჭირო, მეტი ენერგია კი განაპირობებს მაღალ ტემპერატურას, რაც მუშაობის შემაფერხებელი ფაქტორია. ამ ყველაფრის თავიდან აცილების ერთ-ერთ ალტერნატივას წარმოადგენს სრულიად ოპტიკური კომპიუტერები ან ელექტრო-ოპტიკური კომპიუტერების ჰიბრიდული ვარიანტი. როგორც უკვე ავღნიშნეთ ოპტიკურ კომპიუტერში გამოთვლები მოხდება ლაზერების ან ფოტოდiodების მიერ წარმოებული ფოტონებით. ეს კი აჩენს შესაძლებლობას მიღებულ იქნეს განუზომლად დიდი სწრაფმედება, ანუ ფოტონები იძლევიან უფრო მეტ გამტარობას ვიდრე ელექტრონები. სრულად ოპტიკური კომპიუტერის გამოსვლამდე კომერციაში განიხილება სხვა უფრო რეალური ვარიანტი, ეს არის ელექტრონულ-ოპტიკური კომპიუტერის ჰიბრიდული ვარიანტი. აქ მოხდება ინფორმაციის დამუშავება ორობით ფორმაში, რაც შესაძლებლობას იძლევა მოხდეს ოპტიკური კომპონენტების ინტეგრირება ტრადიციულ კომპიუტერში. ოპტოელექტრონული მოწყობილობები მოიხმარენ თავიანთი ენერგიის 30%-ს, რაც ნიშნავს იმას, რომ ისინი გარდაქმნიან ელექტრულ ენერგიას ფოტონებად და პირიქით. თანამედროვე კომპიუტერის მთავარი შემადგენელი კომპონენტი არის ტრანზისტორი. მისი ოპტიკურით ჩანაცვლებისთვის არის საჭირო ოპტიკური ტრანზისტორი, რომლის მიღება არაწრფივი რეფრაქციის ინდექსის მქონე მასალებით არის შესაძლებელი. ეს გულისხმობს პროცესს როდესაც ინტენსიური ლაზერული სხივი გადის მასალაში. ამ

დროს სხივის ელექტრულ ველს შეუძლია გამოიწვიოს მასალის გარდატეხის ინდექსის ცვლილება, რომელიც პროპორციულია სხივის ინტენსიურობის. გამოთვლითმა სისტემამ რომ კარგად შეძლოს ფუნქციონირება მოითხოვება გარკვეული საჭიროებები. ეს საჭიროებები ოპტიკური კომპიუტერის შემთხვევაში არის: ოპტიკური პროცესორი, ოპტიკურ ბოჭკოვანი კაბელი და ოპტიკური მეხსიერება. ოპტიკური კომპიუტერების მთავარი კომპონენტია ოპტიკური პროცესორი, ის ასრულებს არითმეტიკულ-ლოგიკურ ოპერაციებს. პროცესორთან არის სტანდარტულად კავშირში ოპერატიული დამხსოვრებული მოწყობილობა და ინფორმაციის შემტანი და გამომტანი არხები. ეს ყველაფერი განლაგებულია კრისტალზე. ამ კრისტალს აქვს სივრცითობის და ორგანოზომილებიანი ინფორმაციის დამუშავების პარალელურობის თვისება.

და ბოლოს კიდევ ერთხელ ხაზი გავუსვით ოპტიკური ტექნოლოგიის უპიტატესობას ელექტრონულ ტექნოლოგიასთან შედარებით:

- გამოსახულების პარალელური გადაცემის და დამუშავების შესაძლებლობა. პარალელური მუშაობის შესაძლებლობა სხვადასხვა სიგრძის ტალღების შემთხვევაში;
- გამჭვირვალე გარემოს გამოყენების შესაძლებლობა ინფორმაციის შენახვის, დამუშავების და კომუტაციისათვის (გამჭვირვალე გარემოში ოპტიკური სხივით კოდირებული ინფორმაცია, შესაძლებელია დამუშავდეს ენერგიის დანახარჯის გარეშე);
- ოპტიკურ კომპიუტერში ინფორმაციული არხების კომუტაცია ხორციელდება დიდი სიჩქარით და გამოირჩევა რელიზაციის სიმარტივით იმის გამო, რომ სინათლის სხივები ცარიელ გარემოში ვრცელდებიან ისე, რომ არ ურთიერთქმედებენ ერთმანეთთან;
- ოპტიკური სისტემა არაფერს არ ასხივებს გარემოში, რაც უზრუნველყოფს კომპიუტერის დაცვას ინფორმაციის დაჭერის თვალსაზრისით;
- ოპტიკური პროცესორის სქემა, რომელიც რელიზებას უკეთებს ორი გამოსახულების შეკვრის ოპერაციას უდევს საფუძვლად ასოციაციური მეხსიერების და სახეების ამოცნობის მრავალი მოწყობილობების მუშაობას.

ოპტოელექტრონიკის სფეროში ბოლო ათწლეულში მიღწეულმა წარმატებებმა შესაძლებელი გახადა საკმარისად მაღალ დონეზე დამუშავებულიყო ოპტიკური

კომპიუტერის ყველა ძირითადი კომპონენტის წარმოების ტექნოლოგია. მისი გამოყენების ძირითადი სფეროებია - სამხედრო ტექნიკა, ინფორმაციის დაცვა, ხელოვნური ინტელექტი, მანქანური მხედველობა, სამეცნიერო კვლევები, კრიმინოლოგია და სხვ.

ოპტიკური ტექნოლოგიის განვითარებაში მიღწეული წარმატებების მიუხედავად, სრულფასოვანი წმინდა ოპტიკური კომპიუტერის შექმნა ჯერ კიდევ შორსაა.

ოპტიკურ კომპიუტერებს აქვთ პოტენციური იმისა, რომ მნიშვნელოვანი როლი შეასრულონ მომავალში ავიაციაში და მოპოვონ უპირატესობა ტრადიციულ ელექტრონულ კომპიუტერებთან შედარებით. ოპტიკური გამოთვლა გულისხმობს ფოტონების (სინათლის ნაწილაკების) გამოყენებას ელექტრონების ნაცვლად ინფორმაციის დასამუშავებლად და გადასაცემად. მიუხედავად იმისა, რომ ოპტიკური კომპიუტერები ჯერ კიდევ ექსპერიმენტულ ეტაპზეა და ჯერ კიდევ არ არის ფართოდ დანერგილი, მათ აქვთ რამდენიმე პოტენციური გამოყენება ავიაციაში. აქ მოცემულია რამდენიმე სფერო, სადაც ოპტიკურ კომპიუტერებს შეუძლიათ გავლენა მოახდინონ:

მონაცემთა დამუშავება და კომუნიკაცია: ოპტიკურ კომპიუტერებს შეუძლიათ რთული გამოთვლების შესრულება და მონაცემთა დიდი მოცულობის დამუშავება წარმოუდგენლად მაღალი სიჩქარით. ავიაციაში, ამან შეიძლება გააუმჯობესოს სხვადასხვა ამოცანები, როგორიცაა ამინდის პროგნოზირება, ფრენის ტრაექტორიის ოპტიმიზაცია და რეალურ დროში მონაცემების ანალიზი თვითმფრინავის სენსორებიდან და ინსტრუმენტებიდან.

უსაფრთხო კომუნიკაცია: ოპტიკური გამოთვლა გვთავაზობს მაღალ უსაფრთხო საკომუნიკაციო სისტემების პოტენციალს. ძნელია ოპტიკური სიგნალების დაჭერა ან შეხება, რამაც შეიძლება გააძლიეროს მონაცემთა გადაცემის უსაფრთხოება თვითმფრინავებს, საჰაერო მოძრაობის კონტროლსა და სახმელეთო სისტემებს შორის.

რადარი და გამოსახულების სისტემები: ოპტიკურ გამოთვლას შეუძლია გააუმჯობესოს ავიაციაში გამოყენებული რადარის და გამოსახულების სისტემების მუშაობა. ოპტიკური კომპიუტერების სიჩქარისა და პარალელური დამუშავების შესაძლებლობების გამოყენებით, შესაძლებელია ამ სისტემების გარჩევადობის,

დიაპაზონის და სიზუსტის გაზრდა, რითაც გაუმჯობესდება პილოტებისა და საჰაერო მოძრაობის კონტროლერების სიტუაციური ცნობიერება.

ფრენის სიმულაცია და სწავლება: ოპტიკური კომპიუტერების უნარმა, დაამუშავოს დიდი რაოდენობით მონაცემები სწრაფად და ზუსტად, შეიძლება გააძლიეროს ფრენის სიმულატორები და სასწავლო სისტემები. ამან შეიძლება უზრუნველყოს უფრო რეალისტური და ჩამირული სასწავლო გამოცდილება პილოტებისთვის, რაც მათ საშუალებას მისცემს ივარჯიშონ რთული სცენარები და საგანგებო სიტუაციები უსაფრთხო და კონტროლირებად გარემოში.

ოპტიკური სენსორები: ოპტიკური გამოთვლები შეიძლება გამოყენებულ იქნას მოწინავე ოპტიკურ სენსორებთან ერთად, რათა მოხდეს თვითმფრინავის სისტემების უფრო ზუსტი და ეფექტური მონიტორინგი, როგორცაა ძრავის მუშაობა, სტრუქტურული მთლიანობა და საწვავის მოხმარება. ამან შეიძლება გამოიწვიოს თვითმფრინავების ტექნიკური და საიმედოობის გაუმჯობესება, შეფერხების დროის შემცირება და უსაფრთხოების გაძლიერება.

მნიშვნელოვანია აღინიშნოს, რომ ოპტიკური გამოთვლები ჯერ კიდევ საავიაციო განვითარებადი ტექნოლოგიაა. საჭიროა შემდგომი კვლევა, განვითარება და ტესტირება ტექნიკური გამოწვევების დასაძლევად და ოპტიკური გამოთვლითი სისტემების საიმედოობის, მასშტაბურობისა და ეკონომიურობის უზრუნველსაყოფად. თუმცა, მუდმივი მიღწევებით, ოპტიკურ კომპიუტერებს შეუძლიათ წვლილი შეიტანონ საავიაციო ტექნოლოგიებისა და ოპერაციების მნიშვნელოვან გაუმჯობესებაში.

15.2. ნეიროკომპიუტერები

ნეიროკომპიუტინგი - ეს მეცნიერული მიმართულებაა, რომელიც დაკავებულია მეექვსე თაობის გამოთვლითი სისტემების - ნეიროკომპიუტერების დამუშავებით, რომლებიც შედგებიან მარტივი გამოთვლითი ელემენტებისგან - ნეირონებისგან. დიდი რაოდენობის პარალელურად მომუშავე ელემენტები დაკავშირებული არიან ერთმანეთთან და ქმნიან ნეირონულ ქსელს. ისინი ასრულებენ ერთი სახის გამოთვლით

მოქმედებებს და არ მოითხოვენ გარე მართვას. პარალელურად მომუშავე გამოთვლითი ელემენტების დიდი რიცხვი უზრუნველყოფენ მათ მაღალ სწრაფქმედებას.

ნეიროკომპიუტერები იძლევიან საშუალებას მაღალი ეფექტურობით გადაწყვიტონ მთელი რიგი ინტელექტუალური ამოცანები. ეს ამოცანებია ადაპტური მართვა, სახეების ამოცნობა, პროგნოზირება, დიაგნოსტიკა და ა.შ.

ნეიროკომპიუტერები წინა თაობების კომპიუტერებისგან განსხვავდებიან არა მარტო უბრალოდ დიდი შესაძლებლობებით, არამედ მათში პრინციპულად შეცვლილია კომპიუტერის გამოყენების ხერხი. ასეთ კომპიუტერებში პროგრამირება შეცვლილია სწავლებით - ნეიროკომპიუტერი სწავლობს ამოცანების გადაწყვეტას.

სწავლა - ეს არის კავშირების დონეების კორექტირება, რომლის შედეგად შესასვლელი მოქმედება იწვევს შესაბამისი გამოსავლელი სიგნალის ფორმირებას. სწავლის შემდეგ ქსელს შეუძლია გამოიყენოს შეძენილი სიახლეები ახალი შესასვლელის მიმართ. დაპროგრამებიდან სწავლებაზე გადასვლისას მაღლდება ინტელექტუალური ამოცანების გადაწყვეტის ეფექტურობა.

ნეირონულ ქსელებში გამოთვლები არსებითად განსხვავდებიან ტრადიციული გამოთვლებისგან. მათი მაღალი პარალელურობის გამო, ისინი შეგვიძლია განვიხილოთ როგორც კოლექტიური მოვლენა. ნეირონულ ქსელში არ არის ლოკალური არეები, რომლებშიც დამახსოვრებული იქნება კონკრეტული ინფორმაცია. მთელი ინფორმაცია დაიმახსოვრება მთელ ქსელში.

ნეიროკომპიუტერების განვითარების ბიძგს ხელი შეუწყო ბიოლოგიურმა გამოკვლევებმა. ნეირობიოლოგიის მონაცემების მიხედვით ადამიანის ნერვული სისტემა შედგება ცალკეული უჯრედებისგან - ნეირონებისგან. ადამიანის ტვინში მათი რიცხვი აღწევს 100 მილიარდამდე. ნეირონებს შორის დაახლოებით 60 ტრილიონი კავშირია და ასრულებენ შედარებით მარტივ მოქმედებებს. ნეირონის ამუშავების დროა 2 – 5 მწმ. ყველა ნეირონის ერთობლივი მუშაობა განაპირობებს ტვინის რთულ მუშაობას, რომელიც რეალურ დროში წყვეტს ურთულეს ამოცანებს.

ნეიროკომპიუტერების განსხვავება წინა თაობების გამოთვლითი მოწყობილობებისგან შემდგება:

- ძალიან დიდი რიცხვის მარტივი გამოთვლითი სიტემების პარალელური მუშაობა უზრუნველყოფს უდიდეს სწრაფქმედებას;
- ნეირონულ ქსელს გააჩნია სწავლის უნარი, რომელიც ხორციელდება ქსელის პარამეტრების აწყობის გზით;
- მაღალია ნეირონული ქსელების დაბრკოლებამდგრადობა და მტყუნება-მდგრადობა;
- ცალკეული ნეირონების მარტივი აგებულება საშუალებას იძლევა შეიქმნას ინფორმაციის დამუშავების ახალი ფიზიკური პრინციპები ნეირონული ქსელების აპარატული რეალიზაციისათვის.

ნეიროკომპიუტერები ლოგიკის ტიპის მიხედვით არსებობს ანალოგური, ციფრული და ჰიბრიდული. ალგორითმის რეალიზაციის ტიპის მიხედვით ისინი არსებობენ როგორც სრულად აპარატული რეალიზაციის ალგორითმების, ისე პროგრამულ-აპარატული ალგორითმების სახით (როდესაც ნეიროალგორითმი მთელი ფუნქციონირების განმავლობაში ინახება მუდმივ მეხსიერებაში (ROM)).

თანამედროვე ნეიროკომპიუტერებისათვის იქმნება სპეციალიზებული ნეირო-პროცესორები, ისინი არიან 64 ბიტის და მათი დამზადების პროცესი ეყრდნობა ორ თანამედროვე არქიტექტურას: VLIW(გრძელი მანქნაური ბრძანება), SIMD (ერთჯერადი ინსტრუქცია, მონაცემთა მრავალჯერადი ნაკადი).

ნეიროკომპიუტერები შექმნილია სპეციფიკური პრობლემების გადასაჭრელად, რომელიც დაკავშირებულია არაწრფივ ლოგიკასთან და თვითორგანიზაციის თეორიასთან.

ნეიროკომპიუტერების დანერგვა და განვითარება იწყება 1950-იანი წლებიდან და ის იმ დოისათვის არსებულ ციფრულ კომპიუტერებთან ერთად ფუნქციონირებდა. რთული ფორმალური და არაფორმალური ამოცანების რაოდენობის ზრდამ განაპირობა,

ნეიროკომპიუტერების პროგრამების მასშტაბების ზრდა, ასეთი ტიპის ამოცანების გადაწყვეტა ჩვეულებრივი ტიპის კომპიუტერებზე თითქმის შეუძლებელი იყო.

1943 წელს უ. მაკალატმა და უ. პიტსმა მოახერხეს და შექმნეს პირველი ხელოვნური ნეირონის მოდელი და ჩამოაყალიბეს ძირითადი თეორიები თავის ტვინის ფუნქციონირების შესახებ.

1949 წელს კი დ.ჰერბმა უკვე გამოთქვა მოსაზრება, თავის ტვინში არსებული ნეირონების შესაძლო შეერთებაზე, რამაც ხელი შეუწყო მას ჩამოეყალიბებინა ნეირონული სწავლების კანონები.

მსოფლიოში პირველ ნეიროკომპიუტერად წოდებული “MARK 1” ანუ პერსეპტონი (ტვინის მიერ აღქმის მათემატიკური და კომპიუტერული მოდელი) 1958 წელს შექმნა ამერიკელმა ნეიროფიზიოლოგმა ფ.როზენბლანტმა.

პერსეპტონი წარმოადგენს ნერვული ქსელის ერთ-ერთ პირველ მოდელს. როზენბლანტის ნეიროკომპიუტერში წარმოდგენილი პერსეპტონი იყო ერთშრიანი, ამიტომ ის ვერ ახერხებდა ბევრი პრაქტიკული ამოცანის გადაწყვეტას.

ერთშრიანი პერსეპტონის შეზღუდულ შესაძლებლობებზე იმ დროისთვის მრავალი მეცნიერი საუბრობდა, რამაც გამოიწვია ხელოვნური ნეირონული ქსელების მიმართ ინტერესის განელება გარკვეული დროის განმავლობაში, თუმცა ამ დროის მანძილზეც მეცნიერთა გარკვეული ჯგუფი იკვლევდა ხელოვნურ ნეირონულ ქსელებს და 80 -იანი წლების დასაწყისში ისინი მივიდნენ იმ დასკვნამდე, რომ მრავალშრიანი პერსეპტონის გამოყენების შემთხვევაში შესაძლებელი გახდებოდა მრავალი სხვადასხვა სახის პრაქტიკული ამოცანის მარტივად გადაწყვეტა, ვიდრე ეს შესაძლებელი იყო ერთშრიანი პერსეპტონის შემთხვევაში.

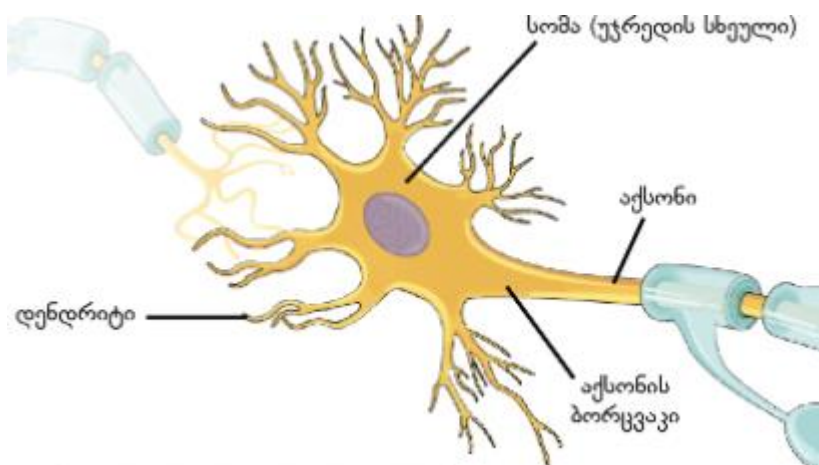
ნეიროკიბერნეტიკის (ტექნიკური საშუალებების შექმნა ნეირონულ პრინციპებზე) იდეის განხორციელება ინტენსიურად დაიწყო 1980 - იანი წლებიდან, ხოლო 1987 წლისთვის ჩატარდა ოთხი უმსხვილესი სამეცნიერო შეკრება ხელოვნური ნეირონული ქსელების შესახებ, რასაც მოჰყვა მრავალი მეცნიერის მიერ უმნიშვნელოვანესი სამეცნიერო ნაშრომების გამოცემა, ასეთი ტიპის ნაშრომების გამოქვეყნებამ ძალიან დააჩქარა ამ სფეროს განვითარება, ამასთან ერთად მნიშვნელოვანი როლი ითამაშა

ტექნოლოგიურმა პროგრესმა, რამაც მეცნიერებს საშუალება მისცა ეწარმოებინათ უფრო ზუსტი გამოთვლები და დაეხვეწათ არსებული გამოთვლითი აპარატურა.

ამჟამად ნეიროკომპიუტინგის ყველაზე უფრო მასობრივ მიმართულებას წარმოადგენს ნეირონული ქსელების მოდელირება ჩვეულებრივ კომპიუტერებზე, უპირველეს ყოვლისა პერსონალურ კომპიუტერებზე. ქსელების მოდელირება მიმდინარეობს მათი მეცნიერული გამოკვლევის მიზნით, პრაქტიკული ამოცანების გადაწყვეტისათვის, ასევე ელექტრონული და ოპტოელექტრონული ნეიროკომპიუტერების პარამეტრების მნიშვნელობების განსაზღვრისათვის.

ბუნებრივი ნეირონი

ბუნებრივი ნეირონი წარმოადგენს ნერვული სისტემის უჯრედს და მისი განმასხვავებელი ნიშანი სხვა უჯრედებთან შედარებით არის ის, რომ მას გააჩნია მრავალი გამოთვლითი ფუნქცია. ნეირონები გვხვდება სხვადასხვაგვარი, თუმცა ამ განსხვავებების მიუხედავად ისინი ერთნაირი დანიშნულების ელემენტებისაგან შედგებიან, ესენია: სომა (უჯრედის სხეული), დენდრიტები და აქსონები(ნახ.15.3).



ნახ.15.3

სომა მემბრანულ გარსშია მოქცეული. არეგულირებს უჯრედში მიმდინარე პროცესებს, მართავს ნეირონის ენერგიის ხარჯვას და არეგულირებს უჯრედში მიმდინარე სხვა პროცესებს. უჯრედის სხეულის გარე მემბრანას შეუძლია ნერვული

იმპულსების გენერირება, რომლებიც წარმოადგენენ ნერვული სისტემის სიცოცხლო-სათვის საჭირო ფუნქციას და მისი გამოთვლითი უნარის ცენტრს.

სომას დანიშნულებაა შემოსული იმპულსების დაგროვება (აჯამვა). სომას პოლარიზაციის ხარისხი განაპირობებს მის დამუხტვა-განმუხტვას. თუ პოლარიზაციის პროცესი აჭარბებს დეპოლარიზაციის პროცესს, მაშინ აკუმულირების გარკვეული ხარისხის მიღწევისას სომა განიმუხტება და აქსონში ეს გამოიწვევს შესაბამისი სიგნალის გენერირებას. განმუხტულ ნეირონს დროის გარკვეულ მონაკვეთში აღარ შეუძლია მოახდინოს რეაგირება შემოსულ სიგნალებზე. დროის ამ მონაკვეთს ლატენცური პერიოდი ეწოდება. ამ პერიოდის გასვლის შემდეგ ნეირონი ისევ ახერხებს შემოსულ სიგნალებზე რეაგირების მოხდენას.

დენდრიტები გარემოდან მიღებულ სიგნალებს ატარებენ და გარდაქმნიან. გარადქმნისას ისინი მიღებულ სიგნალს ამუხრუჭებენ (ამცირებენ) ან ზრდიან. დენდრიტების გავლით ნეირონებიდან გამომავალი სიგნალები მოხვდებიან სომაში. დენდრიტებს აქვთ განშტოებადი სტრუქტურა, რომლებიც გამოდიან სომიდან და მათზე განლაგებულია სინაფსური შენაერთები, რომლებიც ღებულობენ სხვა აქსონებიდან სიგნალებს. დენდრიტების დიამეტრის ცვლილებაზე და მის სინაფსურ კავშირებზე არის დამოკიდებული სიგნალების გატარების პროცესი. კერძოდ, დიამეტრის შემცირება შეძლება იწვევდეს სიგნალის გაზრდას, ხოლო დიამეტრის გადიდება - შემცირებას ანუ დამუხრუჭებას.

აქსონი წარმოადგენს განუტოტებელ ნერვულ ბოჭკოს, რომლის დანიშნულებაა ნეირონის მიერ გამომუშავებული სიგნალის გატარება. აქსონი განშტოებების საშუალებით უკავშირდება სხვა აქსონებს, დენტრიტებს და მათი საშუალებით სხვა ნეირონებს. აქსონში სიგნალი არსებობს იმ შემთხვევაში, როდესაც სომა განიმუხტება. იმპულსების ამპლიტუდა განმუხტვის ძალის მიუხედავად მუდმივია, მაგრამ იცვლება მათი სიხშირე. დენდრიტებში მისი დიამეტრის ცვლილებებით და სინაფსების საშუალებით ხდება თანაკვეთის, არჩევის და აკრძალვის ოპერაციები, ანუ ლოგიკური „და“, „ან“ და „არა“ მოქმედებები. აქსონის დაბოლოებას გააჩნია უამრავი განშტოება,

რომელთა დაბოლოება წარმოადგენენ სინაფსებს, საიდანაც სიგნალი დენდრიტების საშუალებით გადაეცემა სხვა ნეირონებს, ხოლო ზოგიერთ შემთხვევაში სომას.

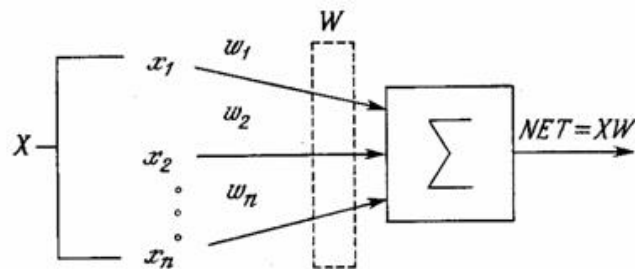
სინაფსი ეწოდება დაბოლოებებს რომლებიც გააჩნია აქსონს. აქედან ხდება სიგნალის გადაცემა დენდრიტების საშუალებით ნეირონებზე. სინაფს აქვს სფეროსებრი ფორმა ის მდიდარია ნეიროტრანსმიტერული მოლეკულებით. ეს ნეიროტრანსმიტერები მოთვასებულია ბუშტულებში. როდესაც იმპულსი გადის აქსონში ზოგიერთი მისი ბუშტულები გამოანთავისუფლებენ თავის შემადგენლობას სინაფსურ ხვრელში, რომლებსაც იჭერენ დენდრიტის სპეციალური რეცეპტორები და ხდება მათი დანერგვა უჯრედის სხეულში. არსებობს 30-ზე მეტი სახის ნეიროტრანსმიტერები, რომლებთან ზოგი წარმოადგენს აღმგზნებს და იწვევენ უჯრედის აღგზნებას და ზოგიერთი გამოიმუშავებს გამომავალ იმპულსებს. სხვები წარმოადგენენ დამუხრუჭების ნეიროტრანსმიტერებს, რომლებიც ცდილობენ იმპულსის შემცირებას ან ჩაქრობას.

სომა შემუსულ სიგნალებს აჯამავს და თუ აჯამული სიგნალი ზღურბლურ სიგნალზე მეტია, გამოიმუშავებს იმპულსს, რომელიც აქსონის გავლით გადაეცემა სხვა ნეირონებს.

ხელოვნური ნეირონი

ხელოვნური ნეირონის შექმნის ისტორია მოდის 1943 წლიდან, როცა შოტლანდიელმა მაკალოკმა და ინგლისელმა პიტსმა ფორმალური ნეიროქსელების თეორია შეიმუშავეს, ხოლო თხუთმეტი წლის შემდეგ როზენბლატმა შექმნა ხელოვნური ნეირონი (პერსეპტრონი), რომელმაც შემდგომ ნეიროკომპიუტერებს ჩაუყარა საფუძველი.

ხელოვნური ნეირონი არის ბუნებრივი ნეირონის მსგავსი მოწყობილობა, რომელსაც აქვს რამდენიმე შესასვლელი და ერთი გამოსასვლელი (ნახ.15.4). შესასვლელზე მოდებული სიგნალები ნეირონის მიერ გადამუშავდება და გამოსავლელზე აღიმკრება გამომავალი სიგნალი. ხელოვნური ნეირონი წარმოადგენს ბუნებრივი ნეირონის იმიტაციას, რომლის შესასვლელზე შედის გარკვეული რაოდენობის სიგნალების ერთობლიობა, რომლებთან თითოეული მათგანი წარმოადგენს სხვა ნეირონის გამოსასვლელს. ეს შესასვლელი სიგნალები შეესაბამებიან ბიოლოგიური ნეირონის სინაფსებში გამავალ სიგნალებს.



ნახ.15.4

ხელოვნურ ნეირონს შესასვლელზე მიეწოდება მრავალი სიგნალი (იმპულსი) და თითოეული წარმოადგენს სხვა ნეირონის გამოსასვლელს. თითოეული შესასვლელი მრავლდება სინაფსური სიმძლავრის შესაბამის წონაზე და ხდება ყველა ელემენტის შეჯამება, ნეირონის აქტივაციის დონის განსაზღვრის მიზნით. ნახ.15.4-ზე წარმოდგენილია ამ იდეის რეალური მოდელი. მიუხედავად იმისა, რომ ქსელური პარადიგმები ძალიან მრავალფეროვანია, თითქმის ყველა მათგანი ეფუძნება ამ კონფიგურაციას. აქ შესასვლელი სიგნალების სიმრავლე აღნიშნულია $X_1, X_2, X_3 \dots X_n$ -ით რომლებიც მიეწოდება ხელოვნურ ნეირონის შესასვლელზე. ეს შესასვლელი სიგნალები, რომლებიც ერთობლიობაში აღინიშნებიან X ვექტორით, შეესაბამებიან სიგნალებს, რომლებიც შემოდის ბიოლოგიური ნეირონის სინაფსებში. თითოეული შესასვლელი მრავლდება შესაბამის $W_1, W_2, W_3 \dots W_n$ წონაზე, რომელიც სინაფსური ძალის ანალოგიურია და ყველა ნამრავლი იჯამება, რითაც განისაზღვრება ნეირონის აქტივაციის დონე და მიეწოდება ამჯამავ ბლოკს, რომელიც აღინიშნება SUM (ადაპტური ამჯამავი). თითოეული წონა შეესაბამება ერთი სინაფსური კავშირის „ძალას“ (წონების სიმრავლე ერთობლიობაში აღინიშნება ვექტორით W). ამჯამავი ბლოკი, რომელიც შეესაბამება ბიოლოგიური ელემენტის სხეულს, აჯამებს აწონილ შესასვლელებს ალგებრულად და მიიღება NET გამოსასვლელი, რომელიც ვექტორულ აღნიშვნებში ის შეიძლება კომპაქტურად ჩაიწეროს ასეთი სახით

$$NET = XW .$$

ამრიგად ხელოვნური ნეირონი არა უბრალოდ ახდენს შესასვლელი სიგნალების შეჯამებას, არამედ გამოითვლის შესასვლელი სიგნალების ვექტორისა და წონითი კოეფიციენტების ვექტორის სკალარულ ნამრავლს.

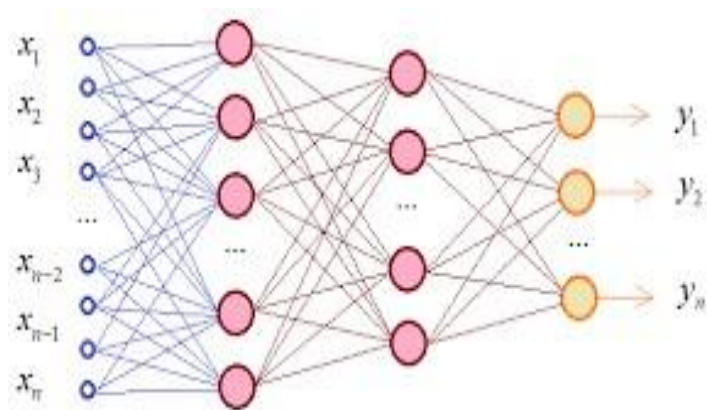
ნეირონული ქსელი

ადამიანის თავის ტვინი შედგება ნეირონებისაგან, რომელიც საერთო ჯამში ქმნის ნერვულ სისტემას, ნეირონების მთავარი ამოცანაა ინფორმაციის გავრცელება სხეულში ელექტრო და ქიმიური სიგნალების გამოყენებით, ისინი ამ ინფორმაციას იღებენ გარემოდან ან თავად ორგანიზმიდან, აფასებენ მას და ფიქრობენ თუ როგორ უნდა მოახდინონ მასზე რეაგირება და დაიმახსოვრონ ეს პროცესი.

ხელოვნური ნეირონული ქსელი არის ბიოლოგიური ქსელის მოდელი, რომელიც წარმოადგენს ეთმანეთზე მიერთებული ნეირონების ერთობლობას და გამოიყენება მის შესასვლელებსა და გამოსასვლელებს შორის რთული ურთიერთდამოკიდებულების დასამყარებლად. გამოსასვლელ სიგნალებს - ამოცნობილ სიგნალებს ან ქსელის გადაწყვეტილებას უწოდებენ. ნეირონული ქსელის მიზანი სწორი გადაწყვეტილების მიღებაა. სწორი გადაწყვეტილების მისაღებად საჭიროა ქსელის ტოპოლოგია და თითოეული ნეირონის პარამეტრები ოპტიმალურად შეირჩეს. ქსელის ტოპოლოგიის და ნეირონების პარამეტრების შერჩევის პროცესს ნეირონული ქსელის სწავლებას უწოდებენ. მრავალი ტიპის ნეირონული ქსელებიდან ერთ-ერთ ყველაზე გავრცელებულ და ეფექტურ ნეირონულ ქსელს წარმოადგენს - სრულად ბმული ქსელი (ნახ.15.5). ამ ქსელში ნეირონები რამდენიმე შრეზეა განლაგებული.

პირველი შრე ნეირონების შესასვლელი სიგნალების $X_1, X_2, \dots, X_n$ განაწილებისათვის გამოიყენება. ამ შრეში ნეირონები არ გვაქვს. ყველა შესასვლელი სიგნალი მიეწოდება მეორე შრეზე განლაგებულ ყველა ნეირონს.

მეორე და ყველა შემდგომ შრეს, ბოლოს გარდა, ფარულ შრეს უწოდებენ. ფარულ შრეებში არის განლაგებული ნეირონები. ფარული შრის ყველა ნეირონის გამოსასვლელი



ნახ.15.5

უერთდება შემდეგი შრის ყველა ნეირონს. ბოლო შრეზე განლაგებული ნეირონების გამოსასვლელი წარმოადგენს ნეირონული ქსელის გამოსასვლელებს $Y_1, Y_2, \dots, Y_n$.

სრულად ბმული ნეირონული ქსელის დანიშნულებაა შესასვლელი სიგნალების მიხედვით, წინასწარ მოცემული გამოსავლელი სიგნალების გენერაცია.. ფაქტობრივად მოცემული გვაქვს $X_1, X_2, X_3, \dots, X_n$ შესასვლელი და $Z_1, Z_2, Z_3, \dots, Z_n$ სასურველი გამოსავლელი სიგნალების n ელემენტია სიმრავლე. თითოეულ წყვილს სასწავლო ნიმუშს უწოდებენ, ხოლო მათ სიმრავლეს სასწავლო ნიმუშების სიმრავლეს. ქსელი იდეალურად იქნება შედგენილი, თუ მასზე ყოველი სასწავლო ნიმუშის $X_1, X_2, X_3, \dots, X_n$ შესასვლელი სიგნალების მოდებისას, გამოსავლელზე მიღებული $Y_1, Y_2, Y_3, \dots, Y_n$ გამოსასვლელი სიგნალები ამავე ნიმუშის $Z_1, Z_2, Z_3, \dots, Z_n$ სასურველ გამოსასვლელ სიგნალებს დაემთხვევა.

ნეირონული ქსელი წარმოადგენს მათემატიკური მოდელს, ის შედგება აპარატული და პროგრამული უზრუნველყოფისაგან, რომლებიც აგებულია ბიოლოგიური ნერვული ქსელების-ნერვული უჯრედების ფუნქციონირების პრინციპზე, ეს კონცეფცია გაჩნდა მას შემდეგ რაც მეცნიერებმა აქტიურად დაიწყეს თავის ტვინის შესწავლა და დაიწყო კვლევები იმის შესახებ თუ როგორ ფუნქციონირებდა ადამიანის ტვინი და რა პროცესები მიმდინარეობდა მასში.

ნეირონული ქსელის ძირითად უპირატესობას, რა თქმა უნდა წარმოადგენს ინფორმაციის პარალელური დამუშავება და მისი დასწავლის უნარი. ნეირონულ ქსელებს შეუძლიათ ამოხსნან ისეთი ტიპის ამოცანები, სადაც რაიმე კანონზომიერების დანახვა შეუძლებელი და უცნობია თუ რა დამოკიდებულება აქვთ ქსელში შემავალ სიდიდეებს ერთმანეთთან. ნეირონულ ქსელებში მრავალშრიანი პერსპექტონების გამოყენებამ მნიშვნელოვნად გაამარტივა სხვადასხვა სირთულის ამოცანების ამოხსნა. ამასთან ერთად დაიხვეწა ალგორითმებიც. ერთ ერთ ასეთ ალგორითმს წარმოადგენს ცდომილების უკუგავრცელების ალგორითმი რომელმაც საკმაოდ გააფართოვა გადასაწყვეტი ამოცანების ქსელი და შესაძლებელი გახდა სხვადასხვა ტიპის სირთულის პრობლემატური ხასიათის ამოცანების გადაწყვეტა.

ნეირონული ქსელის ერთ - ერთ უპირატესობას წარმოადგენს მისი მდგრადობის უნარი, ნეირონული ქსელის შიგნით იქმნება ეგრეთწოდებული ხმაური და მონაცემთა ისეთი ნაკადები რომლებიც არაინფორმაციულია. ამ პრობლემის თავიდან ასაცილებლად ნეირონული ქსელი ამ მონაცემების სიმრავლიდან ამოარჩევს ყველაზე ნაკლებად ინფორმაციულ მონაცემებს და იგნორირებას უკეთებს მას. ამის წყალობით არც ქსელში მოხდება შეფერხება და მაქსიმალურად გამოსასვლელზეც არ გვექნება არაზუსტი მნიშვნელობები.

ნეირონულ ქსელებს შეუძლიათ მოახდინონ ადაპტაცია გარემოს ცვლილებასთან ერთად. თუ ნეირონული ქსელს ერთ გარემოში აქვს ნასწავლი და ის მოხვდება სხვა გარემოში, მას შეუძლია უკვე შეძენილი ცოდნა გამოიყენოს ახალ გარემოში ადაპტაციის შემდეგ. თუმცა ასეთი ადაპტაციის უნარი ყოველთვის არ იძლევა სასურველ შედეგს, რომელშიც იგულისხმება სისტემის მდგრადობის დაღრვევა, რაც გამოწვეულია სისტემის ძირითადი პარამეტრების არასტაბილურობით.

ნეირონული სისტემის მთავარ უპირატესობად ტრადიციულ გამოთვლით სისტემებთან შედარებით თამამად შეიძლება დასხელდეს მის მიერ ამოცანის ამოხსნის (გადაწყვეტის) მაღალი სიჩქარე, ასევე ის რომ მას შეუძლია ისეთი ამოცანების გადაწყვეტაც, რომლებსაც ტრადიციული გამოთვლითი სისტემები საერთოდ ვერ წყვეტენ. ნეირონულ სისტემებში მონაცემების დამუშავება მიმდინარეობს პარალელურ რეჟიმში,

რაც საშუალებას იძლევა ძალიან დიდი რაოდენობის მასივები დამუშავდეს რაც შეიძლება მოკლე დროში.

ნეირონული ქსელი აპარატულად ისეა რეალიზებული, რომ მაქსიმალურადაა გათვალისწინებული ინფორმაციის დაკარგვის ან სხვა სახის ხარვეზების თავიდან აცილება. თუ ადგილი ექნება ნეირონულ ქსელში რომელიმე ნეირონის დაზიანებას ან რაიმე კავშირის დაკარგვას, ასეთ შემთხვევაში დაიკარგება ის ინფორმაცია, რასაც ინახავდა ესა თუ ის ნეირონი ან კავშირი. ამ პრობლემის თავიდან აცილებისათვის გამოიყენება ე.წ. ქსელის ინფორმაციის შენახვის განაწილების მეთოდი, რომლის გამოყენება საშუალებას იძლევა თუ არ მოხდება ქსელის სტრუქტურის სერიოზული დაზიანება, გარკვეული ნეირონების დაზიანება ვერ მოახდენს გავლენას მთლიანი ქსელის გამართულ ფუნქციონირებაზე.

ნეირონული ქსელები ფართოდ გამოიყენება საბორტო სისტემებში. ამოცანის გადაწყვეტა სუსტადაა დაკავშირებული ცალკეული ნეირონების გაუმართაობისგან. ეს იძლევა მის უნიკალურობას - გამოყენებულ იქნეს საბორტო ინტელექტუალურ სისტემებში.

ნეიროკომპიუტერი ზოგადად რომ განვმარტოთ არის კომპიუტერული სისტემა (სურ.15.6), რომელიც წარმოდგენილია აპარატული და პროგრამული უზრუნველყოფის ერთობლიობით, რომელიც შეესაბამება ნეირონული ქსელის მუშაობის პრინციპებს.



სურათი 15.6

ჩვეულებრივ ციფრულ კომპიუტერებთან შედარებით ნეიროკომპიუტერები ფლობენ მრავალ უპირატესობას: გააჩნიათ ამოცანის შესრულების სწრაფი უნარი, რაც მიღწეულია მისი ალგორითმების მაღალი პარალელიზმით; ნეიროსისტემები ხასიათ-დებიან მაღალი მდგრადობით გარეშე ზემოქმედებისა და ჩარევის მიმართ; სტაბილური ნეიროსისტემების შექმნა შესაძლებელია არასაიმედო ელემენტებისაგან პარამეტრების მნიშვნელოვანი გაფანტვით; მათი გამოყენება შესაძლებელია საკმაოდ რთული და სპეციფიკური ამოცანების გადასაწყვეტად.

დღეისათვის ნეიროკომპიუტერებს აქტიურად და წარმატებით იყენებენ სხვადასხვა საავიაციო ობიექტების სამართავად.

ნეირონული ქსელები წარმოიშვნენ ხელოვნური ინტელექტის სფეროში კვლევების შედეგად, სახელდობრ, მცდელობიდან აღწარმოებული ყოფილიყო ბიოლოგიური ნერვული სისტემის სწავლებისა და შეცდომების გასწორების უნარი. ასეთი სისტემები დაფუძნებული იყო აზროვნების პროცესის მაღალდონიან მოდელირებაზე ჩვეულებრივი კომპიუტერის გამოყენებით. მალე გახდა ცნობილი, რომ ხელოვნური ინტელექტის შესაქმნელად, აუცილებელია აიგოს ისეთი სისტემა, რომელიც მსგავსი იქნება ბუნებრივი არქიტექტურისა, რაც იმის მიმანიშნებელია, რომ უნდა მოხდეს აზროვნების პროცესის პროგრამული რეალიზაციიდან აპარატულზე გადასვლა.

აპარატული და პროგრამული მიდგომის ბუნებრივ გაგრძელებას ნეიროკომპიუტერის რეალიზაციისათვის წარმოადგენს პროგრამულ-აპარატული მიდგომა. აპარატული მიდგომა დაკავშირებულია ელექტრონულ-ანალოგური, ოპტოელექტრონული და ოპტიკური ტიპის ნეიროკომპიუტერების შექმნასთან ნეირომსგავსი სტრუქტურების სახით. ასეთი კომპიუტერებისათვის მუშავდება სპეციალური ზედიდი ინტეგრალური მიკროსქემები. ნეიროქსელების საფუძველს წარმოადგენს შედარებით მარტივი, უმრავლეს შემთხვევაში ერთნაირი ტიპის ელემენტები, რომლებიც იმიტირებას უკეთებენ ტვინის ნეირონებს ანუ წარმოადგენენ ხელოვნურ ნეირონებს. ნეირონი აღჭურვილია სინაფსების ჯგუფით- ერთმიმართულებიანი შესასვლელი კავშირებით, რომლებიც შეერთებულია სხვა ნეირონების გამოსასვლელებთან, ასევე მას გააჩნია აქსონი - მოცემული ნეირონის გამოსასვლელი კავშირი, რომლისგანაც სიგნალი მიეწოდება მომდევნო ნეირონების სინაფსებს. თითოეული სინაფსი ხასიათდება სინაფსური კავშირის სიდიდით ან მისი წონით, რომელიც ფიზიკური არსით ექვივალენტურია ელექტრული გამტარებლობისა ელექტრულ კავშირებში. ცალკეული ტიპის ამოცანებისათვის არსებობს ნეირონული ქსელების ოპტიმალური კონფიგურაცია. თუ ამოცანები არ იქნებიან დაყვანილი ცნობილი ამოცანების არცერთ ტიპზე, მაშინ დამმუშავებელი იძულებულია გადაწყვიტოს ახალი კონფიგურაციის სინთეზის რთული პრობლემა. ამასთან ის ხელმძღვანელობს რამდენიმე მთავარგანმსაზღვრელი პრინციპით: ქსელის შესაძლებლობებს ზრდიან ქსელის უჯრედების რიცხვის, მათ შორის კავშირების სიმკვრივის და ნეირონების შრეების რიცხვის გაზრდით.

ნეირონული ქსელის ერთ-ერთ მნიშვნელოვან განსაკუთრებულობას წარმოადგენს სწავლების შესაძლებლობა. ნეიროქსელის სწავლება შეიძლება წარიმართოს მასწავლებლით ან მის გარეშე. სწავლების შემდეგ მაგალითების დიდი რაოდენობის დაგროვების შედეგად შესაძლებელია გამოყენებულ იქნეს ნასწავლი ქსელი პროგნოზირებისათვის, იმ შემთხვევაში თუ ის აღიჭურვება ახალი შესასვლელი მნიშვნელობებით. ეს არის ნეიროკომპიუტერის უმნიშვნელოვანესი ღირსება, რომელიც საშუალებას აძლევს მას გადაწყვიტოს ინტელექტუალური ამოცანები, დაგროვილი გამოცდილების საფუძველზე.

ნეირონული სისტემის მთავარ უპირატესობად ტრადიციულ გამოთვლით სისტემებთან შედარებით შეიძლება ჩაითვალოს გამოთვლების შესრულების მაღალი სიჩქარე და ამოცანების გადაწყვეტაზე დახარჯული ნაკლები დრო. რადგან ნეირონულ სისტემებში მონაცემების დამუშავება მიმდინარეობს პარალელურ რეჟიმში, ეს საშუალებას იძლევა ძალიან დიდი მასივები დამუშავდეს რაც შეიძლება მოკლე დროში. ნეირონული ქსელების გამოყენების სპექტრი დღითიდღე უფრო და უფრო იზრდება და მათი დახმარებით შესაძლებელი ხდება უამრავი ისეთი პროცესის ავტომატიზაცია რომელიც მნიშვნელოვნად ამცირებს ადამიანური რესურსის საჭიროებას და დაშვებული შეცდომების ალბათობას. ნეირონული ქსელების დანერგვა დღესდღეისობით წარმატებით მიმდინარეობს ავიაციის სფეროში იქნება ეს სამხედრო ავიაცია თუ სამოქალაქო, ასევე აქტიურად მიმდინარეობს ამ ტენდენციის დანერგვა და საკმაოდ კარგადაც ხდება მათი ათვისება აეროპორტებში.

ნეირონული ქსელები შესაძლებელია შესრულებულ იყოს ორი გზით: პირველი - ეს არის ნეირონული ქსელის პროგრამული მოდელი და მეორე - აპარატული. ძირითად კომერციულ აპარატულ ნაკეთობებს ნეირონული ქსელის საფუძველზე წარმოადგენს ნეირო დიდი ინტეგრალური სქემები, რომლებიც სავარაუდოდ პრიორიტეტული დრჩებიან მომავალშიც.

ამჟამად დამუშავებადი ნეირო დიდი ინტეგრალურ სქემებს შორის გამოიყოფიან Adaptive Solutions (აშშ) და Hitachi (იაპონია) ფირმების მოდელები. ფირმა Adaptive Solutions-ის დამუშავებული ნეირო დიდი ინტეგრალური სქემა, სავარაუდოდ გახდება ერთ-ერთი ძლიერ სწრაფქმედი - დამუშავების გამოცხადებული სიჩქარე შეადგენს 1,2 მილიარდ შეერთებას წამში. ნეიროქსელი შეიცავს 64 ნეირონს 262 144 სინაფსს. Hitachi ფირმის ნეირო დიდი ინტეგრალური სქემა საშუალებას იძლევა რელიზებულ იქნეს ნეირონული ქსელი, რომელიც შეიცავს 576 - მდე ნეირონს. ეს ნეირო დიდი ინტეგრალური სქემები, უდაოდ გახდებიან საფუძველი ახალი ნეიროკომპიუტერების და სპეციალიზებული მრავალპროცესორული ნაკეთობების შექმნისა.

დღევანდელი ნეიროკომპიუტერებიდან უმრავლესობა წარმოადგენს პერსონალურ კომპიუტერს ან სამუშაო სადგურს, რომელთა შემადგენლობაში შედის დამატებითი ნეიროფირფიტა. მათ რიცხვს მიეკუთვნება მაგალითად Fujitsu ფირმის FMR სერიის კომპიუტერები. ასეთ სისტემებს გააჩნიათ უპირობო უფლება არსებობაზე,

იმდენად რამდენადაც მათი შესაძლებლობები სრულებით საკმარისია ახალი ალგორითმების დამუშავებისათვის და დიდი რაოდენობის გამოყენებითი ამოცანების გადაწყვეტისთვის ნეირომათემატიკის მეთოდების საშუალებით.

თუმცა უდიდეს ინტერესს იწვევენ სპეციალიზებული ნეიროკომპიუტერები, რომლებიც უშუალოდ რელიზებას უკეთებენ ნეირონული ქსელის პრინციპებს.

ასეთი სისტემების ტიპური წარმომადგენლები არიან TRV ფირმის Mark ოჯახის კომპიუტერები (პერსეპტრონის პირველი რეალიზაცია, დამუშავებული როზენბლატის მიერ, იწოდებოდა Mark I - ად).

ფირმა TVR- ის მოდელი Mark III წარმოადგენს მუშა სადგურს, რომელიც შეიცავს Mototola 68 000 ოჯახის 15 პროცესორამდე ერთეულს მათემატიკურ სოპროცესორთან ერთად. ყველა პროცესორი გაერთიანებულია VME სალტით. სისტემის არქიტექტურა, რომელიც მოიცავს პროცესორულ ელემენტებს 1 მილიონი აწყობადი შეერთებებით, საშუალებას იძლევა დაამუშაოს 450 ათასი ერთმანეთშორისი შეერთებები წამში.

Mark IV - ეს არის ერთპროცესორიანი სუპერკომპიუტერი კონვეირული არქიტექტურით. ის მოიცავს 236 ათასამდე ვირტუალურ პროცესორულ ელემენტებს, რაც იძლევა საშუალებას დამუშავდეს მილიონამდე ერთმანეთშორისი შეერთებები წამში.

Mark ოჯახის კომპიუტერებს აქვთ საერთო პროგრამული გარსი ANSE (Artificial Neural Zystem Eenvironment), რომელიც უზრუნველყოფს მოდელების პროგრამულ შეთავსებადობას. გარდა მითითებული მოდელებისა ფირმა TRV გვთავაზობს ასევე Mark II პაკეტს, რომელიც ნეირონული ქსელის ემულატორს წარმოადგენს.

სხვა საინტერსო მოდელს წარმოადგენს ნეიროკომპიუტერი METSIM, რომელიც შემნილია Texas Instruments ფირმის მიერ კემბრიჯის უნივერსიტეტის ბაზაზე. მისი ტოპოლოგია წარმოადგენს სამგანზომილებიან გისოსს - სტანდარტულ გამოთვლით კვანძებს 80188 პროცესორის ბაზაზე. კომპიუტერი NETSIM გამოიყენება ხელოვნური ქსელების ისეთი მოდელების მოდელირებისთვის როგორიცაა ხოფილდ კოხონენის ქსელი უკუგავრცელებით. მისი მწარმოებლურობა აღწევს 450 მილიონი ერთმანეთ-შორისი შეერთებები წამში.

ფორმა Computer Recognition Systems (CRS) ყიდის WIZARD/CRC 1000 სერიის ნეიროკომპიუტერებს, რომელთა დანიშნულებაა ვიდეოგამოსახულების დამუშავება. შესასვლელი გამოსახულების ზომაა 512x512 პიკსელი. მოდელმა CRS 1000 უკვე ჰპოვა გამოყენება ავტომატური კონტროლის სამრეწველო სისტემებში.

ნეიროკომპიუტერები წარმოადგენენ თანამედროვე მაღალმწარმოებლურ გამოთვლითი ტექნიკის განვითარების პერსპექტიულ მიმართულებას და ცნობილი არიან როგორც ნეირომორფული კომპიუტერული არქიტექტურის ტიპი, რომელიც შთაგონებას იღებს ადამიანის ტვინის სტრუქტურიდან და ფუნქციონირებიდან. ეს კომპიუტერები შექმნილია გარკვეული ამოცანების უფრო ეფექტურად შესასრულებლად, ნერვული ქსელების და პარალელური დამუშავების პრინციპების გამოყენებით. ხოლო ნეირონული ქსელების თეორია და ნეირომათემატიკა თავის მხრივ წარმოადგენს დღევანდელი გამოთვლითი მეცნიერების პრიორიტეტულ მიმართულებას.

ავიაციის კონტექსტში, ნეიროკომპიუტერებს აქვთ ინდუსტრიის სხვადასხვა ასპექტების რევოლუციის პოტენციალი. მოვიყვანოთ რამდენიმე პოტენციური აპლიკაცია:

ფრენის კონტროლის სისტემები: ნეიროკომპიუტერებს შეუძლიათ გააძლიერონ ფრენის მართვის სისტემების შესაძლებლობები უფრო სწრაფი და ინტელექტუალური გადაწყვეტილების მიღების გზით. ამ სისტემებს შეუძლიათ რეალურ დროში გააანალიზონ სენსორების დიდი რაოდენობით მონაცემები, როგორიცაა რადარი და სანავიგაციო ინსტრუმენტები, რაც ფრენის პირობების შეცვლაზე უფრო სწრაფი და ზუსტი რეაგირების საშუალებას იძლევა.

პილოტების სწავლება და სიმულაცია: ნეიროკომპიუტერები შეიძლება გამოყენებულ იქნას პილოტების წვრთნისა და სიმულაციურ სისტემებში, რათა შეიქმნას უფრო რეალისტური გარემო. რეალური სამყაროს სცენარების მაღალი ხარისხის სიმულაციის გზით, ამ სისტემებს შეუძლიათ პილოტებს მიაწოდონ ტრენინგის ღირებული გამოცდილება, რაც აძლიერებს მათ ოპტიმალური გადაწყვეტილების გამომუშავების უნარს რეალური ფრენების რთულ სიტუაციებში მოხვედრის შემთხვევაში.

საჰაერო მოძრაობის მენეჯმენტი: ნეიროკომპიუტერებს შეუძლიათ დაეხმარონ საჰაერო მოძრაობის მართვის სისტემების ოპტიმიზაციას რეალურ დროში ფრენის დიდი რაოდენობით მონაცემების ეფექტურად დამუშავებით და ანალიზით. ამან შეიძლება გამოიწვიოს მარშრუტების დაგეგმვის გაუმჯობესება, საჰაერო სივრცის მართვა და გადატვირთულობის თავიდან აცილება, რაც საბოლოოდ გამოიწვევს ოპერაციული ეფექტურობის გაზრდას და შეფერხებების შემცირებას.

ტექნიკური მომსახურება და დიაგნოსტიკა: ნეიროკომპიუტერები შეიძლება გამოყენებულ იქნეს თვითმფრინავების წინასწარი ტექნიკური მომსახურებისა და დიაგნოსტიკისათვის. სხვადასხვა კომპონენტებისა და სისტემების სენსორის მონაცემების ანალიზით, ამ კომპიუტერებს შეუძლიათ პოტენციური პრობლემების იდენტიფიცირება და ტექნიკური მოთხოვნილებების პროგნოზირება, რაც ხელს შეუწყობს წარუმატებლობის თავიდან აცილებას და მოცდენის დროის შემცირებას.

მნიშვნელოვანია აღინიშნოს, რომ მიუხედავად იმისა, რომ ნეიროკომპიუტერები დიდ პოტენციურ დაპირებას იძლევიან, მათი პრაქტიკული განხორციელება ავიაციაში ჯერ კიდევ ადრეულ ეტაპზეა. კვლავაც მნიშვნელოვანი კვლევისა და განვითარების ძალისხმევის გატარებაა საჭირო ტექნიკური გამოწვევების დასაძლევად და მათი საიმედოობის ასამაღლებლად და არსებულ საავიაციო სისტემებთან თავსებადობის უზრუნველსაყოფად.

15.3. კვანტური კომპიუტერები

კვანტური გამოთვლების ისტორია დაიწყო 1980 - იანი წლების დასაწყისში, როცა ფიზიკოსმა პოლ ბენოფმა გააკეთა შემოთავაზება 1980 წელს ტიურინგის მანქანის კვანტურ-მექანიკური მოდელის შესახებ.

კვანტური გამოთვლების იდეა ასევე გამოთქმული იქნა იური მანინის მიერ 1980 წელს.

კვანტური კომპიუტერების ერთ-ერთი პირველი მოდელები შემოთავაზებულ იქნა რიჩარდ ფეინმანის მიერ 1981 წელს. მალე პოლ ბენოფმა აღწერა ასეთი კომპიუტერის აგების თეორიული საფუძვლები.

კვანტური კომპიუტერის კონცეფცია ასევე შემოთავაზებულ იქნა 1983 წელს სტივენ ვიზნერის სტატიაში.

კვანტური კომპიუტერის აუცილებლობა წარმოიშვა მაშინ, როცა გადაწყდა ფიზიკის მეთოდებით გამოკვლეულიყო ბიოლოგიურის მსგავსი რთული მრავალნაწილაკიანი სისტემები. ასეთი სისტემების კვანტურ მდგომარეობათა სივრცე იზრდება როგორც ექსპონენტა მათი შემადგენელი რეალური ნაწილაკების n რიცხვისგან დამოკიდებულებით, რაც შეუძლებელს ხდის მოხდეს მათი ქცევის მოდელირება კლასიკური კომპიუტერების საშუალებით უკვე მაშინაც კი, როცა $n = 10$. ამიტომ ვიზნერი და ფეინმანი იყვნენ პირველი, რომლებმაც გამოთქვეს კვანტური კომპიუტერის აგების იდეა.

მომავალში კომპიუტერულ სფეროში სილიციუმის მიკროელექტრონიკის ერა დასრულდება და ელექტრონულ კომპიუტერებს ჩანაცვლებს კვანტური კომპიუტერები, რომლებიც კვანტური ალგორითმების შესრულების გზით მუშაობისას გამოიყენებს კვანტურ-მექანიკურ ეფექტებს. ისინი თანამედროვე კომპიუტერებთან შედარებით გაცილებით სწრაფი, კომპაქტური და ეკონომიური იქნება. კვანტური გამოთვლები ჯერ კიდევ ჩანასახის სტადიაშია, თუმცა მიმდინარეობს პერმანენტული პრაქტიკული და თეორიული კვლევა კვანტური კომპიუტერების დამუშავებაში მათი გამოყენებისათვის. თუ მოხდება მსხვილმასშტაბიანი კვანტური კომპიუტერების აგება, მაშინ ისინი შეძლებენ გარკვეული პრობლემების გადაჭრას გაცილებით სწრაფად (მაგალითად, შორის ალგორითმი) ვიდრე დღეს არსებული ტრადიციული კლასიკური კომპიუტერები.

ტრადიციულ კომპიუტერს შეუძლია ერთი მოქმედების შესრულება ერთ ჯერზე, ანუ კითხულობს ინფორმაციის რაღაც ნაწილს (ერთი „ბიტი“) და მასზე ასრულებს რამდენიმე ძირეულ ლოგიკურ ოპერაციას. ეს შედარებით მარტივი პროცესები უდევს საფუძვლად ყველაფერს, რისი კეთებაც შეუძლიათ თანამედროვე კომპიუტერებს. ისინი აკეთებენ ამ მარტივ ოპერაციებს თანმიმდევრულად ინფორმაციის ნაწილზე გასაოცრად სწრაფად, ისე, რომ შესაძლებელია საკმაოდ რთული შედეგების მიღწევა (მაგალითად, როგორიცაა სამგანზომილებიანი გრაფიკა) თითქმის რეალურ დროში წარმოუდგენლად დიდი რიცხვისა ამ მარტივი ბიჯების შესრულებით. ასეთი კომპიუტერის სიმძლავრის გაზრდა ნიშნავს ამ მარტივი ბიჯების სულ უფრო და უფრო სწრაფად შესრულებას, თუმცა სამწუხაროდ არსებობს ასეთი კომპიუტერებისთვის სიჩქარის ზრდის ფუნდამენტური ზღვარი და თანამედროვე ტექნოლოგია სწრაფად უახლოვდება ამ ზღვარს. სწორედ ამან განაპირობა ისეთი კომპიუტერების შექმნის საჭიროება, რომლებშიც ეს პრობლემა გადაჭრილი იქნებოდა.

ასეთი კომპიუტერები არიან სწორედ კვანტური კომპიუტერები, რომლებიც ტრადიციული კომპიუტერებისგან იმით განსხვავდებიან, რომ ჩიპების ნაცვლად იყენებენ ელექტრონებსა და ატომებს ინფორმაციის შესანახად (კუბიტი). მათ შეუძლიათ ერთდროულად მოიხელთონ ინფორმაციის მილიარდობით ნაწილი და ამით გვერდი აუარონ ტრადიციული კომპიუტერების ნაკლს, რაც გამოწვეულია მათ მიერ ინფორმაციის მხოლოდ თითო ნაწილის თანმიმდევრულად დამუშავების შესაძლებლობით.

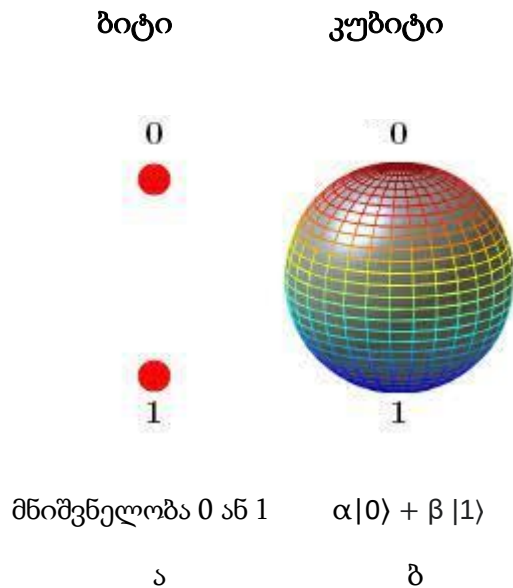
კვანტური ინფორმაციის თეორიაში კუბიტი (კვანტური ბიტი) წარმოადგენს კვანტური ინფორმაციის ერთეულს, რომელიც კლასიკური ბიტის (სურ.15.7. ა) კვანტური ანალოგია. კუბიტის სუფთა მდგომარეობების წარმოდგენა შესაძლებელია ბლოხის სფეროზე წერტილების სახით(სურ.15.7 ბ). კუბიტი ორ მდგომარეობიან კვანტურ სისტემებს აღწერს (მაგალითად, ფოტონის პოლარიზაციას ან ელექტრონის სპინს). კლასიკურ სისტემაში, ბიტი ორი მდგომარეობიდან აუცილებლად ერთ-ერთში უნდა იყოს (0-ში ან 1-ში), მაშინ როცა კუბიტს კვანტური მექანიკის მიხედვით შეუძლია ერთდროულად ორივე ამ მდგომარეობაში იყოს (ორი მდგომარეობის სუპერპოზიცია). კვანტური ინფორმაციის ეს თვისება გამოიყენება კვანტურ კომპიუტერებში. კუბიტი კლასიკურ ბიტზე მეტ ინფორმაციას იტევს. n კომპონენტიანი (ბიტის) სისტემის სრული აღწერა კლასიკურ ფიზიკაში მხოლოდ n ბიტის გამოყენებას მოითხოვს, კვანტურ ფიზიკაში კი ასეთი სისტემის აღსაწერად $2^n - 1$ კომპლექსური რიცხვია საჭირო.

ორ მდგომარეობას, რომელთა მიმართაც კუბიტი შეიძლება გაიზომოს, საბაზისო მდგომარეობებს უწოდებენ. ისე როგორც ნებისმიერი სახის კვანტური მდგომარეობის წარმოდგენა, ამ მდგომარეობების წარმოდგენაც დირაკის „ბრა-კეტ“ აღნიშვნით ხდება. შესაბამისად, პირველ მათგანს ავლნიშნავთ როგორც $|0\rangle$ და მეორეს როგორც $|1\rangle$.

კუბიტი რომელიც არ არის არცერთ სხვა სისტემასთან გადახლართული შეიძლება იყოს საბაზო მდგომარეობების ნებისმიერ წრფივ სუპერპოზიციაში. ეს ნიშნავს რომ კუბიტის მდგომარეობა შეიძლება ჩაიწეროს როგორც $|0\rangle$ და $|1\rangle$ მდგომარეობების წრფივი კომბინაცია (15.7.ბ) :

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle,$$

სადაც α და β შესაბამის მდგომარეობაში ყოფნის ამპლიტუდებია და ზოგადად შეიძლება კომპლექსური რიცხვები იყოს.



სურ.15.7

სტანდარტულ ბაზისში კუბიტის გაზომვის შედეგის $|0\rangle$ -ის ალბათობა არის $|\alpha|^2$, ხოლო $|1\rangle$ -ის შედეგის კი $|\beta|^2$. რადგან ამპლიტუდების კვადრატები ალბათობების ტოლფასია, ამიტომ α და β უნდა აკმაყოფილებენ შემდეგ განტოლებას

$$|\alpha|^2 + |\beta|^2 = 1.$$

კვანტური ფიზიკის სამყაროში პროცესები წარმოდგენილია მცირე მასშტაბებით და დაბალი ტემპერატურის პირობებში მიმდინარეობს, რაც განსაზღვრული საკითხებისადმი უნიკალურ მიდგომებს მოითხოვს. კვანტურ კომპიუტერებში ალგორითმების სივრცე, და მათზე ოპერირების სირთულე და სიჩქარე ნებისმიერი ტრადიციული კომპიუტერის სიმძლავრეს აღემატება.

კვანტურ მექანიკაში სუპერპოზიცია ეწოდება ეფექტს, როდესაც მატერიალური ნაწილაკი ერთდროულად იმყოფება ბევრ სხვადასხვა მდგომარეობაში, კვანტური ჩახლართულობა კი ამ ნაწილაკებს შორის უნიკალურ კორელაციას (ურთიერთობას) განაპირობებს, რაც კვანტური გამოთვლების საფუძველია. მეცნიერებას ჯერ ბოლომდე არ აუხსნია, თუ რა არის ამ ნაწილაკების განსხვავებული თუ საპირისპირო მდგომარეობების (მუხტი, სპინი, მაგნიტური ველი და ა.შ.) საფუძველი, მაგრამ მათი ფუნქციური მახასიათებლები იმდენად კარგადაა გამოკვლეული და შესწავლილი, რომ გამოთვლების ექსპერტებს უნიკალური კვანტური ალგორითმების შემუშავების, ხოლო ინჟინრებს საშუალებას აძლევს მათზე ოპერირებადი ტექნიკის გამოყენებით მოხდეს კვანტური კომპიუტერების აგება.

კვანტური კომპიუტერი გამოთვლითი მოწყობილობაა (სურ.15.8) რომელიც იყენებს კვანტური მექანიკის მოვლენას (კვანტური სუპერპოზიცია და კვანტური ჩახლართულობა) მონაცემების გადაცემისა და დამუშავებისათვის. როგორც უკვე ავლინებთ, კვანტური კომპიუტერი განსხვავებით ჩვეულებრივი კომპიუტერისგან ოპერირებას აკეთებს არა ბიტებზე (რომლებიც დებულობენ მნიშვნელობებს 0-ს ან 1-ს) არამედ კუბიტებზე, რომელთაც ერთდროულად გააჩნია მნიშვნელობები 0 და 1. თეორიულად ეს იძლევა საშუალებას ერთდროულად დამუშავდეს ყველა შესაძლო მდგომარეობა, რითაც მიიღწევა გიგანტური უპირატესობა ჩვეულებრივ კომპიუტერებთან შედარებით მთელი რიგი ალგორითმების მიხედვით.



სურათი 15.8

სრულფასოვანი უნივერსალური კვანტური კომპიუტერი ჯერ კიდევ წარმოადგენს ჰიპოტეტურ მოწყობილობას, რომლის აგების შესაძლებლობა დაკავშირებულია კვანტური თეორიის სერიოზულ განვითარებასთან მრავალი ნაწილაკების და რთული ექსპერიმენტების სფეროში. ამ სფეროში დამუშავებების განხორციელება დაკავშირე-

ბულია თანამედროვე ფიზიკის ახალ აღმოჩენებთან და მიღწევებთან. 2010 წლიდან მოყოლებული უკვე მიმდინარეობს ერთეული კვანტური ექსპერიმენტული სისტემების (კომპიუტერების) შექმნა. ასეთი ტიპის კომპიუტერებისათვის პირველი პრაქტიკული დაპროგრამების მაღალდონიან ენას წარმოადგენს ენა Quipper.

გამოთვლისთვის კვანტური კომპიუტერი იყენებს არა ჩვეულებრივ (კლასიკურ) ალგორითმებს, არამედ კვანტური ბუნების პროცესებს, ეგრეთწოდებულ კვანტურ ალგორითმებს, რომლებიც იყენებენ კვანტომექანიკურ ეფექტებს, ისეთებს როგორიცაა კვანტური პარალელიზმი და კვანტური ჩახლართულობა.

უფრო დეტალურად შევხვით კვანტური პარალელიზმისა კვანტური ჩალართულობის არსს.

„კვანტური პარალელიზმის“ ცნების შინაარსი გამოთვლებში შესაძლებელია ავხსნათ ასე: „გამოთვლების პროცესში მონაცემები თავის მხრივ წარმოადგენენ კვანტურ ინფორმაციას, რომელიც პროცესის დამთავრების შემდეგ გარდაიქმნება კლასიკურში კვანტური რეგისტრის საბოლოო მდგომარეობის გაზომვის გზით. კვანტურ ალგორითმებში მოგება მიიღწევა იმის შედეგად, რომ ერთი კვანტური ოპერაციის გამოყენების დროს მიიღება კვანტურ მდგომარეობათა სუპერპოზიციის კოეფიციენტების დიდი რიცხვი, რომელიც ვირტუალურ ფორმაში შეიცავს კლასიკურ ინფორმაციას და გარდაიქმნება ერთდროულად.

ჩახლართულობა არის კონტრინტუიციური კვანტური ფენომენი. ის აღწერს ქცევას რომელსაც კლასიკურ სამყაროში არასდროს ვხვდებით, ანუ მდგომარეობა რომელიც კლასიკური ლოგიკით არ აიხსნება.

როგორც ცნობილია კლასიკურ პროცესორს დროის მოცემულ მომენტში შეუძლია იმყოფებოდეს $|0\rangle, |1\rangle, \dots, |N-1\rangle$ მდგომარეობათა სიმრავლიდან (დირაკის აღნიშვნა) მხოლოდ ერთ-ერთ მდგომარეობაში, მაშინ როცა კვანტური პროცესორი დროის ყოველ მოცემულ მომენტში შესაძლებელია იმყოფებოდეს ერთდროულად ყველა ამ ბაზისურ მდგომარეობებში, ამასთან, თითოეულ j მდგომარეობაში თავისი კომპლექსური ამპლიტუდით λ_j . ეს კვანტური მდგომარეობა იწოდება მონაცემების კლასიკური მდგომარეობების „კვანტურ სუპერპოზიციად“.

ბაზისურ მდგომარეობებს შესაძლებელია ჰქონდეთ უფრო რთული სახე. ასეთ შემთხვევაში კვანტური სუპერპოზიცია შესაძლებელია წარმოვსახოთ ასეთნაირად:

წარმოვიდგინოთ ატომი, რომელიც დაექვემდებარებოდა ან არ დაექვემდებარებოდა რადიოაქტიურ დაშლას დროის განსაზღვრულ შუალედში. ჩვენ უნდა ვიფიქროთ, რომ ამ ატომს გააჩნია მხოლოდ ორი შესაძლო მდგომარეობა: „დაშლა“ და „არ დაშლა“, მაგრამ კვანტურ მექანიკაში ატომს შესაძლებელია გააჩნდეს რაღაც გაერთიანებული მდგომარეობა „დაშლა – არ დაშლა“, ანუ არც ერთი და არც მეორე მდგომარეობა, არამედ რაღაც მათ შორის. სწორედ ეს მდგომარეობა იწოდება „სუპერპოზიციად“.

კვანტური სუპერპოზიცია - მდგომარეობათა სუპერპოზიციაა, რომელიც კლასიკური გაგების თვალსაზრისით შეუძლებელია რეალიზებულ იქნეს ერთდროულად. კვანტური მექანიკის კონტექსტის მიხედვით სუპერპოზიციის არსებობის პრინციპი დასაშვებია.

კვანტური გამოთვლა არის კლასიკური მმართველი კომპიუტერის მიერ კონტროლირებადი მარტივი სახის უნიტარული ოპერაციის თანმიმდევრობა (ერთ, ორ და სამ კუბიტებზე). გამოთვლის ბოლოს კვანტური პროცესორის მდგომარეობა იზომება, რაც იძლევა სწორედ გამოთვლის საძიებო შედეგს.

კიდევ ერთხელ ხაზგასმით რომ ავღნიშნოთ კუბიტი - ეს არის კვანტური მუხტი ან უმცირესი ელემენტი კვანტურ კომპიუტერში მონაცემების შენახვისათვის. კუბიტი წარმოადგენს ხატოვნად რომ ვთქვათ „ორი ბატონის მსახურს“, როცა ის ერთ შემთხვევაში გამოდის კლასიკური ბიტის როლში, ხოლო მეორე შემთხვევაში კი შიძლება არსებობდეს მისი (კლასიკური ბიტის) ორი მდგომარეობის სუპერპოზიციაში. ერთ შემთხვევაში, თითქოს სუპერპოზიცია წარმოადგენს ალბათობის განაწილებას ცალ-ცალკე ორ მნიშვნელობაზე, მეორე შემთხვევაში კი კვანტური გამოთვლისას შეუძლია გავლენა იქონიოს ორივე მნიშვნელობაზე და შეინახოს ისინი ერთდროულად.

თითოეული კუბიტი არის მიკროპროცესორი, რომელიც მუშაობს ზეგამტარობის პრინციპებით და კვანტური მექანიკის კანონებით.

ფიზიკური სისტემები, რომლებიც რეალიზებას უკეთებენ კუბიტებს, შეიძლება იყვნენ ნებისმიერი ობიექტები, რომლებსაც ექნებათ ორი კვანტური მდგომარეობა: ფოტონების პოლარიზაციული მდგომარეობები, იზოლირებული ატომების ან იონების ელექტრონული მდგომარეობები, ატომების ბირთვების სპინური მდგომარეობები და ა.შ.

ორ მდგომარეობას რომლის მიმართაც კუბიტი შეიძლება გაიზომოს, საბაზისო მდგომარეობებს უწოდებენ. ეგრეთწოდებული სუფთა კუბიტი (ანუ კუბიტი რომელიც არ

არის არცერთ სხვა სისტემასთან ჩახლართული) შეიძლება იყოს საბაზისო მდგომარეობები ნებისმიერ წრფივ სუპერპოზიციაში.

კუბიტების ჯგუფზე ერთი ოპერაცია გამოითვლება მაშინვე ყველა მის შესაძლო მნიშვნელობებზე, მაშინ როცა მისგან განსხვავებით კლასიკური ბიტების ჯგუფისგან შესაძლებელია გამოყენებულ იქნეს მხოლოდ ერთი მიმდინარე მნიშვნელობა. სწორედ ეს უზრუნველყოფს გამოთვლების უპრეცედენტო პარალელიზმს.

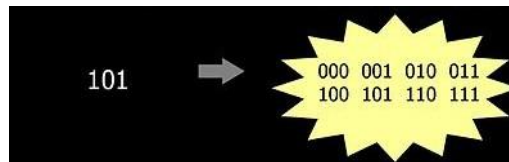
კვანტურ კომპიუტერზე გამოთვლის გამარტივებული სქემა ასე გამოიყურება: აიღება კუბიტების სისტემა რომელზეც ჩაიწერება საწყისი მდგომარეობა. შემდეგ სისტემის ან მისი ქვესისტემების მდგომარეობა იცვლება უნიტარული გარდაქმნების საშუალებით, რომლებიც ასრულებენ ამა თუ იმ ლოგიკურ ოპერაციებს. ბოლოს იზომება მნიშვნელობა და ეს წარმოადგენს კომპიუტერის მუშაობის შედეგს. კვანტურ კომპიუტერში კლასიკური კომპიუტერების გამტარების როლს თამაშობენ კუბიტები, ხოლო კლასიკური კომპიუტერის ლოგიკური ბლოკების როლს კი უნიტარული გარდამქმნელები. კვანტური პროცესორის ასეთი კონცეფცია და კვანტური ლოგიკური ვენტილები შემოთავაზებული იქნა დევიდ დოიჩის მიერ 1989 წელს. დევიდ დოიჩის მიერ ასევე 1995 წელს შემოთავაზებულ იქნა უნივერსალური ლოგიკური ბლოკი, რომლის დახმარებით შესაძლებელია ნებისმიერი კვანტური გამოთვლის შესრულება. ნიშანდობლივია, რომ ნებისმიერი კვანტური გამოთვლის აგებისათვის საკმარისია ორი საბაზო ოპერაცია. კვანტური სისტემა იძლევა შედეგს, რომელიც მხოლოდ რაღაც ალბათობით წარმოადგენს სწორს. თუმცა ალგორითმებში ოპერაციათა არც თუ ისე დიდი გაზრდით შესაძლებელია მიღწეული სწორი შედეგის მიღების ალბათობა მიუახლოვდეს ერთს.

საბაზო კვანტური ოპერაციების დახმარებით შესაძლებელია ჩვეულებრივი ლოგიკური ელემენტების (რისგანაც შედგებიან ჩვეულებრივი კომპიუტერები) მუშაობის სიმულირება. ამიტომ ნებისმიერი ამოცანას, რომელიც გადაწყვეტილია დღეს კლასიკური კომპიუტერის მიერ, მომავალში კვანტური კომპიუტერი წარმატებით გადაწყვეტს.

თანამედროვე კომპიუტერების უმრავლესობა მუშაობს ასეთი სქემით: მეხსიერების **n** ბიტი ინახავს მდგომარეობას და დროის თითოეული ტაქტი იცვლება პროცესორის მიერ. კვანტურ შემთხვევაში სისტემა შემდგარი **n** კუბიტებისგან იმყოფება მდგომარეობაში, რომელიც წარმოადგენს ყველა საბაზო მდგომარეობების

სუპერპოზიციას, ამიტომ სისტემის ცვლილება ეხება ყველა 2^n საბაზო მდგომარეობებს ერთდროულად (სურ.15.9).

სამბიტიანი კლასიკური სამკუბიტიანი კვანტური
რეგისტრი რეგისტრი



სურ.15.9

თეორიულად ახალ სქემას შეუძლია იმუშაოს მნიშვნელოვნად უფრო სწრაფად ვიდრე კლასიკურმა. მაგალითად, გროვერის ალგორითმი მონაცემთა ბაზაში ძიებისას აჩვენებს სიმძლავრის კვადრატულ ნაზრდს კლასიკურ ალგორითმებთან შედარებით.

კვანტური კომპიუტერის დამუშავება დაფუძნებულია კვანტური თეორიის განვითარებაზე მრავალი ნაწილაკებისა და რთული ექსპერიმენტების სფეროში, რომელიც ითვალისწინებს თანამედროვე ფიზიკის სფეროში ახალ-ახალ აღმოჩენებს და მიღწევებს. კვანტური კომპიუტერების აგებისას სირთულე მდგომარეობს იმაში, რომ საჭიროა შეიქმნას ისეთი პირობები, რომლებშიც კვანტური ბიტი იქნებოდა სუპერპოზიციის მდგომარეობაში უსასრულოდ დიდი ხნის განმავლობაში. კვანტური კომპიუტერის შემადგენელი მიკროსკოპული ელემენტებისათვის და ფუნქციონირებისათვის საჭიროა მთელი რიგი უნიკალური გარემო პირობების შექმნა:

- ტემპერატურა 0,02 გრადუსი კელვინი (-269,98 ცელსიუსი);
- მაგნიტური და ელექტრული გამოსხივებისგან დაცვის სისტემა (ამცირებს ამ ფაქტორების ზემოქმედებას 50 ათასჯერ);
- სითბოს არინების და ვიბრაციის ამორტიზაციის სისტემა;
- ჰაერის ატმოსფერული წნევის 100 მილიარდჯერ დაწევა.

მოთხოვნილი გარემო პირობებიდან უმნიშვნელო გადახრაც კი მყისიერად იწვევს სუპერპოზიციის მდგომარეობის დაკარგვას და კვანტური კომპიუტერის გაუმართავ მდგომარეობაში გადასვლას.

კვანტური კომპიუტერის აგება რეალური ფიზიკური ხელსაწყო სახით წარმოადგენს XXI საუკუნის ფიზიკის ფუნდამენტურ ამოცანას. 2010-იანი წლების დასაწყისისათვის აგებულია მხოლოდ მისი შეზღუდული ვარიანტები. ყველაზე უფრო დიდ კონსტრუირებულ კვანტურ რეგისტრებს გააჩნიათ რამდენიმე ათეულზე მეტი დაკავშირებული კუბიტი. საკითხი იმის შესახებ თუ როგორ ხარისხამდეა შესაძლებელი ასეთი მოწყობილობის მასშტაბირება (ე. წ. მასშტაბირების პრობლემა) წარმოადგენს ახალი ინტენსიურად განვითარებადი სფეროს — მრავალნაწილაკიანი კვანტური მექანიკის საგანს. აქ ცენტრალურ საკითხს წარმოადგენს დეკოჰერენტულობის ბუნების შესახებ საკითხი, კერძოდ ტალღური ფუნქციის კოლაფსის საკითხი, რომელიც ჯერ კიდევ ღიად რჩება.

XXI საუკუნის გარიჟრაჟზე მრავალ სამეცნიერო ლაბორატორიაში შეიქმნა ერთკუბიტური კვანტური პროცესორები (არსებითად, მართვადი ორდონიანი სისტემები, რომელთა საფუძველზე ნაწინასწარმეტყველი იყო მრავალ კუბიტებზე მასშტაბირების შესაძლებლობა).

2001 წლის ბოლოს IBM-მა გამოაცხადა 7-კუბიტიანი კვანტური კომპიუტერის წარმატებული ტესტირების შესახებ. მასზე შესრულდა შორის ალგორითმი და ნაპოვნი იქნა რიცხვის თანამამრავლები.

2005 წელს მოსკოვის ზეგამტარობის ლაბორატორიაში იაპონელი სპეციალისტების დახმარებით აიგო ორკუბიტიანი კვანტური პროცესორი ზეგამტარ ელემენტებზე.

2009 წლის ნოემბერში აშშ-ში სტანდარტების და ტექნოლოგიების ნაციონალური ინსტიტუტის ფიზიკოსებმა პირველად შეძლეს აეგოთ პროგრამირებადი კვანტური კომპიუტერი, რომელიც შედგებოდა ორი კუბიტისგან.

2012 წელს კომპანია IBM-მა გამოაცხადა ზეგამტარი კუბიტების გამოყენებით კვანტური გამოთვლების ფიზიკური რეალიზაციის თვალსაზრისით მნიშვნელოვანი პროგრესის მიღწევის შესახებ, რომელიც, კომპანიის აზრით, შესაძლებელს გახდის დაიწყოს მუშაობა რეალური კვანტური კომპიუტერის შესაქმნელად.

2017 წელს ფიზიკოსების ჯგუფმა რუსეთის კვანტურ ცენტრში შექმნა პროგრამირებადი 51 კუბიტიანი კვანტური სიმულატორი. ავტორებმა შეამოწმეს სიმულატორის მუშაობის უნარიანობა მრავალი ნაწილაკების რთული სისტემის მოდელირებით. ამან ფიზიკოსებს საშუალება მისცა ეწინასწარმეტყველათ ადრე უცნობი

ზოგიერთი ეფექტი. ამავე წელს მეცნიერთა ჯგუფმა მერილენდის უნივერსიტეტში შექმნა 53 კუბიტისანი სიმულატორი დაფუძნებული იონებზე ოპტიკურ მახეში. თუმცა ეს სისტემები არ წარმოადგენენ უნივერსალურ კომპიუტერებს და შექმნილი არიან მხოლოდ ერთი კონკრეტული ამოცანის გაგაწყვეტისათვის. ამავე წელს IBM-ის მეცნიერებმა წარმატებით შექმნეს და გამოცადეს 50 კუბიტისანი პროცესორის პროტოტიპი.

2018 წელს Intel-ის კომპანიამ გაასაჯაროვა ზეგამტარი კვანტური მიკროსქემის შექმნის შესახებ კოდური სახელწოდებით „Tangle Lake, რომელიც შეცავს 49 კუბიტს. Intel ეწევა კვანტური კომპიუტერების დამუშავებას ორი მიმართულებით: მოწყობილობების შექმნა ზეგამტარებზე და კაჟბადის მიკროქემებზე „სპინური კუბიტების“ საფუძველზე.

2018 წელს Google კომპანიამ გამოაცხადა რომ მათ შეძლეს 72 კუბიტისანი კვანტური პროცესორის Bristlecone შექმნა, რომელთაც გააჩნიათ შეცდომათა ძალიან მცირე ალბათობა გამოთვლების დროს. ამავე წელს იყო შეტყობინება ოპტიკური მიკროჩიპის დამუშავების შესახებ, რომლის გამოყენება მომავალში დაგეგმილია, როგორც კვანტური კომპიუტერის შემადგენელი ნაწილი.

2019 წელს IBM-ის კომპანიამ წარმოადგინა მსოფლიოში პირველი კომერციული კვანტური კომპიუტერი IBM Q System One.

2019 წელს კომპანია Google გამოაცხადა რომ მან შეძლო აეგო 53 კუბიტისანი ზეგამტარი კვანტური პროცესორი Sycamore და აჩვენა „კვანტური უპირატესობა“ ჩვეულებრივ კომპიუტერებთან შედარებით.

2020 წელს ჩინეთის სამეცნიერო ტექნიკური უნივერსიტეტის მკვლევარებმა გამოაქვეყნეს სტატია, სადაც ამტკიცებენ, რომ მათ კვანტურ კომპიუტერს შეუძლია კვანტური უპირატესობის მიღწევა. მან მხოლოდ რაღაც რამდენიმე წუთში შეძლო ჩაეტარებინა ოპერაცია, რომელიც ტრადიციული ხერხით გადწყვეტის შემთხვევაში დაჭირდებოდა დაახლოებით ორი მილიონი წელი. კომპიუტერი მუშაობს ოპტიკურ კვანტურ გამომთვლელებზე (კუბიტები ბაზირდებიან ფოტონებზე).

2021 სელს ჩინელმა მეცნიერებმა შექმნეს კვანტური კომპიუტერის ორი პოტოტიპი:

1. ზეგამტარი კვანტური კომპიუტერი 66 კუბიტით;

2. კვანტური კომპიუტერი 113 აღმოჩენადი ფოტონებით (კუბიტებით).

2021 წლის ბოლოს IBM კომპანიამ წარმოადგინა თავისი ახალი კვანტური პროცესორი ზეგამტარ კუბიტებზე სახელწოდებით Eagle, რომელიც წარმოადგენს სუპერსწრაფი კომპიუტერების შექმნის პროგრამის ნაწილს. ახალ ჩიპს გააჩნია 127 კუბიტი, რაც ორჯერ აჭარბებს IBM-ის წინა კვანტურ კომპიუტერებს.

ახალი კვლევები ჩაატარეს ჩინელმა მეცნიერებმა. მათ შექმნეს გადახლართული ფოტონების წყვილი დედამიწაზე. ამ ფოტონებიდან ერთი გაასხივეს თანამგზავრზე, ხოლო მეორე დატოვეს დედამიწაზე. თანამგზავრი ძალიან მგრძნობიარე ფოტომიმდებია, რომელსაც შეუძლია დედამიწიდან გაშვებული თითოეული ფოტონის კვანტური მდგომარეობების დაფიქსირება. თანამგზავრის გაშვების მიზანი სწორედ კვანტური თვისებების უკეთ გაგება იყო. საბოლოოდ დადასტურდა, რომ დედამიწაზე და ორბიტაზე მყოფ ფოტონებს შორის გადაჯაჭვულობა ნამდვილად ხდებოდა.

ჩინელ მკვლევართა ჯგუფმა არა მარტო პირველი ობიექტის ტელეპორტაცია მოახდინა დედამიწიდან ორბიტაზე, არამედ მათ პირველებმა შეძლეს შეექმნათ თანამგზავრი-დედამიწის კვანტური ქსელი. მეცნიერების მიერ შემუშავებული შორ მანძილზე კვანტური ინფორმაციის გადაგზავნის ტექნოლოგია შემდგომში გამოყენებულ იქნება კვანტური ქსელების შესაქმნელად, რომლებიც გაცილებით უფრო ძლიერები და დაცულები იქნება ვიდრე არსებული ქსელები.

დასარულს კიდევ ერთხელ გავუსვათ ხაზი სამომავლოდ კვანტური კომპიუტერების შექმნის აუცილებლობის საკითხს და იმ სიძნელეებს რომლებსაც ადგილი აქვს ფართომასშტაბიანი კვანტური კომპიუტერების შექმნის საქმეში.

შესძლებელია სულ მალე დადგეს ის დრო, როცა ტრადიციული ელექტრონული კომპიუტერის სიმძლავრე, ინფორმაციის დამუშავების სიჩქარე და თანამედროვე ტექნოლოგია აღმოჩნდეს შეუსაბამო და არასკმარისი დროის მოთხოვნებთან მიმართებაში. თუმცა აქვე უნდა აღინიშნოს რომ რეალურად კვანტური კომპიუტერის აგება უკიდურესად ძნელია. ეს იმით არის გამოწვეული, რომ კვანტური კომპიუტერები ტრადიციული კომპიუტერებისგან განსხვავებით ისინი ჩიპების ნაცვლად იყენებენ ელექტრონებსა და ატომებს ინფორმაციის (კუბიტი) შესანახად, ატომები და ელექტრონები კი შეიძლება მარტივად იქნეს შეშფოთებულნი მათი გარემოთი, რაც გამოიწვევს კვანტურ კომპიუტერში შენახული ინფორმაციის დაზიანებას და ამდენად

შეცდომებს გამოთვლებში. გარდა ამისა, ასევე ძალიან რთულია კვანტური კომპიუტერებისთვის ნაწილების მოპოვება. ზეგამტარ კვანტურ კომპიუტერებს, როგორიცაა Google - ისა და IBM - ის მიერ აგებული კომპიუტერები, სჭირდებათ ჰელიუმ 3 (რომელიც ძალიან ძვირია), ბირთვული კვლევის ქვეპროდუქტი და სპეციალური ზეგამტარი კაბელები, რომლებიც დამზადებულია მხოლოდ იაპონური კომპანიის Coax Co - ს მიერ.

ამჟამად არსებობს კვანტური კომპიუტერების შექმნის რამდენიმე მიმართულება კვანტური ტექნოლოგიების თვალსაზრისით, რომელთა შემთხვევაში კვანტური კომპიუტერები შეიძლება აგებულ იქნეს: სუპერგამტარი კუბიტების, დაჭერილი იონების კუბიტების, ფოტონიკური კუბიტების და მბრუნავი კუბიტების გამოყენების საფუძველზე.

სუპერგამტარი კუბიტები - ეს არის კუბიტები, რომლებიც დაფუძნებული არიან ზეგამტარი მასალებისგან დამზადებულ ელექტრულ წრედებზე, რომლებსაც გააჩნიათ ორი დისკრეტული ენერგეტიკული დონე.

იონური კუბიტები - ეს არის კუბიტები, რომლებიც დაფუძნებული არიან დამუხტულ ატომებზე (იონებზე) რომლებიც დაჭერილი (შენარჩუნებული) არიან ხაფაგში ელექტრული ან მაგნიტური ველით, რაც გულისხმობს ელექტრული ან მაგნიტური ველების გამოყენებას იონების მოძრაობის დასაჭერად და გასაკონტროლებლად.

ფოტონური კუბიტები - ეს არის კუბიტები, რომლებიც ეფუძნება კვანტური ოპტიკის პრინციპებს სინათლის ნაწილაკებზე (ფოტონებზე), რომლებიც შეიძლება კოდირებულ იქნეს პოლარიზაციით ან სიხშირით, რაც გულისხმობს სინათლის გამოყენებას კვანტური მდგომარეობების მანიპულაციისა და კონტროლისათვის.

სპინური კუბიტები - ეს არის კუბიტები (სპინ კუბიტები), რომლებიც დაფუძნებული არიან ელექტრონის ან ატომის ბირთვის სპინზე (სპინის პრინციპი), რომელიც შეიძლება ორიენტირებულ იყოს ზევით ან ქვევით.

აღნიშნულიდან ჩანს, რომ საერთო ჯამში დღესდღეობით არსებობს რამდენიმე განსხვავებული კონკურენტი კვანტური ტექნოლოგია კვანტური კომპიუტერების შესაქმნელად, რომლებთაგან თითოეულს აქვს თავისი უნიკალური მხარეები და შეზღუდვებიც. კონკრეტული აპლიკაციისათვის ტექნოლოგიის არჩევანი დამოკიდე-

ბულია ისეთ ფაქტორებზე, როგორიცაა სასურველი შესრულება, მასალების ხელმისაწვდომობა და საოპერაციო საჭირო პირობები.

კვანტური კომპიუტერები თანამედროვე კომპიუტერებთან შედარებით გაცილებით სწრაფი, კომპაქტური და ეკონომიური იქნება, თუმცა აქვე უნდა აღვნიშნოთ, რომ კვანტური კომპიუტერი ძალზედ მყიფეა და ვერ უძლებს ვიბრაციების მაღალ დონეს.

მიუხედავად იმისა, რომ კვანტური გამოთვლითი ტექნოლოგია ჯერ კიდევ განვითარების ადრეულ ეტაპზეა, კვანტური კომპიუტერების რამდენიმე პოტენციური აპლიკაცია ავიაციაში არსებობს. აქ შესაძლებელია მოყვანილ იქნეს რამდენიმე მაგალითი:

ფრენის მარშრუტების ოპტიმიზაცია: კვანტურ კომპიუტერებს შეუძლიათ გააანალიზონ დიდი რაოდენობით მონაცემები და რთული ცვლადები ფრენის მარშრუტების ოპტიმიზაციისთვის, ისეთი ფაქტორების გათვალისწინებით, როგორიცაა ამინდის პირობები, საჰაერო მოძრაობა, საწვავის ეფექტურობა და მგზავრების კომფორტი.

თვითმფრინავის დიზაინი და სიმულაცია: კვანტურ კომპიუტერებს შეუძლიათ შეასრულონ რთული სიმულაციები და გამოთვლები, რომლებიც ამჟამად რთულია კლასიკური კომპიუტერებისთვის. მათ შეუძლიათ დაეხმარონ თვითმფრინავის სტრუქტურების, აეროდინამიკის და მასალების დიზაინსა და ოპტიმიზაციაში, გაზარდონ საწვავის მოხმარების ეფექტურობა და ფრენის უსაფრთხოება.

საჰაერო მოძრაობის მენეჯმენტი: კვანტურ გამოთვლას შეუძლია დაეხმაროს საჰაერო მოძრაობის ნაკადის მართვასა და ოპტიმიზაციაში. რეალურ დროში მრავალი წყაროდან მიღებული მონაცემების ანალიზით, კვანტურ კომპიუტერებს შეუძლიათ დაეხმარონ საჰაერო მოძრაობის კონტროლერებს მიიღონ უფრო ინფორმირებული გადაწყვეტილებები მარშრუტიზაციის, გადატვირთულობის შემცირებისა და უსაფრთხოების გაუმჯობესების შესახებ.

ამინდის პროგნოზირება: კვანტურ კომპიუტერებს შეუძლიათ ამინდის პროგნოზირების მოდელების გასაუმჯობესებლად ამინდის დიდი რაოდენობის დამუშავება და ანალიზი. ამან შეიძლება გამოიწვიოს ამინდის უფრო ზუსტი და დროული პროგნოზი, რაც ავიაკომპანიებს დაეხმარება ფრენების განრიგის ოპტიმიზაციაში, შეფერხებების შემცირებაში და მგზავრების უსაფრთხოების გაზრდაში.

დაშიფვრა და კიბერუსაფრთხოება: კვანტურ კომპიუტერებს აქვთ პოტენციური დაარღვიონ ამჟამად გამოყენებული დაშიფვრის მეთოდები, როგორიცაა RSA და ელიფსური მრუდის კრიპტოგრაფია. თუმცა, მათ ასევე შეუძლიათ შემოგთავაზონ დაშიფვრის ახალი ტექნიკა, რომელიც დაფუძნებულია კვანტურ პრინციპებზე, რამაც შეიძლება მნიშვნელოვნად გააძლიეროს საავიაციო სისტემებისა და კომუნიკაციების უსაფრთხოება.

მნიშვნელოვანია აღინიშნოს, რომ კვანტური გამოთვლა ჯერ კიდევ განვითარებადი ტექნოლოგიაა და არსებობს მრავალი ტექნიკური გამოწვევა, რომელიც უნდა დაიდლოს მისი სრული პოტენციალის რეალიზებამდე. თუმცა, საავიაციო ინდუსტრიის მკვლევარები და კომპანიები აქტიურად იკვლევენ და სწავლობენ კვანტური გამოთვლის პოტენციურ აპლიკაციებს საჰაერო მოგზაურობის სხვადასხვა ასპექტების გასაუმჯობესებლად.

კვანტურზე დაფუძნებული სენსორები: კვანტურ სენსორებს, როგორიცაა კვანტური გრავიმეტრები ან მაგნიტომეტრები, შეუძლიათ უაღრესად ზუსტი გაზომვები. ეს სენსორები შეიძლება გამოყენებულ იქნას ავიაციაში ისეთი ამოცანებისთვის, როგორიცაა გრავიტაციული ანომალიების გამოვლენა, მაგნიტური ველების რუკა ან ზუსტი ნავიგაცია.

თვითმფრინავის ტექნიკური მომსახურება და საიმედოობა: კვანტურ კომპიუტერებს შეუძლიათ დაეხმარონ თვითმფრინავის სისტემების, ძრავებისა და კომპონენტების სენსორული მონაცემების დიდი რაოდენობის ანალიზში. დახვეწილი შაბლონებისა და ანომალიების გამოვლენით, კვანტურ გამოთვლას შეუძლია უზრუნ-ველყოს პროგნოზირებადი ტექნიკური მომსახურება, შეამციროს დაუგეგმავი გაჩერება და გააუმჯობესოს თვითმფრინავის საიმედოობა.

კვანტური კომუნიკაცია და უსაფრთხო ქსელები: კვანტურ გამოთვლას შეუძლია ხელი შეუწყოს ავიაციისთვის უსაფრთხო საკომუნიკაციო ქსელების განვითარებას. კვანტური გასაღების განაწილების (QKD) პროტოკოლები გვთავაზობენ ურღვევ დაშიფვრას და შეუძლიათ დაიცვან მგრძნობიარე ინფორმაცია, რომელიც გადაეცემა თვითმფრინავებს, სახმელეთო სადგურებსა და საჰაერო მოძრაობის კონტროლს შორის.

სიმულაცია და ვირტუალური ტესტირება: კვანტურ კომპიუტერებს შეუძლიათ დააჩქარონ ახალი თვითმფრინავის დიზაინის, ძრავის კონფიგურაციისა და ფრენის

სისტემების სიმულაცია და ტესტირება. ამან შეიძლება მნიშვნელოვნად შეამციროს დამუშავების დრო და ხარჯები, რაც უფრო სწრაფი ინოვაციების საშუალებას იძლევა საავიაციო ინდუსტრიაში.

აღსანიშნავია, რომ მიუხედავად იმისა, რომ კვანტურ გამოთვლას აქვს უზარმაზარი პოტენციალი, ის ჯერ კიდევ განვითარებადი სფეროა და ავიაციაში მის პრაქტიკულ განხორციელებას შესაძლოა გარკვეული დრო დასჭირდეს. უწყვეტი კვლევა და განვითარება აუცილებელია ტექნიკური გამოწვევების მოსაგვარებლად, ტექნიკის შესაძლებლობების გასაუმჯობესებლად და საავიაციო აპლიკაციებისთვის მორგებული სპეციალიზებული ალგორითმების დასამუშავებლად.

ლიტერატურა

1. Баженов А. В. Основы бортовых вычислительных машин. Ставрополь, 2008. – 338 с.
2. Петров В.И., Матюхин К.Н. Цифровые вычислительные системы и их программное обеспечение. Воронеж, 2019. – 84 с.
3. Половов Р. М., Рошин А. Г. Бортовые вычислительные машины, системы и комплексы. Тексты лекции. Москва, МГТУГА, 2016. – 120 с.
4. Антимиров В.М., Телицин В. В., Бельский Л. Н. Системы автоматического управления: Бортовые цифровые вычислительные системы. Издательство ЮРАИТ. 2020. – 70 с.
5. Кучерявый А. А. Бортовые информационные системы: Курс лекций. Ульяновск, 2014. – 504 с.
6. Цехановский В.В., Бутирский Е.Ю., Жукова Н.А. Искусственные нейронные сети. Кнорус, 2023. – 352 с.
7. Злобин В.К., Ручкин В.Н. Нейросети и нейрокомпьютеры. BNV, 2015. – 256 с.
8. Агафонов Ф. Оптический компьютер : описание, принцип работы, преимущества. ABC – import.ru. 07.12. 2018. – 10 с.
9. Афанасев С.А. Оптические компьютеры и устройства ЭВМ, использующие оптические элементы. А 13 – 08. 2022ю
10. В. Силва. Practical Quantum Computing for Developers (Разработка с использованием квантовых компьютеров). Питер, 2020. - 352 с.
11. Michael A. Niesen and Isaak L. Chuang. Quantum Computation and Quantum Information. UK: Cambridge University press, 2010. – 704 p.
12. Rieffel, E.G., Polak, W.H. Quantum computing : A gentle introduction. MIT, 2011. – 386 p.
13. <https://www.cnet.com/tech/computing/quar-computer-makers-like-their-odds-for-big-progress-soon/>.
14. <http://www.neurnews.iu4.bmstu.ru>

სარჩევი

შესავალი.....	4
ლექცია 1. საბორტო კომპიუტერების ისტორია, დანიშნულება, შემადგენლობა, მუშაობის პრინციპი, სტრუქტურა და ფუნქციონირება.....	5
1.1. საბორტო კომპიუტერების განვითარების ისტორია.....	5
1.2. საბორტო კომპიუტერები და მათი მუშაობის განსაკუთრებულობები.....	12
1.3. საბორტო კომპიუტერების კლასიფიკაცია, შემადგენლობა და მუშაობის პრინციპი...16	
1.4. საბორტო კომპიუტერის სტრუქტურა და ფუნქციონირება.....	19
1.5. საბორტო კომპიუტერების ძირითადი მახასიათებლები.....	24
ლექცია 2. საბორტო კომპიუტერში ბრძანებებისა და რიცხვების წარმოდგენის ფორმატები და მათი ბრძანებების სისტემა.....	27
2.1. საბორტო კომპიუტერში ბრძანებებისა და რიცხვების წარმოდგენის ფორმატები.....	27
2.2. საბორტო კომპიუტერში ოპერაციების დამისამართების ხერხები.....	31
2.3. საბორტო კომპიუტერის ბრძანებების კოდებში დაპროგრამების განსაკუთრებულობები.....	39
2.4. საბორტო კომპიუტერების ბრძანებების სისტემა.....	44
ლექცია 3. საბორტო კომპიუტერის პროცესორის მართვისა და სინქრონიზაციის მოწყობილობა. მმართველი სიგნალების ფორმირება.....	48
3.1. საბორტო კომპიუტერის მუშაობის მართვის იერარქიული სისტემა.....	48
3.2. საბორტო კომპიუტერის მართვის მოწყობილობისა და სინქრონიზაციის განზოგადებული სტრუქტურა.....	51
3.3. ბრძანებების მართვის ბლოკის აგების სქემა და შემადგენლობა.....	53
3.4. მიკრობრძანებების ბლოკების აგება მმართველი სიგნალების სქემური ფორმირებით.....	56
3.5. მიკრობრძანებების ბლოკების აგება მმართველი სიგნალების პროგრამული ფორმირებით.....	59

ლექცია 4. საბორტო კომპიუტერის არითმეტიკულ-ლოგიკური მოწყობილობები.....	64
4.1. ზოგადი ცნობები არითმეტიკულ-ლოგიკური მოწყობილობების შესახებ.....	64
4.2. არითმეტიკულ-ლოგიკური მოწყობილობა ფიქსირებული მძიმით წარმოდგენილი რიცხვების შეკრება-გამოკლებისათვის.....	66
4.3. არითმეტიკულ-ლოგიკური მოწყობილობა ფიქსირებული მძიმით წარმოდგენილი რიცხვების გამრავლებისათვის.....	70
4.4. არითმეტიკულ-ლოგიკური მოწყობილობა მცურავი მძიმით წარმოდგენილი რიცხვების შეკრება-გამოკლებისათვის.....	74
4.5. საბორტო მიკროპროცესორულ სისტემებში არითმეტიკულ-ლოგიკური მოწყობილობები.....	82
ლექცია 5. პოგრამების წყვეტის სისტემა საბორტო კომპიუტერებში.....	84
5.1. წყვეტის სისტემების ფუნქციები და დანიშნულება.....	84
5.2. წყვეტის სისტემის მახასიათებლები.....	89
5.3. წყვეტის ხერხები.....	96
5.3.1. ვექტორული წყვეტა.....	96
5.3.2. წყვეტა პროგრამულად მართვადი პრიორიტეტით.....	97
5.4. საბორტო კომპიუტერის მუშაობის რეჟიმი გარე მოწყობილობებთან მუშაობის დროს და წყვეტების ინფორმაციული გაცვლის ორგანიზაციის განსაკუთრებულობები.....	102
ლექცია 6. საბორტო კომპიუტერის მეხსიერების მოწყობილობები.....	113
6.1. დამხსომებელ მოწყობილობების დანიშნულება, ძირითადი მახასიათებლები და კლასიფიკაცია.....	113
6.2. ოპერატიული დამხსომებელი მოწყობილობები.....	117
6.2.1. დინამიკური დამხსომებელი მოწყობილობები.....	118
6.2.2. სტატიკური დამხსომებელი მოწყობილობები	120
6.2.3. მეხსიერების უჯრედების მატრიცულ სტრუქტურაში გაერთიანება	122
6.3. მუდმივი დამხსომებელი მოწყობილობები.....	124
6.4. გარე დამხსომებელი მოწყობილობები.....	128
ლექცია 7. მეხსიერების ორგანიზაციის ხერხები ოპერატიულ დამხსომებელ მოწყობილობაში.....	131

7.1.მისამართიანი მეხსიერება.....	131
7.2. ასოციაციური მეხსიერება.....	133
7.3. სტეკური მეხსიერება.....	137
7.4. საბორტო კომპიუტერების სამისამართო სივრცე, დამისამართების სახეები და მათი რეალიზაციის ხერხები.....	139
7.4.1. საბორტო კომპიუტერის სამისამართო სივრცე	139
7.4.2. დამისამართების სახეები და მათი რეალიზაციის ხერხები საბორტო კომპიუტერში.....	140
ლექცია 8. საბორტო კომპიუტერების ინტერფეისული მოწყობილობები.....	150
8.1. ინტერფეისების დანიშნულება, კლასიფიკაცია და ძირითადი მახასიათებლები.....	150
8.2. ინტერფეისის სახეები.....	160
8.3. ინფორმაციის გადაცემის ხერხები, რეჟიმები, გაცვლის პრინციპები და გადაცემის გარემო.....	163
ლექცია 9. საბორტო კომპიუტერის შიგა მაგისტრალი.....	170
9.1. საბორტო შიგა ინტერფეისის განხორციელებისთვის მაგისტრალის გამოყენება და მისი საშუალებით მოწყობილობებს შორის მონაცემების გაცვლის ხერხები.....	170
9.2. პერსპექტიული შიგა მაგისტრალები.....	181
9.3. ARINC-429-ის მიხედვით ორპოლარული კოდით ინფორმაციის გაცვლა.....	184
ლექცია 10. ანალოგურ-ციფრული გარდამქმნელები და ციფრულ-ანალოგური გარდამქმნელები.....	192
10.1. ანალოგურ-ციფრული გარდამქმნელები.....	192
10.2. ციფრულ-ანალოგური გარდამქმნელები.....	200
ლექცია 11. საბორტო კომპიუტერების ექსპლუატაცია და მუშაობისუნარიანობის კონტროლი.....	211
11.1 საბორტო კომპიუტერის ექსპლუატაცია.....	211
11.2 საბორტო კომპიუტერის მუშაობისუნარიანობის კონტროლის მეთოდები	219
11.3. საბორტო კომპიუტერის მუშაობისუნარიანობის კონტროლის პროგრამული მეთოდები.....	220
11.4 საბორტო კომპიუტერის მუშაობისუნარიანობის ტესტური კონტროლი.....	226
11.5. საბორტო კომპიუტერის სქემური (აპარატული) კონტროლი	227

11.5.1. სიგნალების ჭარბი კოდირების ზოგადი ცნობები	228
11.5.2. შეცდომების მაკონტროლებელი კოდები	232
11.5.3. შეცდომების გამსწორებელი კოდები ხემინგის კოდი	234
11.5.4. კასკადური კოდები	238
ლექცია 12. საბორტო გამოთვლითი სისტემები.....	242
12.1. საბორტო გამოთვლითი სისტემების კლასიფიკაცია.....	242
12.2. საავიაციო რადიოელექტრონული კომპლქსის საბორტო ციფრული გამოთვლითი სისტემების პროგრამული უზრუნველყოფა.....	252
12.3. პროგრამების შესრულება, რომლებიც ხისტად არიან დაკავშირებული დროით დიაგრამასთან.....	256
12.4. პროგრამების შესრულება ასინქრონული განაცხადების მიხედვით.....	259
ლექცია 13. ბაგეტის ტიპის კომპიუტერები და მათი გამოყენება.....	265
13.1. აპარატულ-პროგრამული გამოთვლითი პლატფორმის „ბაგეტის“ კონცეფცია და ელემენტური ბაზა.....	265
13.2. ბაგეტის ტიპის საბორტო კომპიუტერებში გამოყენებული პროცესორული მოდულები.....	271
13.2.1. პროცესორული მოდული BT33-205.....	271
13.2.2. პროცესორული მოდული BT01-212.....	273
13.2.3. პროცესორული მოდული BT83-201.....	279
13.3. საფრენ აპარატებზე გამოყენებული ბაგეტის ტიპის კომპიუტერები.....	280
13.3.1. მმართველი კომპიუტერი ბაგეტი 33	284
13.3.2. საბორტო კომპიუტერი ბაგეტი 53-18	282
13.3.3 საბორტო კომპიუტერი ბაგეტი 53-23	283
13.3.4. საბორტო კომპიუტერი ბაგეტი 53-31	285
13.3.5. საბორტო კომპიუტერი ბაგეტი 55	288
13.3.6. მმართველი საბორტო კომპიუტერი ბაგეტი 83	296
13.4. საავიაციო საბორტო კომპიუტერი ტაიბერი.....	298
ლექცია 14. საბორტო კომპიუტერების და სისტემების საიმედოობა.....	300
14.1. საიმედოობის ძირითადი ტერმინები და განსაზღვრებები.....	300

14.2. არააღდგენადი საბორტო კომპიუტერების საიმედოობის საიმედოობის ძირითადი მაჩვენებლები.....	305
14.3. აღდგენადი საბორტო კომპიუტერების საიმედოობის ძირითადი მაჩვენებლები....	317
14.4. საბორტო კომპიუტერების სტრუქტურული საიმედოობა და მისი ამაღლების მეთოდები.....	327
14.5. საბორტო გამოთვლითი სისტემების საიმედოობის ამაღლება ავტომატური რეკონფიგურაციის საშუალებით.....	337
ლექცია 15. საბორტო კომპიუტერების განვითარების პერსპექტივები.....	342
15.1. ოპტიკური კომპიუტერები.....	343
15.2. ნეიროკომპიუტერები.....	353
15.3. კვანტური კომპიუტერები.....	370
ლიტერატურა	386