

HAKIMOV Z.A.  
HAMRAYEVA S.I.

# KOMPYUTERNI TASHKIL ETILISHI



O'QUV QO'LLANMA

O'ZBEKISTON RESPUBLIKASI OLIY TA'LIM, FAN VA  
INNOVATSIYALAR VAZIRLIGI

MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT  
AXBOROT TEXNOLOGIYALARI UNIVERSITETI  
URGANCH FILIALI

HAKIMOV Z.A.  
HAMRAYEVA S.I.

# KOMPYUTERNI TASHKIL ETILISHI

O'zbekiston Respublikasi Oliy ta'lim, fan va innovatsiyalar  
vazirligi tomonidan o'quv qo'llanma sifatida tavsiya etilgan

5330500 "Kompyuter injiniringi (*Kompyuter injiniringi, AT-servis*)",  
5330600 "Dasturiy injiniring", 5350100 "Telekommunikatsiya  
texnologiyalari (*Telekommunikatsiyalar*)", 5350400 "Axborot  
kommunikatsiyalari sohasida kasb ta'limi" yo'nalishlarida tahsil  
olayotgan talabalar uchun o'quv qo'llanma

Toshkent  
"METODIST NASHRIYOTI"  
2024

UDK: 004.3(075.8)

BBK: 32.973ya7

H 25

**Hakimov Z.A.**

**Kompyuterni tashkil etilishi/ Hamrayeva S.I./ O'quv qo'llanma.**  
– Toshkent: “METODIST NASHRIYOTI”, 2024. – 196 b.

Ushbu o'quv qo'llanma 5330500-“Kompyuter injiniringi (Kompyuter injiniringi, AT-servis)”, 5330600-“Dasturiy injiniring”, 5350100-“Telekommunikatsiya texnologiyalari (Telekommunikatsiyalar)”, 5350400-“Axborot kommunikatsiyalari sohasida kasb ta'limi” yo'nalishlari talabalari uchun asosiy va boshqa mustaqil o'rganuvchilar uchun qo'shimcha qo'llanma sifatida foydalanishlari mumkin.

O'quv qo'llanmada zamonaviy hisoblash mashinalarining ichki va tashqi qurilmalari, mantiqiy tashkil etuvchilari, buyruqlar, operandlar va adreslar haqida nazariy kurs, shu jumladan xotira ierarxiyasi va darajalari, tashqi qurilmalar yordamida kompyuterlarning funksional imkoniyatlarini oshirish usullari bayon etilgan.

**Taqrizchilar:**

**PhD O. K. Xo'jayev**, Toshkent axborot texnologiyalari universiteti Urganch filiali “Axborot texnologiyalari” kafedrasini mudiri

**T.f.n B. S. Rahimov**, Toshkent tibbiyot akademiyasi Urganch filiali “Biofizika va axborot texnologiyalari” kafedrasini mudiri

O'zbekiston Respublikasi Oliy va o'rta maxsus ta'lim vazirligining 2022-yil 9-sentabrda 302-sonli buyrug'iga asosan nashrga ruxsat berilgan.

ISBN 978-9910-03-214-1

© Z.A.Hakimov S.I.Hamrayeva, 2024.  
© “METODIST NASHRIYOTI”, 2024.

## KIRISH

“Kompyuterni tashkil etilishi” fani bo'yicha ushbu qo'llanma 5330500 “Kompyuter injiniringi (Kompyuter injiniringi, AT-servis)”, 5330600 “Dasturiy injiniring”, 5350100 “Telekommunikatsiya texnologiyalari (Telekommunikatsiyalar)”, 5350400 “Axborot kommunikatsiyalari sohasida kasb ta'limi” yo'nalishlari talabalari uchun mo'ljallangan bo'lib, talabalarda nazariy bilimlarni, amaliy ko'nikmalarni, asosiy raqamli mantiqiy elementlar va ulardan kompyuterni tashkil qilishda foydalanish, kompyuterni funksional tashkil etilishi, zamonaviy kompyuterlarning funksional tarkibiy qismlari, mashina kodlarining xususiyatlari, buyruq formatlari va manzil rejimlari, xotira ierarxiyasi va tashkil etilishi, protsessor va periferik qurilmalar o'rtasidagi ulanish va aloqasiga oid bilimlarni shakllantirish uchun mo'ljallagan.

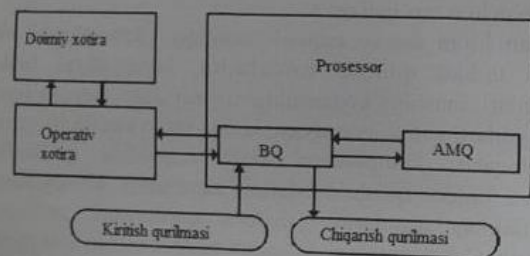
Shu bilan birga asosiy raqamli mantiqiy elementlar va ulardan kompyuterni tashkil qilishda foydalanish, kompyuterni funksional tarkibiy qismlari, mashina kodlarining xususiyatlari, buyruq formatlari va manzil rejimlari, xotira ierarxiyasi, kompyuter va periferik qurilmalar o'rtasida aloqalar, kompyuterni tashkil qilishda qo'llaniladigan vositalarni va tashkil qilish asoslarini, zamonaviy kompyuterlarning funksional tarkibiy qismlarini, periferik qurilmalari, protsessor va periferik qurilmalar o'rtasidagi ulanishni tashkil qilish, kompyuterdagi hisoblash jarayonlari turlarini tahlil qilish hamda tashkil etish, kompyuter tizimlarini ishlatish jarayonida yuz beradigan nosozliklarni bartaraf etish bo'yicha zamonaviy usullarni o'rganish va ishlab chiqish, kompyuter tizimi unumdorligini nazorat qilish va unumdorlikni oshirish usullari bayon etilgan.

# 1. Kompyuter arxitekturasi rivojlanish bosqichlari. Kompyuterlarni tashkil etish tamoyillari va klassifikatsiyasi

## 1.1. Arxitektura

Kompyuter arxitekturasi deganda kompyuterni umumlashtirilgan xolda tasvirlash tushuniladi. Bunda dasturlash imkoniyatlari, komandalar tizimi, adreslash tizimi, xotirani tashkil qilinishi va shunga o'xshash boshqa imkoniyatlari tasvirlanadi (yoki shular haqida tasavvur hosil qilinadi). Arxitektura harakat tamoyilini, axboriy aloqalarni va asosiy mantiqiy tashkil etuvchilarni (protssessor, O3Y (tezkor xotira), tashqi xotira qurilmasi, tashqi qurilmalar) aniqlab beradi.

Kompyuter strukturasi uning funksional elementlari va ular orasidagi munosabatlarning jamlanmasidir. Funksional elementlar deganda barcha kattayu-kichik elementlar nazarda tutiladi. Masalan: oddiy mikrosxemalar, ona plata va h.k.



1.1-rasm. Fon Neyman arxitekturasi bo'yicha kompyuter qurilmalarining o'zaro ta'siri sxemasi (Belgilar: BQ – boshqarish qurilmasi, AMQ - arifmetik mantiqiy qurilma)

Kompyuterlarni qurishda eng ko'p tarqalgan quyidagi arxitekturalar mavjud.

## 1.2. Klassik arxitektura

Fon Neyman arxitekturasi. Bunda ma'lumotlarni o'tkazishni tashkil etuvchi bitta arifmetik-mantiqiy qurilma (AMQ) va komandalar potokini o'tkazuvchi bitta boshqarish qurilmasi (BQ) mavjud.

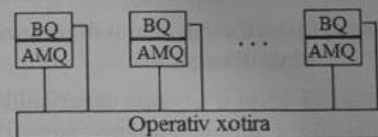
Bu bir protssessorli kompyuter. Bunda kompyuterni tashkil etuvchi barcha funksional bloklar sistema magistrali deb nomlanuvchi umumiy shina orqali o'zaro bog'langan. Magistral shina deganda elektron sxemalarni ulash uchun uyalar mavjud bo'lgan ko'p qatorli o'tkazgich

liniyasi tushuniladi. uch xil magistral shinalar mavjud: ma'lumotlar shinasi, adres shinasi va boshqarish shinasi.

Tashqi qurilmalar esa kontroller deb ataluvchi tashqi qurilmalarni boshqarish qurilmalari orqali ulanadi. Kontroller tashqi qurilmani yoki aloqa kanalini markaziy protssessor bilan ulaydi va tashqi qurilmani boshqarish vazifasini o'ziga oladi, bu esa protssessorni ish unumdorligi oshishiga olib keladi.

## 1.3. Ko'p protssessorli arxitektura

Kompyuterda bir nechta protssessorning mavjudligi ko'p oqimli ma'lumotlar va ko'p oqimli komandalarni parallel tashkil qilish mumkinligini bildiradi. Bitta masala bo'laklarga bo'lingan xolda bir vaqtning o'zida barcha fragmentlarni bajarish imkoniyati mavjud bo'ladi. Bunday mashinalar bitta umumiy O3Yga ega bo'ladi va quyidagi shaklda tasvirlash mumkin



1.2-rasm. Ko'p protssessorli kompyuter arxitekturasi

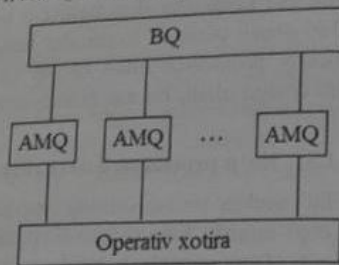
## Ko'p mashinali hisoblash tizimlari

Bunday tizimni tashkil qilishda tizimga kiruvchi har bir protssessor/mashina lokal operativ xotirasiga ega bo'lishini ta'kidlash lozim. Bu mashinalarning har biri klassik arxitekturali bo'ladi. Bu tizimda masalani yechish uchun masalani shunday qismlarga ajratish kerakki, ularning o'zaro bog'liqligi maksimal darajada kamaysin va shu bilan bir vaqtda tizimda nechta kompyuter bo'lsa shuncha qismga ajratilsin (yoki masala nechta qismdan iborat bo'lsa shuncha mashina jalb qilinadi)

## 1.4. Parallel protssessorli arxitektura

Bunday tizimda bir nechta AMQ bitta boshqarish qurilmasi (BQ) boshqaruvida ishlaydi. Bundan ko'rinadiki, ma'lumotlarni qayta ishlash faqatgina bitta programma orqali, boshqacha aytganda bitta buyruqlar potoki (oqimi) orqali bajariladi. Bu arxitekturada yuqori samaradorlikka

erishish mumkin, qachonki bir xil bo'lgan hisoblashlar jarayoni muayyan vaqtda turli AMQlarda amalga oshirilsa.



1.3-rasm. Parallel protsessor arxitekturasi

Zamonaviy hisoblash mashinalarida ishlab chiqaruvchining yondoshuvidan kelib chiqib turli arxitekturalar qo'llaniladi.

#### 1.5. Kompyuterlarni tashkil etish tamoyillari (prinsiplari) va klassifikatsiyasi

Hisoblash mashinalarini tashkil etishda amerikalik olim Jon fon Neymanning 1945 yilda olg'a surgan umumiy tamoyillari ishlatiladi. Bular quyidagilar:

- 1. Ikkilik kodlash tamoyili.** Barcha ma'lumotlar ikkilik kod ko'rinishida mavjud bo'ladi.
- 2. Dasturiy boshqarish tamoyili.** Protsessor aniq bir ketma-ketlikda yozilgan komandlarni bajaradi. Bu programma yoki algoritmi deyiladi. Xotiradan bajariladigan dasturni tanlab olish komandalar hisoblagichi orqali bajariladi (счетчик команд). Protsessorning bu registri o'zida saqlanayotgan keyingi keladigan komanda adresini buyruq (komanda) uzunligigacha ketma-ket uzaytiradi. Dasturdagi komandalar ketma-ket joylashganligini inobatga oladigan bo'lsak, komandalar zanjirini tanlash ketma-ket joylashgan yacheykalarda bajariladi. Agar tanlab olishda keyingi emas balki boshqa komandaga o'tish kerak bo'lib qolsa, shartli yoki shartsiz operatorlaridan foydalaniladi. Bu operatorlar esa komandalar hisoblagichiga o'tilishi kerak bo'lgan yacheyka adresini ko'rsatadi. Xotiradan komandalarni tanlash jarayoni "stop" komandasiga yetish bilan tugaydi. (Shu tartibda protsessor dasturni inson omilisiz, avtomatik tarzda bajaradi (avtomatik tizim inson ishtirokisiz, yarim avtomat inson ishtiroki bilan, zavodlar misolida)

**3. Xotiraning bir xilligi tamoyili (prinsipi).** Barcha axborotlar – u dastur bo'lsin yoki ma'lumot, bitta xotirada joylashadi. Kompyuter uchun biror konkret yacheykada raqam, matn yoki komanda, nima joylashganligining umuman farqi yo'q. Kompyuter (protsessor) komandalari bilan xuddi ma'lumotlar kabi amallarni bajaradi (ya'ni hech kanday farq yo'q). Bu esa ko'p imkoniyatlarga yo'l ochadi. Masalan, dastur bajarilishi jarayonida o'zini o'zi qayta ishlashga majburlanishi mumkin (sikllar va podprogrammalar, funksiyalar). Bundan tashqari bir dasturning komandalari boshqa dasturning hisoblash natijalari ham bo'lishi mumkin. Bu tamoyilga translyatsiya (kompilyatsiya) uslublari (YUDTdan mashina tiliga tarjima) asoslanadi.

**4. Adreslilik (manzillilik) tamoyili (prinsipi).** Asosiy xotira raqamlangan yacheykalardan tashkil topgan bo'ladi. Protsessor istalgan vaqtda istalgan xotira yacheykasiga bevosita murojaat qila oladi. Bundan esa xotira qismlariga nom berish imkoniyati mavjudligi ko'rinadi, bu esa, o'z navbatida shu nom orqali unga murojaat qilish yoki undagi ma'lumotni o'zgartirish imkonini beradi. (*yacheyka adresi o'zgarimas, ichidagi ma'lumot o'zgaruvchan*) Yuqoridagi tamoyillar asosida qurilgan kompyuterlar fon-neyman tipidagi kompyuterlari kiradi.

#### Nazorat uchun savollar:

1. Arxitektura tushunchasi.
2. Bir protsessorlik kompyuter tuzilishi va funksiyalari.
3. Ko'p protsessorli kompyuter arxitekturasi va xususiyatlari.
4. Kompyuterlarni tashkil etish tamoyillarini keltiring.

## 2. Kompyuterlarni tashkil qilishning raqamli mantiqiy asoslari

AMQ arifmetik va mantiqiy amallarni bajaruvchi qurilma bo'lib, Markaziy prosessorning asosiy qismi hisoblanadi. AMQ ma'lumotlarni qayta ishlashning muhim bosqichini amalga oshiradi. AMQda bajariladigan amallar 3 turli bo'ladi: arifmetik, mantiqiy va bitlar ustidagi amallar.

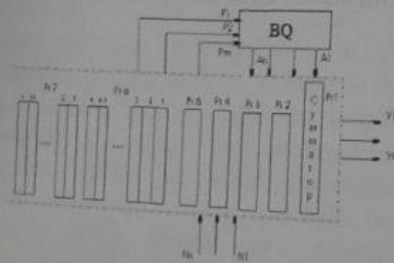
Arifmetik amal deganda argumentlari va hisoblash natijalari raqamlar bo'lgan hisoblash jarayoni tushiniladi (qo'shish, ayirish, ko'paytirish, bo'lish). Mantiqiy amal deganda murakkab so'zlarni xosil qilish jarayoni tushiniladi (va, yoki, emas va x.k). Bitlar ustidagi amallar deganda ko'chirish amali tushiniladi.

J. fon Neyman birinchi bo'lib AMQning ishlab chiqqan. Neymanning ta'kidlashicha AMQ kompyuter uchun xayotiy zaruratdir, chunki faqat ugina qo'shish, ayirish, ko'paytirish, bo'lish kabi matematik amallarni bajaradi.

### 2.1. AMQ strukturasi

AMQ lar registrlar va summatorlardan iborat bo'ladi. Qurilma o'ziga uzatiladigan operatsiyalar kodlari bilan mos ishlaydi va 2 ta funktsional qismdan tashkil topadi:

1. Mikroprogrammali qurilma (boshqarish qurilmasi) – komandalarni uzatuvchi qism
2. Operatsion qurilma (AMQ) – berilgan komandalar ketma-ketligi bajariladigan qism.



2.1-rasm. AMQning struktura sxemasi

Bu sxemadan ko'rinib turibdiki, AMQ tarkibiga kiradigan Rg1 - Rg7 registrlarida N1, N2, ..., NS xotiradan uzatiladigan ma'lumotlar

qayta ishlanadi. Mantiqiy sxemalar esa boshqarish qurilmasidan keluvchi mikrokomandalarni bajaradi.

A1, A2, ..., An-1, An mikrokomandalarining ketma-ket yozilishidan hosil qilingan mikroprogramma ma'lumotlarni qanday tartibda qayta ishlanishini belgilaydi. Mikrokomandalar tashqi (sxemadagi A1, A2, ..., An-1, An) va ichki (r1, p2, ..., rm) bo'ladi. Ichki mikrokomanda boshqarish qurilmasidan ketma-ket kelayotgan komandalarning ketma-ketligini buzish va ulardan oldin bajarilib ketishi ham mumkin.

AMQning hisoblash natijalari y1, y2, ..., yn shinalari orqali operativ xotiraga yoziladi. AMQga kiruvchi registrlarning funksiyalari quyidagilar:

- Pr1 – summator (lar), asosiy registr, bu yerda hisoblash natijalari shakllanadi
- Pr2, Pr3 – qo'shiluvchi, ko'paytma, bo'linuvchi, bo'luvchilar registri (bajariladigan amal turiga qarab)
- Pr4 – adres registri, operandlar adreslari va natija adresini saqlashga mo'ljallangan
- Pr6 – adreslar hosil qilish registri
- Pr7 – yordamchi registrlar, jarayonning borishiga qarab akkumulator, indeks registri yoki oraliq natijalarni saqlovchi sifatida ishlatiladi.

Ba'zi registrlar dastur orqali boshqarilishi va hisoblash jarayonida komandalar ulardagi ma'lumotlar bilan bevosita ishlashi mumkin. Summator, indeks registri, yordamchi registrlar

Boshqa registrlarni dasturiy boshqarish imkoniyati yo'q, chunki ularni dasturiy adreslash imkoniyati mavjud emas.

Operatsion qurilmalarni qayta ishlanuvchi ma'lumot shakli, qayta ishlash turi va mantiqiy shakliga ko'ra tasniflash mumkin.

AMQlar 4 turdagi ma'lumot obektlari bilan ishlashi mumkin: mantiqiy (1 bit), raqamli (4 bit), baytli (8 bit) va adresli (16 bit) AMQda ma'lumotlarni uzatish yoki o'zgartirishning 51 xil operatsiyalari bajariladi. Adreslashning 11 xil usuli mavjudligini inobatga olsak (7 ta ma'lumotlar uchun va 4 ta adreslar uchun) "KOMANDA/ADRESLASH REJIMI" kombinatsiyasini turli shaklda qo'llab, 255 xil komandalarni hosil qilish mumkin.

### 2.2. AMQning tasniflanishi

Operandlar bilan bajariladigan amallar uslubiga ko'ra. Parallel va ketma-ket bo'ladi. Ketma-ket AMQlarda operandlar ketma-ket kodlarda

beriladi, amallar esa ularning aloxida razryadlari bilan ketma-ket bajariladi. Parallel AMQLarda operandlar parallel beriladi va amallar bir vaqtda barcha razryadlar bilan bajariladi.

Sonlarni ifodalash usuliga ko'ra.

- qo'zg'almas vergulli sonlar bilan ishlovchi
- qo'zg'aluvchi vergulli sonlar bilan ishlovchi
- 10lik sonlar bilan ishlovchi

Elementlar va uzellarni ishlatish xarakteriga ko'ra blokli va ko'p funksiyali AMQLar. Blokli AMQLarda hisoblashlar alohida bloklarda va parallel bajariladi. Afzalligi xisoblash tez bajariladi, kamchiligi qurilma harajatlari oshib ketadi. Ko'p funksiyali AMQLarda talab qilinadigan ish rejimiga mos ravishda o'zgarib turuvchi sxemalarda bajariladi.

O'z funksiyasiga ko'ra AMQ hisoblashlarni bajaruvchi, boshqa qurilmalardan (mas xotiradan) ma'lumot oluvchi, ularni o'zgartiruvchi va hisoblash natijasini boshqa qurilmaga uzatuvchi operatsion blok hisoblanadi.

### 2.3. AMQdagi amallar

AMQda bajariladigan amallarni quyidagi guruxlarga bo'lish mumkin:

- ikkilik sonlar ustida amallar
- o'n oltilik sonlar ustida amallar
- o'nlik arifmetika
- indeks arifmetikasi
- mantiqiy amallar

Zamonaviy hisoblash mashinalari yuqorida ko'rsatilgan barcha amallarni bajaradi. Boshqa hisoblash mashinalari, masalan kichik EXM, mikroEXM, maxsus mashinalar ba'zilar qo'zg'aluvchi vergulli sonlar bilan ishlamasa, ba'zilar o'nli arifmetika bilan ishlamaydi, ba'zilar esa alfavit-raqamli maydonlar bilan ishlay olmaydi. Bu operatsiyalar maxsus quyi dastur yordamida bajariladi. Bularga misol qilib turli radarlar, kuzatuv kameralari va ularni boshqaruvchi qurilmalarni ko'rsatish mumkin.

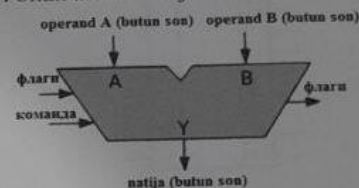
### 2.4. Arifmetik-mantiqiy qurilma (AMQ)

AMQ – ikkilik sonlar ustida arifmetik va mantiqiy amallarni bajarishga mo'ljallangan qurilmadir. AMQ mikroprotsessor tarkibida bo'ladi. (ALU-Arithmetic Logic Unit)

Operandlar bilan ishlash prinsipiga ko'ra AMQ 2 turga bo'linadi

1. Parallel AMQ. Bunda operandlar parallel kod shaklida ifodalanadi va amallar bir vaqtda operandlarning barcha razryadlari bilan bajariladi.

2. Ketma-ket AMQ. Operandlar ustida amallar ketma-ket ravishda, ularning razryadlari bilan alohida bajariladi.



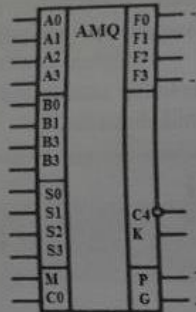
2.2-rasm. Arifmetik mantiqiy qurilma

### AMQning ishlash prinsipi

#### 2.1-jadval

#### K155III3 AMQ da bajariladigan operatsiyalar ro'yhati

Tanlash signallar funksiyasi				Mantiqiy funksiyalar (M=1)	Arifmetik-mantiqiy funksiya va $C_0=0$ (M=0)
$S_3$	$S_2$	$S_1$	$S_0$		
0	0	0	0	$\bar{A}$	A
0	0	0	1	$\bar{A} + B$	A+B
0	0	1	0	$\bar{A}B$	$A + \bar{B}$
0	0	1	1	0	-1
0	1	0	0	$\bar{A}\bar{B}$	$A + \bar{A}\bar{B}$
0	1	0	1	$\bar{B}$	$(A+B) + \bar{A}\bar{B}$
0	1	1	0	$A \oplus B$	A-B-1
0	1	1	1	$A\bar{B}$	$\bar{A}\bar{B}-1$
1	0	0	0	$\bar{A}+B$	A+AB
1	0	0	1	$A \oplus B$	A+B
1	0	1	0	B	$(A+\bar{B})+AB$
1	0	1	1	AB	AB-1
1	1	0	0	1	A+A
1	1	0	1	$A + \bar{B}$	$(A+B)+A$
1	1	1	0	A+B	$(A+\bar{B})+A$
1	1	1	1	A	A-1

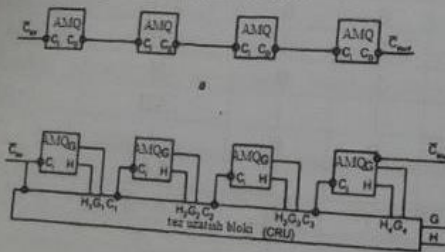


2.2-rasm. AMQning elektrik sxemasi

Bu rasmda 16 ta arifmetik va 16 ta mantiqiy amallarni bajaruvchi AMQ keltirilgan. Bunda  $A_0...A_3$  va  $V_0...V_3$  lar operand sifatida tanlangan va ular ustida arifmetik va mantiqiy amallar bajariladigan to'rt razryadli A va V sonlarining kirishlaridir.

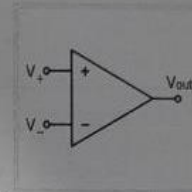
Bajariladigan amal  $S_0...S_3$  kirishlaridagi signallar kombinatsiyasi bilan aniqlanadi.  $M=1$  bo'lganda mikrosxema mantiqiy amallarni bajaradi,  $M=0$  bo'lganda arifmetik amallarni bajaradi.

Mikrosxema ham musbat ham manfiy mantiqda ishlay oladi. Musbat mantiqda A, V va S kirishlariga haqiqiy qiymatlari kiradi,  $F_0...F_3$ , P, G chiqishlarida haqiqiy qiymatlar olinadi. Manfiy mantiqda A, V va S kirishlariga signalning invers qiymati kiritiladi,  $F_0...F_3$ , P, G chiqishlarida invers qiymatlari olinadi. Razryadliliği 4 dan katta bo'lgan sonlar bilan amallar bajarishda  $C_n$  kirishi ishlatiladi. AMQ sxemasini sonlarni teskari koddan to'g'ri kodga o'tkazish va aksincha ishlatish mumkin. Bundan AMQning funksional jihatdan moslashuvchan ekanligini ko'rish mumkin.



2.3-rasm. AMQlarni ketma-ket ulash va parallel ulash

$F_0...F_3$  chiqishlarida chiqish so'zi (8 bit axborot) shakllanadi,  $C_n$  chiqishi esa keyingi AMQga ulanish uchun qo'llaniladi, G va H chiqishlari tezkor summatorlarda ko'chirishni ta'minlovchi generatsiya va ko'chirishni uzatish funksiyalarini beradi. K chiqishi esa tenglikni aniqlovchi komparator chiqishini bildiradi.



2.4-rasm. Komparatorning blok sxemasi

### Komparator

Komparator (lotincha comparare – "tenglash") tenglovchi qurilma.

Bu elektron sxema bo'lib, ikkita kirish va bitta chiqishga egadir. Agar musbat kuchlanishning qiymati yuqori bo'lsa chiqishda yuqori qiymatli signal beriladi, agar manfiy kuchlanishning qiymati yuqori bo'lsa chiqishda past qiymatli signal beriladi. Mantiqiy sxemalarni qurishda yuqori qiymatli signal (mantiqiy) 1 ga teng, past qiymatli signal (mantiqiy) 0 ga teng deb olinadi.

Komparatorlar orqali kuchlanish va mantiqiy o'zgaruvchilari orasidagi kabi raqamli qurilmalarning uzluksiz signallar orasida munosabat ta'minlanadi. Komparatorlar turli elektron qurilmalarda, signalizatsiya qurilmalarida, o'tkazish-nazorat punktlarida qo'llaniladi.

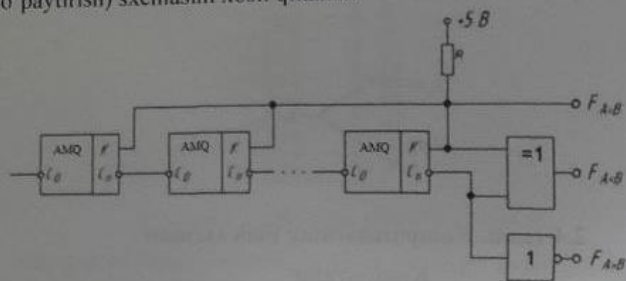
Kiruvchi signallarning birini tayanch kuchlanish deb olinadi. Tayanch kuchlanish keyingi komparatorning kirishiga borgan signal diapazonini ikkiga ajratadi. Komparatorning chiqish qiymati keyingi komparatorning kirish qiymati qaysi diapazonda ekanligini aniqlaydi. Bitta tayanch kirishiga ega bo'lgan komparator bir tayanchli komparator, ikkita tayanch kirishiga ega bo'lgani ikki tayanchli komparator deyiladi va h.k.

Mikrosxemadan chiqadigan komparator qiymati quyidagi funksiyaning qiymatiga teng:

$$F_{A=B} = \overline{F_0} \overline{F_1} \overline{F_2} \overline{F_3}$$

Bunda AMQ ayirish rejimida ishlaydi (A-V yoki A soniga inversiv V sonini qo'shish amalga oshiriladi). Agar  $A=V$  bo'lsa barcha kirish razryadlarida 0 lar bo'ladi, F ning barcha inverslangan qiymatlari 1 teng

bo'ladi va  $F_{A-B}$  funksiyaning qiymati 1 ga teng bo'ladi. Boshqa barcha holatlarda (kiruvchi qiymatlarning hech bulmaganda 1 donasi 0 ga teng bo'lib qolsa)  $F$  ning qiymati 0 ga teng bo'ladi. Bir nechta AMQning K chiqishlarini parallel ulash natijasida "VA" mantiqiy (mantiqiy ko'paytirish) sxemasini xosil qilamiz.



### 2.5-rasm. AMQlar yordamida tenglikni aniqlash sxemasi

Ushbu rasmdan ko'rinib turibdiki,  $F_{A-B}$  signalini ko'chirish signali bilan kombinatsiyalash natijasida tenglik bilan bir vaqtda tengsizlik qiymatini ham hosil qilish mumkin ( $F_{A<B}$  yoki  $F_{A>B}$ )

Agar  $A < B$  bo'lsa  $F_{A-B} = 0$  bo'ladi va ayirish amalini bajarishda yuqori razryaddan ko'chirish bajariladi va  $F_{A-B} = 1$  bo'ladi. Agar  $A > B$  bo'lsa,  $F_{A-B} = 0$  bo'ladi va ko'chirish bajarilmaydi va natijada  $F_{A-B} = 1$  bo'ladi.

Agar  $A > B$  bo'lsa,  $F_{A-B} = 0$  bo'ladi va ayirish amalini bajarishda yuqori razryaddan ko'chirish bajarilmaydi va mos ravishda  $F_{A-B} = 0$  bo'ladi va  $F_{A-B} = 1$  bo'ladi.

Mantiqiy amallar xar bir razryadda mustaqil bajariladi. Arifmetik amallar ko'chirish va almashtirishni inobatga olib bajariladi.

Katta hajmli so'zlar bilan ishlaganda AMQlar ketma-ket bajariladi. Bu xolatda sxemaning tezkorligiga ko'chirishni tarqatish tezligi katta ta'sir ko'rsatadi. AMQ sxemasida ko'chirish vaqtini maksimal kamaytirishga harakat qilinadi, lekin katta razryadli so'zlarni (16 bit, 32 bit) qo'shishda vaqt oshib ketishi mumkin. Bunday xolatlarda AMQ bilan birgalikda ko'chirishni tezlatish bloki deb nomlanuvchi mikrosxemani ishlatiladi. Bu mikrosxemada ko'chirish jarayoni  $G$  va  $H$  chiqishlarini generatsiya qilish orqali bajariladi. Bitta tezkor o'tkazish bloki bir nechta AMQlarning tezkor o'tkazishlarini tashkil qiladi.

Shuni alohida ta'kidlash lozimki, yuqorida ko'rib o'tilgan kombinatsion tipdagi funksional bloklardan tashqari xisoblash

mashinalarida yuqori razryad ko'rsatkichlari, nazorat uzellari va boshqa uslublar keng qo'llaniladi. Bu haqda xisoblash mashinalarini loyixalash bo'yicha maxsus adabiyotlardan ma'lumot topish mumkin.

### 2.5. Arifmetik summatorlar.

Summatorlar mantiqiy so'zlarni qo'shish amalini bajaradi. Qo'shish jarayonida arifmetik yig'indini olish va boshqa qo'shimcha amallar bajariladi. Barcha amallar AMQlarda bajariladi, AMQlarning asosiy yadrosini summatorlar tashkil etadi.

Kiruvchi elementlar soniga qarab yarim summatorlar, bir razryadli summatorlar va ko'p razryadli summatorlar farqlanadi. Ko'p razryadli summatorlar ketma-ket va parallel summatorlarga bo'linadi. Ketma-ket summatorlarda ma'lumotlarni qayta ishlash razryadma razryad bitta qurilmada bajariladi, parallel summatorlarda esa qayta ishlanuvchi ma'lumotlar hamma razryadlari birdaniga qayta ishlanadi va har bir razryad uchun alohida qurilma mavjud bo'ladi.

Razryadlararo o'tkazishni tashkil qilish uslubi bo'yicha parallel summatorlar ketma-ket o'tkazuvchi sxemalar, parallel o'tkazuvchi sxemalar va guruhli o'tkazuvchi strukturalarga bo'linadi. Guruhli o'tkazuvchi strukturalarda razryad setkasi razryad sxemalari guruxi qayta ishlaydigan maydonlarga bo'linadi. O'zaro va guruxlar orasida o'tkazishning turli usullari ishlatiladi, bu narsa summatorning nomlanishida ko'rsatilgan bo'ladi. Masalan "parallel-parallel o'tkazgichli parallel summator" termini guruxlararo va summatorlararo parallel ko'chirishli summator ekanligini bildiradi. Kombinatsion summatorlar ham mavjud bo'lib, ular kombinatsion zanjirlar (murakkab) yordamida bog'langan bo'ladi. Bundan tashqari yig'uvchi summatorlar ham mavjud. Ularda mavjud ma'lumotlarga keyingi keluvchi ma'lumotlar qo'shilib boradi.

Summatorlar ishlash takti bo'yicha sinxron va asinxron summatorga tasniflanadi. Sinxron summatorlarda qo'shiluvchilarning qiymatlari va natijasi qanday bo'lishidan qat'iy nazar, hamma operatsiya uchun bir xil vaqt ajratilgan bo'ladi, asinxron summatorlarda esa amalni yakunlash belgisi ishlatiladi. Bunda qo'shishning o'rtacha vaqti kamayadi, bunday deyilishiga sabab amallarni bajarishdagi o'rtacha vaqt extimoliy maksimumdan ancha kamligi xisobga olinadi.

Shuningdek, sanoq sistemasini ishlatish yuzasidan ikkilik, ikkilik-o'nlik va boshqa turdagi summatorlar farqlanadi.





### 3.1. Komandalarni bajarilishi

Markaziy protsessor har bir komandani bir necha qadamda bajaradi;

- Xotiradan keyingi komandani chaqiradi va registrga yozadi
  - Keyingi komandani ko'rsatish uchun komanda hisoblagichi ko'rsatkichi xolatin o'zgartiradi (joriy komandani kodlagandan keyin yoki uni bajargandan keyin)
  - Chaqirilgan komanda tipini aniqlaydi
  - Agar komanda xotiradagi so'zni ishlatssa, bu so'z qayerda joylashganligini aniqlaydi
  - Zarur holatda so'zni markaziy protsessor registrga o'tkazadi
  - Komandani bajaradi
  - Keyingi komandani bajarish uchun yana 1-qadamga qaytadi
- Ishni bajarishning bunday ketma-ketligi (tanlash-kodlash-bajarish) barcha kompyuterlarning o'zgarimas asosidir.
- Kompyuter ishini tashkil qilinishining JAVA dastur sifatida ko'rinishi.

```
public class Interpreter
{
    static int PC; //PC so'zlarini xotira so'zlarini ko'rsatadi
    static int AC; //akumulyator, registri uchun qo'llaniladi
    static int instr; //instr komandani ko'rsatadi
    static int instr_type; //instr komandani tipini ko'rsatadi
    static int data_loc; //instr komandani ma'lumot manzili
    static int data; //instr komandani ma'lumot
    static boolean run_bit = true; //bit, komandani bajarishni davom ettirish uchun

    public static void interpret(int memory[], int starting_address)
    {
        //bu funksiya interpretatorni ishlatish uchun
        //instr komandani ko'rsatadi va o'zgaritish uchun
        //instr komandani tipini ko'rsatadi
        //instr komandani ma'lumot manzili
        //instr komandani ma'lumot
    }
}

private static int get_instr_type(int instr) {
    private static int find_data_loc(int instr, int type) {
    private static void execute(int instr_type, int data) {
}
```

Markaziy protsessor ishini bajaruvchi dasturning mavjud bo'lishi shuni bildiradi, dastur real protsessorida ishlanishi shart emas. Bu ishlarni (xotiradan o'qish-komanda tipini aniqlash-bajarish) ishini boshqa dastur ham bajarishi mumkin. Bunday dasturlar **interpretatorlar** deyiladi.

Interpretatorlarni ishlab chiqish kompyuter tizimlarini qurishda keng qo'llaniladi. Hisoblash mashinasini ishlab chiqaruvchilar yangi kompyuter uchun mashina tilini tanlashda quyidagilar: interpretator tilida ishlaydigan protsessor qurish, mashina tilida ishlovchi interpretator qurish. Agar interpretator yozishga kelishsa, shu interpretatorni ishlatuvchi qurilmani ham ishlab chiqishlari lozim. Shuningdek, ba'zi ishlar qurilmada, ba'zilar interpretatorda bajariladigan gibrid konstruksiyalar ham mavjud.

Interpretator komandalari kichik bo'laklarga bo'ladi. Shunday qilib interpretatorli mashinaning narxi interpretatsiyasiz protsessorli mashinaning narxidan arzon va qurilishi sodda bo'ladi. Agar kompyuter ko'p tarmoqli ko'plab murakkab komandalarni bajarishiga to'g'ri kelsa. Bunday tejamkorlik ko'zga tashlanadi.

Dastlabki kompyuterlar oddiy va kam sonli komandalar bilan ishlagan. Kompyuterlarning imkoniyatlari oshishi murakkablashgan komandalarning paydo bo'lishiga olib keldi. Murakkab komandalarni (masalan: qo'zg'aluvchi vergulli sonlar bilan ishlash, massiv elementlariga to'g'ridan-to'g'ri murojaat) ishlatish natijasida datsurlarning bajarilishi tezlashadi. Interpretatorlarni yaratish jarayonida ikkita komanda ketma-ket bajarilsa ularni bitta komanda qilib olinadi.

Murakkab komandalarni ishlatish parallel ishlatish natijasida unumdorlik ortadi. Bunda turli komandalar turli qurilmalarda bajarildi. Bunday tizim kompyuterlarning narxi oshib ketishiga olib keldi. Natijada ba'zi korxonalarda sekin ishlovchi arzon kompyuterlar, ba'zilarida esa tez ishlovchi qimmat kompyuterlar o'rnatildi. Bu esa o'zaro moslikni ta'minlashda muammolarni keltirib chiqardi. 1950-yillar oxirlarida arxitektura termini kiritildi. Bu termin bir xil amallarni bir xil tartibda bajaruvchi kompyuterlarning umumiy qurilishi (struktirasi) ni bildiradi. Bir xil arxitekturali kompyuterlarni narxini arzonlashtirish bo'yicha qator izlanishlar olib borildi va buning yechimi sifatida interpretatsiya taklif qilindi. 1951-yilda tashkil qilingan bu texnologiya oddiy va arzon, shu bilan birgalikda ko'plab amallarni bajara oladigan kompyuterlarni ishlab chiqish imkonini yaratdi. Izlanishlar natijasida IBM System/360 arxitekturasiga asos solindi. Interpretatsiyasiz texnik ta'minot faqat qimmatbaho kompyuterlarda ishlatiladigan bo'ldi. Bu kompyuterlarning alohida xususiyatlari quyidagilar edi:

- a) Noto'g'ri bajarilgan komandalarni belgilash, texnik ta'minotning yetishmovchiligini to'ldirish;

b) Kam xarajat evaziga yangi komandalarni qo'shish imkoniyati;  
 c) Strukturali tashkil qilinganlik (murakkab komandalarni teshirish va hujjatlashtirish);  
 1970-yillarda kompyuterlarning tarqalishi jadallashdi. Interpretatsion kompyuterlarga talab oshishi kompyuterlarni interpretatsiyali ishlab chiqishni keltirib chiqardi. Texnik ta'minot va interpretatorlar moslashuvchanligi arzon protsessorlar ishlab chiqarish jarayonini tezlashtirdi. Yerimo'tkazgichli texnologiyalar tezkor rivojlandi, interpretatorli kompyuterlar ishlab chiqarish keskin suratda oshdi.

1970-yillarning oxirida interpretatorlar hamma modellarda keng qo'llanildi. Ishlab chiqaruvchilar interpretatorlar uchun yangi murakkablashgan komandalar ishlab chiqardilar, ayniqsa operandlarni qo'llash bo'yicha.

Bu tendensiya VIX kompyuterini ishlab chiqish bilan o'zining cho'qqisiga yetdi. VIXda 300 dan ortiq komanda va operandlarni aniqlashning 200 dan ortiq usullari mavjud edi. VIXni ishlab chiqishda komandalar ko'pligiga e'tibor qaratsih bilan uning unumdorligiga e'tbor qaratilmay qolgan. Natijada ikkinchi darajali bo'lgan, juda kam ishlatiladigan ko'plab komandalar paydo bo'ldi.

### RISC va CISC arxitekturalari

1970-yillar oxiriga kelib dasturlash tillari rivojlanishi yangi bosqichga chiqdi. Endi dasturlash tillari tabiiy tilga yaqinlashtirilgan holga keldi. Buning natijasida esa mashina tili va dasturlash tili orasidagi tafovutni kamaytirish muammosi kelib chiqdi. 1980 yilda interpretatsiyasiz protsessor ishlab chiqildi. Bu RISC texnologiya deb nomlandi. Bu protsessorlar o'z davridagi boshqa protsessorlardan farq qiladi. Ular joriy kompyuterlar bilan nomuvofiq bo'lgan, ishlab chiqaruvchilar unga qo'shimchalar kiritish imkoniga ega bo'lgan. Asosiy e'tibor tez bajariladigan sodda komandalarga qaratilgan, keyinchalik esa tez bajariluvchi zuzn komandalarga bilan ishlash, shuningdek bitta komanda qancha vaqtda bajariladi deb e'tibor qaratala boshlandi.

### 3.2. Zamonaviy kompyuterlar ishlab chiqish tamoyillari

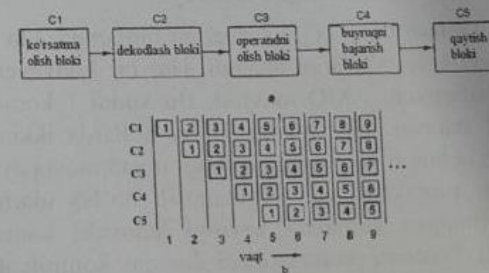
RISC texnologiyali kompyuterlar keng tarqalishi bilan ishlab chiqarishning o'zgarish xususiyatlarini e'tiborga olish lozim. Agar

texnologiyafda keskin o'zgarish sodir bo'lsa (masalan xotira sikli 10 baravar kichiklashsa) barcha shartlar o'zgaradi. Shu sababli ishlab chiqaruvchilar ushbu ko'rsatkichni nazarda tutishlari lozim bo'ladi.

RISCda barcha oddiy komandalar texnik ta'minotda bajariladi. Ularni interpretatsiya qilish shart emas. Bu esa yuqori unumdorlikka olib keladi. 500 MIPS degan belgi protsessorni bir sekunda 500mln komandani bajarishini bildiradi. (MIPS-Million of Instructions Per Second). Komandalar oson dekodlanishi lozim. 1 sekunda bajariladigan amallar soni dekodlanadigan amallar soniga bog'liq. MP ko'p registrlil bo'lishi maqsadga muvofiq. Ma'lumki, xotiraga murojaat ko'p vaqt talab qiladi. Xotiradan o'qilgan so'zni regitrd, toki kerak bo'lmay qolguncha, saqlab turish mumkin. Chunki bir marta o'qilgan so'zni xotiraga qaytarish, keyin zarur bo'lganda yana xotiradan o'qish vaqtni cho'zilishiga olib keladi.

Komandalar parallelligi. Mashinaning tezligi oshirishning usullaridan yana biri bir vaqtda bir necha komandalarni bajarishdir. Parallellikning ikki shakli mavjud: komandalardagi parallellik va protsessorlardagi parallellik. 1-xolatda bir vaqtda bir necha komanda bir vaqtda bajarilsa, 2-xolatda bir masala ustida bir necha protsessor ishlaydi.

Konveyerlar. Ma'lumki, xotiradan so'zni o'qish ko'p vaqt talab qiladi. Natijada mashina tezligi pasayadi. Buni hal qilishning yana bir yo'li ba'zi komandalarni oldindan chaqirib qo'yish amaliyotidir. Ko'p ishlatiladigan komandalar oldindan tanlash buferiga yuklab qo'yiladi. Zarur vaqtda unii chaqirish kam vaqtda bajariladi. Bu g'oya 1959 yilda qo'llanilgan. Bunday konveyerlarni qo'llash natijasida komandalar bir necha qismlarga bo'linib, har bir qism alohida qurilmalarda parallel bajariladi.



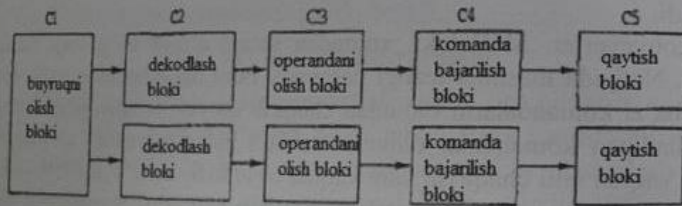
3.2.-rasm. 5 bosqichli koveyer. (a); o'tgan davrlar soniga har bir bosqichning holati (b); p sikl ko'rsatilgan

3.2 (a) rasmda 5 blokli konveyer keltirilgan. C1 bloki xotiradan komandani chaqiradi va buferga joylaydi. Komanda buferda chaqirilguncha saqlanib turadi. C2 blokida komanda dekodlanadi, uning tipi va operandlar tipi aniqlanadi. C3 blokida operandlarning adresi aniqlanadi va chaqiriladi. C4 blokida komanda bajariladi (1-slaydda ko'rsatilganidek).

3.2 (b) rasmda vaqt konveyeri qanday ishlashi keltirilgan. 1-siklda C1 bloki 1-komandani xotiradan chaqiradi. 2-siklda C2 bloki 1-komandani dekodlaydi, C1 bloki esa 2-komandani xotiradan chaqiradi. 3-siklda C3 bloki 1-komandani bajarish uchun operandlarni chaqiradi, C2 bloki 2-komandani dekodlaydi < C1 bloki 3-komandani chaqiradi. 4-siklda C4 bloki 1-komandani bajaradi, C3 bloki 2-komandani bajarish uchun operandlarni chaqiradi, C2 bloki 3-komandani dekodlaydi, C1 bloki 4-komandani chaqiradi. 5-siklda C5 bloki 1-komandaning hisoblash natijasini registrga yozadi, qolgan bloklar yuqoridagi tartibda ishini davom ettiradi.

### 3.3. Superskalyar arxitektura

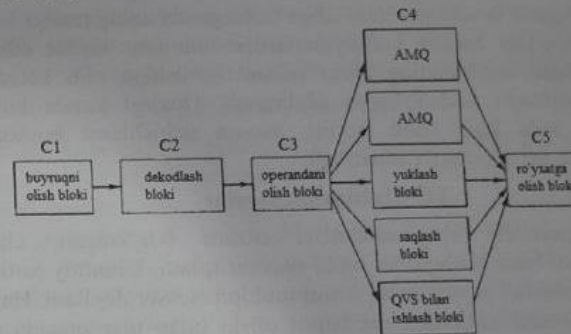
Konveyerlar 2 langan bo'lishi yaxshi.



3.3-rasm. Buyruqlarni chaqirishda umumiy bo'limli 5 bosqichli ikkilangan konveyer

Komandani tanlash bloki (C1) xotiradan birdaniga 2 ta komandani o'qiydi va alohida konveyerlarga uzatadi. Har bir konveyerda parallel operatsiyalarni bajaruvchi AMQ mavjud. Bu xuddi 1 konveyerli kabi ishlaydi, faqat 2 baravar tez ishlaydi. Ba'zi hollarda ikkita komanda parallel bajarilish uchun o'zaro mutanosibligi muammosi paydo bo'ladi. Agar komandalar murakkab va nomutanosib bo'lsa ularning bittasi bajariladi. Bajarilmagan komanda keyingi keluvchi komanda bilan juftlik hosil qiladi. Pentium protsessorlari shunday kompilyatorlar bilan ta'minlangan. Bir xil takt chastotali 486 va Pentium protsessorlari taqqoslanganda Pentium deyarli 2 baravar unumdor ekanligi aniqlangan. Bu esa ikkinchi konveyer hisobiga bo'lgan. 4 ta konveyerli tizimga

o'tish ham mumkin, lekin bunda texnik qurilmaning (MP ning) gabaritlari oshib ketadi. Buning o'rniga ko'p blokli konveyerdan foydalanish tavsiya etiladi.

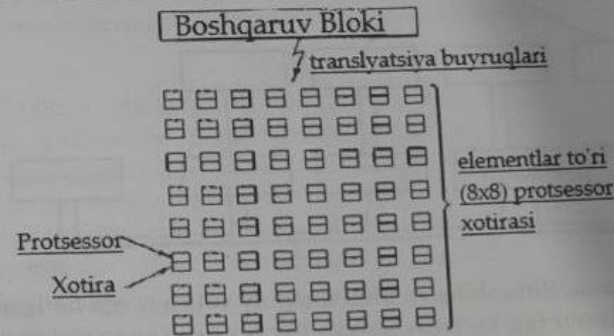


3.4-rasm. Beshta funksional blokli superskalyar protsessor

PentiumII shu strukturaga ega. Bu yondashuv superskalyar arxitektura deb nomlanadi. Bu yerda C3 blokida komandani shakllantirishi C4 da qayta ishlashida tez bajariladi. Shu sababli C4 da ko'p AMQlardan foydalanish zarur.

### 3.4. Vektorli kompyuterlar

Massiv-parallel protsessorlar bir xil bo'lgan juda ko'plab protsessorlardan tashkil topadi. Uning sxemasi



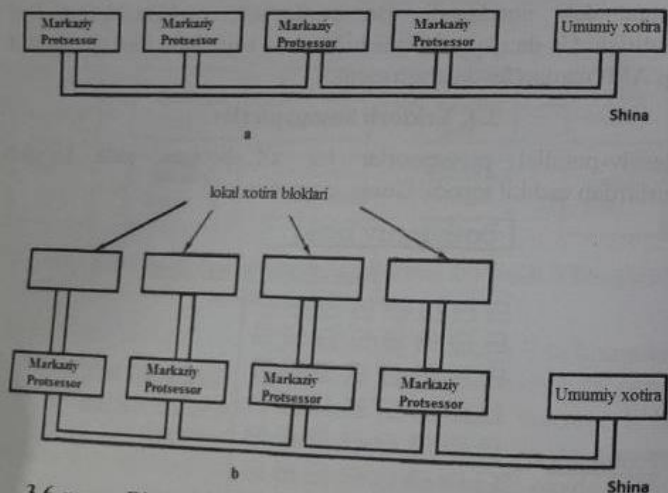
3.5-rasm. Massiv-parallel protsessorlarga misol.

Dastlabki variantda 4 ta sektorli, har birida 8x8 prots/xotira bo'lgan sxemali mashina i/ch rejalangan. Har sektor alohida nazorat blokiga

ega. Blok yuborgan komanda hamma sektorda bir vaqtda bajarilgan, bunda har bir prots.o'zining xotirasidan foydalangan. Bu sxemaning narxi baland bo'lgaligi sababli 50 mln amal bajaradigan 1 dona sektor i/chiqildi. Agar 4 ta sektorni ham ishga tushirganda uning tezligi 1 mlrd bo'lib, dunyodagi barcha kompyuterlardan unumdor bo'lar edi. Bu texnik jihatdan mashinaning narxi baland bo'lishiga olib keladi va ishlab chiqaruvchi undan foyda ololmaydi. Hozirgi kunda bunday mashinalar juda kam sonli bo'lib, asosan uchishlarni boshqarish markazlarida, aeroportlarda ishlatiladi.

### 3.5. Multiprotsessorlar

Massiv-parallel prots.elementlari o'zaro bog'langan, chunki ularning ishini bitta boshqarish bloki nazorat qiladi. Umumiy xotiraga ega bo'lgan parallel prosessorlar tizimi multiprotsessor deyiladi. Har bir prosessor bevosita xotiradan ma'lumot olishi bilan ular orasida nizo kelib chiqmasligi uchun dasturiy ta'minot bo'lishi kerak. Bunga turli yondashuvlar mavjud. Eng oddiyi bir necha prosessorlar va xotirani birlashtiruvchi shina.



3.6-rasm. Bitta shina va bitta umumiy xotiraga ega bo'lgan multiprotsessor (a); ko'p protsessor, har bir protsessor o'zining lokal xotira blokiga ega bo'lgan multiprotsessor (b)  
Bundan (a qismida) ko'rinib turibdiki, bir necha prosessor bitta shina orqali xotiraga murojaat qilishi bilan nizo kelib chiqadi. Bu

muammoni yechish uchun turli modellar taklif qilingan. Ulardan biri (b) rasmda. Bunda har bir prosessor boshqalari kira olmaydigan alohida xotiraga ega. Unda ma'lumotlar va dasturlar joylashadi. Xotiraga murojaat qilinganda asosiy shina ishlatilmaydi va uning yuklanmasi kamayadi. Multi prosessorlaning boshqa parallel prosessorlardan ustunligi alohida ajratilgan xotira bilan ishlash qulayligidadir.

### 3.6. Multikompyuterlar

64 dan kam bo'lgan prosessorlar ishtirokida multi prosessor hosil qilish qiyinchilik tug'dirmaydigan ish. Ko'p sonli prosessorlardan multi prosessor qurish ancha murakkabliklarni keltiradi. Asosiy qiyinchilik hamma prosessorlarni xotira bilan bog'lashda ko'zga tashlanadi. Bunday muammoni hal qilish uchun o'zaro bog'langan ko'p sonli kompyuterlardan foydalanish g'oyasi ishlab chiqildi. Buning yutug'i har bir kompyuter o'zining lokal xotirasiga ega ekanligidadir. Bunday tizimlar multikompyuterlar deb nomlandi. Multikompyuter protsessorlari bir-biriga ma'lumot uzatadi (el.pochtaga o'xshash). Har bir kompyuterni hamma kompyuter bilan bo'lash shart emas, asosan 2D, 3D topologiyalari ishlatiladi. Uzatilgan ma'lumot tegishli adresga yetib borishi uchun bir necha kompyuterlardan o'tib borishi kerak. Shunday bo'lsa ham uzatish vaqti juda qisqa, bir necha millisekund. Zamonaviy multikompyuterlar 10000 tagacha prosessorlarga ega. Multiprotsessorlarni dasturlash oson, multikompyuterarni esa loyihalash osonligidan, gibrid tizimlarni yaratish g'oyasi paydo bo'ldi. Bunday kompyuterlar taqsimlangan xotira ko'rinishini bersa ham, haqiqatda konstruksiyalanmaydi va katta harajatlarni talab qilmaydi.

### 3.7. Asosiy xotira

Xotira – dasturlar va ma'lumotlar saqlanadigan kompyuterning texnik qurilmasidir. "Saqlovchi qurilma" termini ham ishlatish to'g'ri bo'ladi. Xotirasiz hozirgi kompyuterlarni tasavvur qilish mumkin emas. Xotiraning asosiy birligi bit deyiluvchi 1 yoki 0 ga teng ikkilik razryaddir. Bu eng xotiraning kichik birligi. Ba'zilarning fikricha kompyuterda binar arifmetikaning ishlatilishi sababi tezkor bo'lganligidir. Ikkilik axborotni saqlashda qandaydir fizik kattalikning turli qiymatlari farqiga asoslangan (0 yoki 1, bor yoki yo'q). Ikkilik tizim faqatgina ikkita qiymatga ega bo'lgani sababli axborotlarni kodlashning eng ishonchli tizimi deb olingan.

Ba'zi kompyuterlar ikkilik-o'nlik sanoq tizimini ishlatadi. Bitta o'nlik razryadni saqlash uchun 4 bit ishlatiladi. Bu 4 bit 10 ta turli qiymatlarni saqlash uchun 16 kombinatsiyaga ega bo'ladi. Bunda qolgan 6 ta kombinatsiya ishlatilmaydi. 1944 (10lik) soni tasvirlanishi:

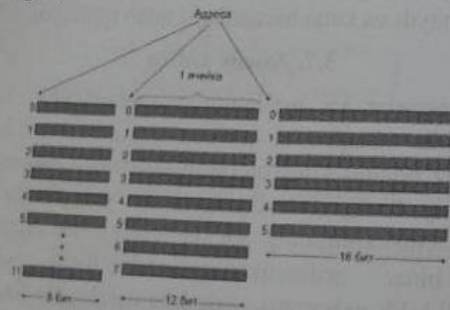
O'nlik 0001 1001 0100 0100  
Ikkilik 0000 0111 1001 1000

Ikkilik-o'nlik formatda 16 bit axborot 0 dan 9999 gacha sonlarni ifodalashi mumkin, yoki 10.000 ta kombinatsiya, ikkilik formatda esa 65536 ta kombinatsiya. Shu sababli ham ikkilik tizim samarali deyiladi.

Tasavvur qilaylik 0 dan 9 gacha razryad saqlovchi qurilma mavjud. Unda 10 ta interval bo'lib, 10ta darajaga ajratilgan. Shunday qurilmaning 4 tasi 0 dan 9999 gacha, yoki 10.000 ta kombinatsiyani beradi. (Birinchi 10 gacha, ikkinchi 100 gacha, uchinchi 1000 gacha, to'rtinchi 9999 gacha). Shu qurilmada ikkilik sonlarni tasvirlasak bor-yo'g'i 16 ta kombinatsiya mavjud bo'ladi. Bu qurilmada faqat 10 lik tizimda ishlash maqbul ekanligi shundan ko'rinib turibdi.

### 3.8. Xotira adreslari

Xotira yacheykalardan iborat. Har yacheykada ma'lumot joylashadi. Har bir yacheykaning adresi bor. Shu adres bo'yicha dastur tegishli yacheykaga murojaat qiladi. Agar n ta yacheyka bo'lsa, ularning adreslari 0 dan n-1 gacha adreslanadi. Agar yacheyka k bitli ma'lumotga ega bo'lsa,  $2^k$  turli kombinatsiyadagi ma'lumotni aks ettirishi mumkin. Quyidagi rasmda 96 bitli xotirani tashkil qilishning 3 xil usuli keltirilgan



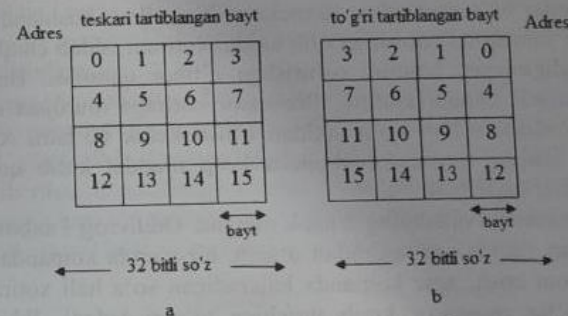
3.7-rasm. 96 bitli xotirani tashkil qilishning uchta usuli

Ikkilik sistema ishlatiladigan kompyuterlarda (8 lik va 16 likda n) xotira adreslari ham ikkilikda ifodalanadi. Agar adres m bitli bo'lsa  $2^m$  ta yacheykani adreslash mumkin. Masalan slayddagi (a

qismida) xotira adresi kamida 4 bitli bo'lishi lozim. (b) va (v) rasmda ko'rsatilgandek tashkil qilingan xotirada esa 3 bitli adreslash yetrali. Adresdagi bitlar soni xotiraning maksimal adreslanadigan sonini bildiradi va yacheykadagi bitlar soniga bog'liq emas.

Yacheyka murojaat qilish mumkin bo'lgan eng kichik xotira birligi. Zamonaviy kompyuterlarda asosan 8 bitli yacheykalar (1 bayt) ishlatiladi. Bir bayt bir so'z deyiladi. 32 bitli kompyuter 8 ta 4 baytli so'zga ega bo'ladi. 32 bitli mashina 32 bitli registrlardan va 32 bitli komandalardan tashkil topadi, shuningdek, 64 bitli mashina ham shuningdek.

Baytlarni taqsimlash. So'zdagi baytlar o'ngga yoki chapga tomon raqamlanishi mumkin.



3.8-rasm. Bayt chapdan o'ngga raqamlangan xotira (a), Bayt o'ngdan chapga raqamlangan xotira (b)

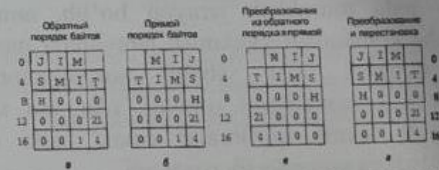


Рис. 2. 10. Записанные справа для машины с нумерацией байтов слева направо (а); та же запись для машины с нумерацией байтов справа налево (б); результат преобразования записи из нумерации слева направо в нумерацию справа налево (в); результат перестановки байтов в произвольном случае (г)

Baytlarni orqadan tartiblash	To'g'ri tartibli baytlar	Teskari tartibda oldinga siljish	O'qibish va to'xtatish
0 J I M	M I J	M I J	J I M
4 S M I T	T I M S	T I M S	S M I T
8 H 0 0 0	0 0 0 H	0 0 0 H	H 0 0 0
12 0 0 0 21	0 0 0 21	21 0 0 0	0 0 0 21
16 0 0 1 4	0 0 1 4	4 1 0 0	0 0 1 4

### 3.9. Kesh xotira

Processor hamisha xotiraga nisbata tez ishlagan. Ular parallel ravishda mukammallashgan, shu sababli tezlikdagi nomutanosiblik saqlanib qolgan. Mikroshemadagi tranzistorlarning soni oshib borishi bilan konveyerlar va superskalyar arxitekturada qo'llash boshlandi. Bu processorlarni yanada tezlashishiga olib keldi. Xotirani ishlab chiqishda esa uning tezliginimas, hajmini oshirishga e'tibor qaratildi. Bu esa muammoni yanada chuqurlashtirdi. Processor xotiraga murojaat qilib, kerakli so'zni olishi uchun ko'p sikldan o'tish kerak bo'ladi. Xotira qancha sekin ishlasa, processor shuncha ko'p muddat kutib qoladi, shuncha ko'p sikl bajariladi.

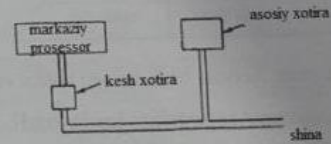
Bu muammoni hal qilishning 2 usuli mavjud. Oddiyrog'i axborotga ehtiyoj tug'ilgan vaqtda uni xotiradan o'qish, bir vaqtda komandalarni bajarishda davom etish, agar komanda bajaradigan so'z hali xotiradan o'qilmagan bo'lsa processor kutib turishiga to'g'ri keladi. Ikkinchi yechim kompilyatorlarga so'z xotiradan o'qilmasdan oldin komandani bajarmasligini buyurish. Shu vaqtda mashina ishini ham to'xtatmasligi talab etiladi. Bu juda murakkab jarayon bo'lib, amalga oshirish mushkul. Ko'pincha komandani yuklanish jarayonida mashina boshqa ishlarni bajara olmaydi, shu sababli kompilyator hech ish bajarmaydigan bo'sh komandalarni o'qiydi, ular esa xotirani band qiladi. Bu vaziyatda texnik qurilmalar emas, balki dasturlarning to'xtalishi sodir bo'ladi va unumdorlik pasayadi.

Aslida bu texnologik emas, balki iqtisodiy muammodir. Processor kabi tez ishlaydigan xotira yaratish mumkin, agar uni bevosita processorga joylashtirilsa, chunki shinadan ma'lumotlar sekin oqib o'tadi. Bu esa protsessorning hajmi kattalashishiga va narxi oshishiga olib keladi. Tanlash lozim: xotira katta tezikli va kichik hajmli bo'lsinmi yoki kichik tezlikli katta hajmli bo'lsinmi. Odatda xotiraning katta bo'lish tamoyili tanlanadi.

Kichik va tezkor xotirani katta va sekin xotira bilan birlashtirish texnologiyalari mavjud. Bu esa katta tezlik va katta hajmni nisbatan arzon narxda shakllantirish imkonini beradi. Bunday xotira **kesh-xotira** (fr.kesh-yashirish) deyiladi. Kesh xotirada eng ko'p ishlatiladigan so'zlar joylashtiriladi. Processor kerakli so'zni dastlab keshdan qidiradi, agar unda bo'lmasa keyin asosiy xotiraga o'tadi. Lekin, ko'pincha u so'z keshda mavjud bo'ladi.

Ma'lumki, dasturlar xotiraga tasodifiy murojaat qilmaydi. Agar A adresga murojaat qilinsa, keyingi qadamda A+1 yoki A-1 adresga murojaat qilish ehtimoli juda yuqori. Chunki dasturlar ketma+ket yozilgan, xotirada ketma-ket joylashgan va shunday bajariladi (shartli va shartsiz o'tish komandalari bundan mustasno). Bundan tashqari dasturning ba'zi qismlari takroriy ravishda bajariladi. (sikllar) shuningdek matritsalar bilan ishlaganda ham qayta murojaatlar soni juda ko'p takrorlanadi.

Mana shu xotiraning bir qismiga ketma-ket ko'p bor murojaat qilish **lokallik prinsipi** deyiladi. Bu prinsip barcha kesh xotiralarning asosini tashkil qiladi. Prinsip quyidagicha: agar dastur xotiradan biror so'zni chaqirib olsa, u atrofidagi so'zlar bilan birgalikda keshga yuklanadi. Dasturning keyingi murojaatida keshdagi ma'lumotlarga murojaat qilish ehtimoli yuqori bo'ladi va kerakli so'z keshdan topiladi.



### 3.10-rasm. Kesh-xotira mantiqiy ravishda protsessor va asosiy xotira o'rtasida joylashgan bo'lishi kerak. Aslida uchta mumkin bo'lgan kesh joylashuvi mavjud

Kesh xotirani ishlab chiqish yuqori unumdorlikli protsessor uchun ahamiyati katta. Kesh xotira hajmi qancha katta bo'lsa, unumdorlik shuncha yuqori.

#### 3.10. Xotira modullarini yig'ish va ularning tiplari

Dastlabki mashinalarda xotira qurilmalari alohida ishlanib, plataga alohida o'rnatilgan. Platada qo'shimcha xotira uchun qo'shimcha razyomlar qoldirilgan. Hozirda esa (8 yoki 16) mikroshemalar guruhi

bitta plataga birlashtirilib, bir blok sifatida sotilmoqda. Bu blok SIMM deyiladi (Single Inline Memory Module-bir tomonlama chiqishli xotira moduli) yoki DIMM (Dual Inline Memory Module-ikki tomonlama chiqishli xotira moduli). Oddiy SIMMda 4 Mbaytli 8 ta mikrosxema bor. U 32 mb axborotni saqlaydi. 4 modul ketma-ket o'rnatilsa  $4 \cdot 32 = 128$  MB bo'ladi. 64, 128 MB modullar ham mavjud

Dastlabki SIMMlarda 30 ta kontakt bo'lib, bir vaqtda 8 bit axborot uzata olgan. Boshqa kontanctlar adreslash va nazorat uchun qoldirilgan. Keyingi modullar 72 kontaktli bo'lib, 32 bitni uzata olgan.

DIMM modulining har tarafida 84 tadan, jami 168 ta kontakt mavjud. Bir vaqtda 64 bit uzatadi. DIMMning hajmi odatda 64 MB va yuqori bo'ladi.

### Xotiraning iyerarxik strukturasi

Ko'p hajmli ma'lumotlarni saqlash muammosi ierarxik stuktura orqali hal qilinadi.



3.11-rasm. Xotirani besh darajali tashkil etish

Eng yuqorida protsessor registrlari joylashadi. Unga murojaat qisqa muddatda bajariladi. Keyingi bosqichlarda turgan xotiralarga murojaat mos ravishda cho'qqidan uzoqlashganligiga qarab bo'ladi.

Pastga tushib borgan sari 3 ta qiymat oshib boradi. 1-chaqiruv vaqti. 2-xotira hajmi. Registr ko'pi bilan 128 bayt olsa, kesh bir necha mb, asosiy xotira bir necha gb, va h.k. 3-narxi. Bu yerda 1 dollarga yetib keladigan bitlar soni aytilmoqda.

### Nazorat uchun savollar:

1. Kompyuter tizimining tuzilish turlari nimalardan iborat?
2. Kompyuter arxitekturasi dagandanima tushunasiz?
3. Kompyuter arxitekturasi va uni tashkil qilishda qo'llaniladigan standartlar nimalardan iborat?
4. Kompyuterlarning tashkiliy qismlari?

## 4. Buyruqlar tizimi arxitekturasi tasniflash. Operandlarning turlari va formatlari. Buyruqlar turlari va formatlari

Buyruqlar tizimi arxitekturasi deganda kompilyator uzatayotgan ma'lumotlarning darajasi tushuniladi. Buyruq arxitektura darajasini ishlab chiqishda kompilyator yaratuvchi mashinada qanday xotira ishlatilayotganini, qanday registrlar, ma'lumotlar tipi, qanday komandalar ishlatilishi kabilarni bilishi talab etiladi. Mana shu hamma axborotni umumlashgan xolda buyruqlar tizimi arxitekturasi deyish mumkin.

Bundan kelib chiqib, mikroarxitektura dasturlanadimi yoki yo'q, mashina konveyerlanganmi yoki yo'q, superskalyarmi yoki yo'q va h.k kabi muammolar kompilyator ishlab chiquvchi uchun ahamiyatsizdir.

### 4.1. Buyruqlar tizimi arxitekturasi. RISC va CISC protsessorlari

Buyruqlar tizimi arxitekturasi (ISA – Instruction Set Architecture) deganda buyruqlar tizimining tarkibi va imkoniyatlari, buyruqlar tizimining umumiy ko'rinishi va dasturchi nuqtai nazaridan protsessorning tegishli mikroarxitekturasi tushuniladi. Ko'p jihatdan, komandalar tizimi arxitekturasi umuman kompyuter arxitekturasi ta'rifini "...dasturchi nuqtai nazaridan kompyuterning mavhum g'oyasi" sifatida belgilaydi.

Tarixiy jihatdan, XX asrning 70-yillarida paydo bo'lgan birinchi mikroprotsessorlar nisbatan oddiy komandalar tizimiga ega bo'lib, bu integral mikrosxemalarning kichik imkoniyatlari bilan izohlanadi. Integral sxemalarning integratsiya darajasi oshib borgani sayin, markaziy protsessor ishlab chiquvchilar buyruqlar tizimini kengaytirishga va buyruqlarni ko'proq funksional, "semantik yuklangan" qilishga harakat qilishdi. Bu, xususan, ikkita jihat bilan bog'liq edi - birinchidan, dasturlarni joylashtirish uchun xotirani tejash, ma'lumotlar uchun ko'proq xotira qoldirish va boshqa talablari, ikkinchidan, protsessor chipi ichidagi murakkab ko'rsatmalarni dasturiy ta'minotni bajarishdan ko'ra tezroq amalga oshirish qobiliyati.

Natijada katta komandalar to'plamiga ega protsessorlar paydo bo'ldi va bu komandalar ham ko'pincha juda murakkab tuzilishga ega bo'lgan. Keyinchalik, bu mikroprotsessorlar CISC deb nomlandi (Complete Instruction Set Computer - to'liq ko'rsatmalar to'plamiga ega kompyuter) yoki Kompleks ISC - murakkab komandalar to'plami. CISC protsessorlarining tipik misollari sifatida Intel firmasining x86



protessorlari va uning raqobatchilari (shuningdek, Motorola 68K va boshqalar) ni ko'rsatib o'tish mumkin.

Qayd etilgan afzalliklar bilan bir qatorda, CISC protessorlari ham bir qator kamchiliklarga ega edi, xususan, komandalar bajarilish vaqti (turli xil sikllar soni), noto'g'ri yo'naltirilganligi, murakkab (va uzoq) dekodlash va bajarilishini talab qilishda tengsiz bo'lib chiqdi.

Unumdorlikni oshirish uchun kristallarning muntazamligi va murakkabligida o'z aksini topgan qattiq boshqaruv mantig'i (жесткую логику управления) dan foydalanish boshlandi (doimiy bo'lmagan kristallar ishlab chiqarishda texnologik jihatdan kam rivojlangan).

Bundan tashqari, tadqiqotlar shuni ko'rsatdiki, kompilyator ishlab chiqaruvchilari va oddiy dasturchilar ko'p sonli murakkab komandalardan foydalanmaydilar balki, qisqa buyruqlar ketma-ketligini ishlatishni afzal ko'rishadi.

Ishlab chiquvchilar biroz orqaga qaytish - oddiy va qisqa komandalarga ega bo'lgan sodda va texnologik jihatdan ilg'or protessor konseptsiyasiga yaqinlashdilar. 1970-yillarning oxiridan 1980-yillarning o'rtalariga qadar Stenford universiteti va Berkli universitetida (Kaliforniya) bunday protessorlarning loyihalari - MIPS va RISC paydo bo'ldi.

RISC (Reduced Instruction Set Computer) arxitekturasi, xususan, murakkab va ko'p funksiyali buyruqlardan voz kechish, ularning sonini kamaytirish, shuningdek, barcha ma'lumotlarni asosan xotiraga minimal sonli murojaat bilan protessor chipida qayta ishlashga e'tibor qaratish tamoyillariga asoslanadi.

RISC arxitekturasining asosiy xususiyatlari:

1. Komandalar sonini qisqarishi (30-40 tagacha).
2. Buyruqlar formatlarini soddalashtirish va unifikatsiya qilish.
3. Buyruqlar tizimida qisqa buyruqlar ustunlik qilishi (masalan, komandalar tizimida ko'paytirish amali ko'pincha bo'lmaydi).
4. Xotira-xotira tipidagi buyruqlarda foydalanishni rad etish (masalan, x86 da MOVSB).
5. Xotira bilan ishlash registrlarni yuklash va saqlash bilan bog'liq (shuning uchun RISC arxitekturasining yana bir nomi Load-Store Architecture - "yukni saqlash" tipidagi arxitektura).
6. Ko'pincha 3-adresli buyruqlar amalga oshiriladi, masalan:  $add\ r1, r2, r3 // r2\ va\ r3\ ni\ yig'indisini\ hisoblash\ va\ natijani\ r1\ ga\ joylashtirish$

7. Qattiq boshqaruv mantig'iga (жесткой логике управления) ustunlik beriladi.

RISC arxitekturasining afzalliklari:

1. Konveyerlash, superscalyar va boshqa turdagi parallel ishlov berish osonlashadi, yuklanishni rejalashtirish, oldindan tanlash, qayta tartiblash va boshqalar osonlashtiriladi.

2. Chip maydoni yanada samarali ishlatiladi (ko'proq xotira - RON, kesh uchun).

3. Dekodlash va buyruqlarni bajarish tezroq bajariladi - mos ravishda, yuqori takt chastotasi ta'minlanadi.

RISC protessorlari oilalariga misol sifatida DEC Alpha, SGI MIPS, Sun SPARC va boshqalarni keltirishi mumkin.

Ko'pgina zamonaviy superscalyar va VLIW protessorlari (shu jumladan Intel) RISC arxitekturasiga ega yoki RISC ga o'xshash printsiplarni amalga oshiradi yoki CISC ko'rsatmalarini qo'llab-quvvatlaydi, lekin konveyerlarni yuklashni va boshqa vazifalarni hal qilishni osonlashtirish uchun ularni RISC-ga o'xshash ko'rsatmalarga translyatsiya qiladi.

#### 4.2. Operandlar tipi va formati

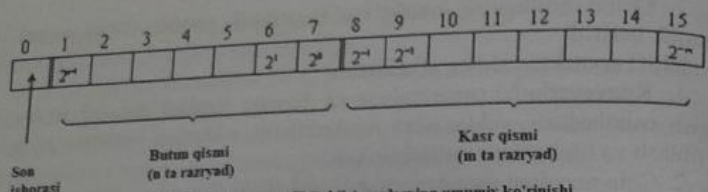
Mashina komandalari operandlar deb ataluvchi ma'lumotlar bilan ishlaydi. Operandlarning asosiy tiplari

- Adreslar
- Sonlar
- Simvollar
- Mantiqiy ma'lumotlar

Hisoblash mashinalari bulardan tashqari murakkabroq bo'lgan axborot birlilari bilan ishlaydi mas: grafik tasvirlar, audio, video, animatsiya. Bular esa bazaviy tipdagi malumotlardan kelib chiqadi va tashqi xotirada mavjud bo'ladi.

Raqamli ma'lumot. HM da sonlar ikki xil tasvirlanadi: qo'zg'almas va qo'zg'aluvchi vergulli. Bu formatlarni **shuningdek tabiiy va yarimlogarifmik** deb aytiladi.

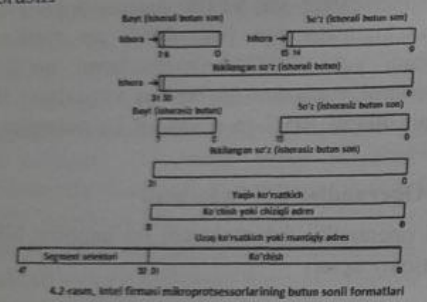
Qo'zg'almas vergulli (tabiiy) sonlar. Bunday shakldagi sonlarda vergul son razryadlariga nisbatan qat'iy o'rinda joylashadi.



4.1-rasm. Qo'zg'almas vergulli (tabiiy) sonlarning umumiy ko'rinishi

Odatda vergul katta razryad oldidan yoki kichik razryaddan keyin joylashadi deb ko'zda tutiladi. 1-holatda moduli 1 dan kichik sonlar tasvirlansa, 2-holatda faqat butun sonlar tasvirlanadi.

Butun sonlarni ifodalashning ikki varianti ishlatiladi: ishorali va ishorasiz.



4.2-rasm. Intel formati mikroprotsessorlarining butun sonli formatlari

Shuningdek butun sonlar adreslar bilan ishlashda ham ishlatiladi.

### 4.3. O'nlik sonlar

O'nlik sonlarni tasvirlashning asosi qilib har bir o'nlik raqamni ekvivalent ikkilik songa (tetradaga) almashtirish olinadi (ikkili-o'nli tizim). O'nlik sonlarni tasvirlashning ikki formatda ishlatiladi (barcha sonlar butun deb qaraladi)

Har ikkala formatda har bir o'nlik raqam ikkilik tetradada ifodalanadi (ikkili-o'nli kodda tasvirlanadi). Qolgan 4 razryadli ikkilik kombinatsiyalar + va - ni kodlash uchun ishlatiladi. Masalan IBM 360/390 tipidagi HM larda + uchun 1100 (C16), - uchun 1101 (D16) olingan.

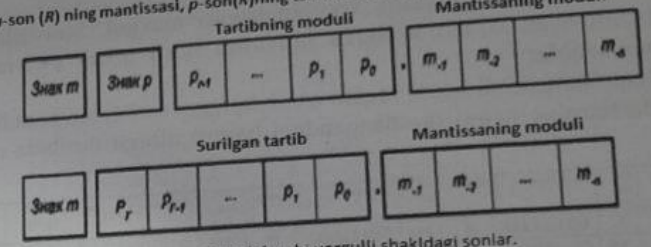
Ikkili-o'nli tizimning juda ko'plab variantlari mavjud ( $2,9 \cdot 10^{10}$  mln). Eng ko'p ishlatiladigan kodlash tizimlari bu 8421 va 8421+3 kodlash tizimlaridir.

Qo'zg'aluvchi vergulli shakldagi sonlar.

Bunday sonlarni normal yoki yarimlogarifmik deb ham ataladi. Umumiy ko'rinishi quyidagi rasmda keltirilgan

$$R = \pm m \cdot q^{\pm p}$$

bunda  $m$ -son ( $R$ ) ning mantissasi,  $p$ -son ( $R$ ) ning tartibi,  $q$ -sanoq sistemasining asosi

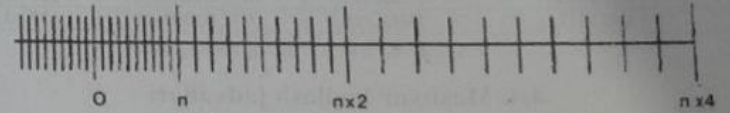


4.3-rasm. Qo'zg'aluvchi vergulli shakldagi sonlar.

Qo'zg'aluvchi vergulli sonlar diapazoni va aniqligi tartib va mantissaga ajratilgan razryadlar soni, sanoq tizimi asosiga bog'liq. Masalan IBMning universal hisoblash mashinalarida 16 lik tizim ishlatiladi.

Qo'zg'aluvchi vergulli sonlarning mantissasi 1 dan kichik bo'lishi va birinchi raqami 0 dan farqli bo'lishi kerak. Bunday mantissa **normallashgan** deyiladi. Agar qo'zg'aluvchi vergulli sonning asosi 2 bo'lsa, aniqlikni oshirish usuli sifatida **yashirin bir usuli** dan foydalaniladi. Normal mantissada katta registr hamisha 1 ga teng, shu sababi **yashirin bir usuli** da bu raqam yozilmaydi, balki bor deb ko'zda tutiladi.

**To'lib ketish** – hisoblash natijasining maksimal tartibdan oshib ketishi. Qiymat yo'qotish – hisoblash natijasi haddan tashqari kichik bo'lishi. Qo'zg'aluvchi vergulli sonlar absissa o'qida notekis joylashgan bo'ladi. Markazdan uzoqlashgan sari tarqoqlanish kuchayib boradi.



4.4-rasm. Qo'zg'aluvchi vergulli sonlar absissa o'qida notekis joylashishi

### Noaniqlik. (NaN)

NaN – Not a Number. Amallar bajarilishi vaqtida sodir bo'lgan xatoning qiymati. NaN son shaklida ifodalanadi, uning barcha ikkilik razryadlari va mantissa 1 ga teng.

**Cheksizliklar.** Bu shunday holatki, son tartibining barcha ikkilik razryadlari 1 ga teng bo'ladi va mantissa 0 ga teng bo'ladi (qo'zg'aluvchi vergulli sonlarda).

**Simvulli ma'lumotlar.**

Har bir simvolning ikkilik kombinatsiyasi mavjud. Simvollar va ularga belgilangan kodlar **kodlash jadvalini** hosil qiladi (4.5-rasm). Kodlashga talablar:

1. Raqamning kodi hajmi raqam oshib borgani sari oshib boradi
2. Harflarning hajmi (kodlangandagi hajmi) alfavit tartibida oshib boradi

Simvol	10 lik kod	2 lik kod	Simvol	10 lik kod	2 lik kod	Simvol	10 lik kod	2 lik kod	Simvol	10 lik kod	2 lik kod
	32	00100000	S	56	00111000	P	80	01010000	h	104	01101000
!	33	00100001	9	57	00111001	Q	81	01010001	i	105	01101001
"	34	00100010	:	58	00111010	R	82	01010010	j	106	01101010
#	35	00100011	:	59	00111011	S	83	01010011	k	107	01101011
\$	36	00100100	<	60	00111100	T	84	01010100	l	108	01101100
%	37	00100101	=	61	00111101	U	85	01010101	m	109	01101101
&	38	00100110	>	62	00111110	V	86	01010110	n	110	01101110
'	39	00100111	?	63	00111111	W	87	01010111	o	111	01101111
(	40	00101000	@	64	01000000	X	88	01011000	p	112	01110000
)	41	00101001	A	65	01000001	Y	89	01011001	q	113	01110001
*	42	00101010	B	66	01000010	Z	90	01011010	r	114	01110010
+	43	00101011	C	67	01000011	[	91	01011011	s	115	01110011
,	44	00101100	D	68	01000100	\	92	01011100	t	116	01110100
-	45	00101101	E	69	01000101	]	93	01011101	u	117	01110101
.	46	00101110	F	70	01000110	^	94	01011110	v	118	01110110
/	47	00101111	G	71	01000111	_	95	01011111	w	119	01110111
0	48	00110000	H	72	01001000	`	96	01100000	x	120	01111000
1	49	00110001	I	73	01001001	a	97	01100001	y	121	01111001
2	50	00110010	J	74	01001010	b	98	01100010	z	122	01111010
3	51	00110011	K	75	01001011	c	99	01100011	{	123	01111011
4	52	00110100	L	76	01001100	d	100	01100100		124	01111100
5	53	00110101	M	77	01001101	e	101	01100101	}	125	01111101
6	54	00110110	N	78	01001110	f	102	01100110	~	126	01111110
7	55	00110111	O	79	01001111	g	103	01100111	□	127	01111111

4.5-rasm. Ba'zi simvollar kodlar jadvali

**4.4. Mashhur kodlash jadvallari**

1. ASCII (American Standart Code for Information Interchange) 8 razryadli bo'lib, 8-razryad juftlikni aniqlash uchun ishlatiladi
2. Latin1 – ASCIIning Yevropa modifikatsiyasi (128 dan 255 gacha kodlar turli alfavitlarning maxsus harflarini, psevdografik simvollarini, grek alifbosi simvollarini, matematik va boshqa simvollarini ifodalash uchun ishlatiladi).

3. Unicode – sanoat standarti bo'lib, dunyoning barcha tillaridagi belgilarni, maxsus simvollar kodlash standarti. 1991-yilda taklif qilingan (4.6-rasm).

	"	#	\$	%	&	'	(	)	*	+	,	-	.	/	0	1	2	3	4
5	6	7	8	9	:	;	<	=	>	?	@	A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\
]	^	_	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
q	r	s	t	u	v	w	x	y	z	{		}	~	ı	ç	£	¤	¥	
ı	§	¨	©	ª	«	¬	®	¯	°	±	²	³	´	µ	¶	·	¸	¹	º
»	¼	½	¾	¿	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î
Ï	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß	à	á	â
ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï	ð	ñ	ó	ô	õ	ö	ø
ù	ú	û	ü	ý	þ	ÿ	Ā	ā	Ă	ă	Ą	ą	Ć	ć	Č	č	Ĉ	ĉ	Ċ

4.6-rasm. Unicoed simvollar jadvali

4.1-jadval.

**Boshqaruvchi simvollar jadvali**

DEC	OCT	HEX	BIN	Simvol	HTML kod	Описание
0	000	0x00	00000000	NUL	&#000;	Nolinchi bayt
1	001	0x01	00000001	SOH	&#001;	Sarlavhaning boshi
2	002	0x02	00000010	STX	&#002;	Matnning boshi
3	003	0x03	00000011	ETX	&#003;	Matnning oxiri
4	004	0x04	00000100	EOT	&#004;	Uzatishning yakuni
7	007	0x07	00000111	BEL	&#007;	Ovozli signal
8	010	0x08	00001000	BS	&#008;	Bir simvol orqaga (BACKSPACE)
9	011	0x09	00001001	TAB	&#009;	Tabulyatsiya
10	012	0x0A	00001010	LF	&#010;	Probel klavishi
13	015	0x0D	00001101	CR	&#013;	Enter klavishi
24	030	0x18	00011000	CAN	&#024;	Oxirgi amalni bekor qilish
25	031	0x19	00011001	EM	&#025;	Xotiraning oxiri
26	032	0x1A	00011010	SUB	&#026;	Moslashtirish
27	033	0x1B	00011011	ESC	&#027;	Escape (Kengaytma)
28	034	0x1C	00011100	FS	&#028;	Fayllarni ajratkich

## 5. Adreslash turlari, oqimlarni boshqarish. Buyruqlardagi manzillash usullari, uzilishlar

### 5.1. Adreslash turlari, oqimlarni boshqarish

**Bevosita adreslash.** Operandlarni aniqlashning eng oddiy usuli uning adres qismida adresni emas, balki operandni o'zini saqlashdir. Bu operand **bevosita operand** deyiladi, chunki u xotiradan komanda bilan birgalikda chaqiriladi (5.1-rasm).

MOV	R1	4
-----	----	---

5.1-rasm. Bevosita adreslash komandasi (bu yerdakonstanta 4 R1 registrga yoziladi)

Bevosita adreslashda operandni xotiradan chaqirish uchun qayta murojaat talab qilinmaydi (yutuq?). bu usulning kamchiligi – faqatgina konstantalar bilan ishlash mumkin, qolaversa konstanta qiymati maydon hajmi bilan cheklangan. Shunday bo'lsa ham bu texnologiya ko'pchilik arxitekturalarda konstantalarning butun sonligini aniqlashda ishlatiladi.

Operandni aniqlashning yana bir usuli **to'g'ri adreslash**. Bevosita adreslashga o'xshab to'g'ri adreslashning cheklanganligi sifatida komandaning faqatgina bitta adresga murojaat qila olishini aytish mumkin. Adres o'zgaydi, ma'lumotlar o'zgarishi mumkin. To'g'ri adreslashni kompilyatsiya davrida ishlatiladigan, adresi ma'lum global o'zgaruvchilarga nisbatan ishlatish mumkin. Deyarli barcha dasturlar global o'zgaruvchilar bilan ishlashini e'tiborga oladigan bo'lsak, bu usul ko'p qo'llaniladi.

**Registrli adreslash.** Bu adreslash to'g'ri adreslash bilan o'xshash, faqat bunda xotira yacheykasi o'miga registr ishlatiladi. Registrning juda muhimligi adresi qisqa va tez kirish mumkinligidir. Shu sababli ham bu uslub ko'pchilik kompyuterlarda keng miqyosda qo'llaniladi. Ko'p kompilyatorlar qaysi o'zgaruvchilarga eng ko'p murojaat qilinishini aniqlash uchun ko'p vaqt sarflashadi (masalan sikl indeksi) va bu o'zgaruvchilarni registrarga joylaydi.

Bunday adreslash **registrli adreslash** deyiladi. Yuklab-yozib qo'yish arxitekturasida (mas UltraSparcII) deyarli barcha komandalar ushbu adreslashni qo'llashadi. Faqat operand xotiradan registrga va registrdan xotiraga ko'chirilganda qo'llanilmaydi. Bunday vaziyatda ham bir operand registr vazifasini bajaradi, unga xotiradan so'z uzatiladi yoki undan xotiraga uzatiladi.

**Bilvosita registrli adreslash.** Bunday adreslashda aniqlanayotgan operand xotiradan olinadi yoki jo'natiladi, adres esa komandada qat'iy belgilab qo'yilgan emas. Buning o'rniga adres registrda joylashadi. Bunday ishlatilgan adres **ko'rsatkich** deyiladi. Bu usulning yutuqi shundaki, xotiraga murojaat qilganda komandada to'liq adresni ko'rsatmaslik ham mumkin. Bundan tashqari, bir komandani turli ishlatishda turli xotira so'zlaridan foydalanish mumkin.

Turli xotira so'zlaridan foydalanish qanday foydali bo'lishini tushunish uchun 1024 ta elementli bir o'lchamli sonli massivni yig'indisini R1 registriga yozish misolini ko'ramiz. Bu sikldan tashqaridagi R2 registri massivni 1-elementini ko'rsatadi, R3 registri massivdan keyingi 1-adresni ko'rsatadi. Massiv 1024 ta 4 baytli butun sonlardan tashkil topadi. Agar massiv A dan boshlansa, massivdan keyingi adres A+4096 bo'ladi. Ushbu ishni bajaruvchi assembler dasturi 5.2-rasmda

```
MOVRI.#0
MOV R2,#A
MOV R3,#A+4096
LOOP: ADD R1,(R2)
      ADD R2,#4
      OMP R2,R3
      BLT LOOP
```

5.2-rasm. Massiv elementlari yig'indisini hisoblash dasturi (assemblerda)

Bu dasturda adreslashning bir necha usullari keltirilgan. Dastlabki 3 ta komanda birinchi operand uchun registrli adreslashni va ikkinchi operand uchun bevosita adreslashni bajaradi. (# belgi bilan keltirilgan). Ikkinchi komanda R2 ga A ning qiymatini emas, uning adresini saqlaydi. Shu narsani assemblerga “#” belgisi bildiradi. 3-komanda R3 ga massivdan keyingi birinchi so'zni yozadi.

Sikl tanasida birorta ham xotira adresi yo'q. 4-komandada registrli va bilvosita adreslash ishlatilgan. 5-komandada registrli va bevosita adreslash, 6-komandada ikki marta registrli adreslash ishlatilgan. Shunday qilib xotira adreslarini ishlatmasdan qisqa va tezkor ishlovchi sikl hosil bo'ldi. Bu dastur Pentium II tipidagi kompyuterlar uchun mo'ljallangan.

**Indeksli adreslash.** Ko'pincha xotiradagi so'zlarga ko'chirish orqali murojaat qilishga to'g'ri keladi. Xotiraga bunday uslubda, ya'ni

registr va konstatnta ko'chirish orqali murojaat qilish **indeksli adreslash deyiladi**. Bunda lokal o'zgaruvchiga murojaat qilishda **xotira yacheykasi ko'rsatkichi** ishlatiladi.

Boshqacha usul ham mavjud. Xotira yacheykasi ko'rsatkichi komandada va ko'chirish registrda joylashgan holat. Buni quyidagi misolda ko'ramiz. A va B bir o'lchamli 1024 elementli massiv mavjud bo'lsin. A va B massiv elementlari ko'paymasidan nolga teng bo'lmagan birorta juftlik borligini aniqlash kerak. Birinchi variant A massiv adresini bir registrga, B massiv adresini ikkinchi registrga joylashtirib, keyin ularni ko'rib chiqish lozim.

Bunday dasturning ishini mukammalshtirilgan holati 5.3-rasmda keltirilgan.

```
MOV R1,#0
MOV R2,#0
MOV R3,#4096
```

```
LOOP: MOV R4,A(R2)
      AND R4,B(R2)
      OR R1,R4
      ADD R2,#4
      O.P R2,R3
      BLT LOOP
```

5.3-rasm. A va B bir o'lchamli 1024 elementli massivlarning ko'paymasidan nolga teng bo'lmagan juftlik borligini aniqlash dasturi

Bu yerda 4 ta registr zarur

R1 – yig'indi saqlanadi

R2 – massiv elementlarini ko'rib chiqishda qo'llaniladigan *i* indeksi

R3 – konstanta 4096. *i* ning eng kichik qiymati

R4 – har bir yig'indini saqlab turuvchi vaqtinchalik registr

Initsializatsiyadan keyin 6 satrli sikl boshlanadi. LOOP dan keyingi komanda  $A_i$  ni R4 ga yuklaydi. Hisoblashda bu indeksli adreslash ishlatiladi. R2 registri va konstanta (A elementi adresi) qo'shiladi, natija xotiraga murojaatda ishlatiladi. Bu ikki qiymatning yig'indisi xotiraga yoziladi, registrga emas.

*MOV R4,A(R2)*

yozuvi indeksli registrlash ishlatilayotganligini ko'rsatadi, bunda A-ko'chirish, R2-registr. Agar A ni qiymati 124300 ga teng bo'lsa, mashina komandasi quyidagicha bo'ladi (5.4-rasm).

MOV	R4	R2	124300
-----	----	----	--------

5.4-rasm. *MOV R4, A(R2)* komandasining tasvirlanishi

Siklning birinchi aylanishida R2 registr 0 ga teng, bizga kerakli so'z 124300 adrsli yacheykada bor. Bu so'z R4 registrga yoziladi. Siklning keyingi bosqichida R2 4 ga teng bo'ladi, bizga kerakli so'z 124304 registrda joylashadi va h.k. bu yerda ko'chirish – xotira yacheykasi ko'rsatkichi, registr qiymati esa butun son bo'lib, hisoblashlar jarayonida o'zgarib turadi. Ishni bunday tashkil qilinishi komandadagi ko'chirish maydoni adresni saqlash uchun tegishli katta bo'lishini talab qiladi. Shunday bo'lsa ham bu usul ko'pincha eng samarali deb topiladi.

**Nisbiy indeksli adreslash.** Ba'zi mashinalarda adreslashda adres ikkita registr qiymati va ko'chirishni yig'indisidan olinadi. Bunday yondashuv **nisbiy indeksli adreslash** deyiladi. 1-registr baza, 2-registr indeks. Uning qulayligi sikldan tashqarida R5 registrda A elementi adresini saqlab, R6 da B elementini saqlab, LOOP siklining dastlabki 2 komandasini quyidagicha o'zgartirishimiz mumkin

```
LOOP: MOV R4,(R2+R5)
```

```
AND R4,(R2+R6)
```

**Stekli adreslash.** Yuqorida aytilganidek, mashina komandalarini iloji boricha qisqa yozish kerak. Uning oxirgi chegarasi adressiz yozilgan komandadir. Adrsssiz komandalar stek bo'lganda ishlatish mumkin.

#### Teskari yozuv

Matematik yozuvda operator operandlar orasiga yoziladi  $(x+y)$ , oxirida emas  $(xy+)$ . 1-xolat **infiks yozuv** deyiladi, 2-xolat **postfiks yozuv** deyiladi. Postfiks yozuv algebraik formulalarni yozishda qo'l keladi. 1-dan qavssiz yozish mumkin, 2-dan stekli mashinalar uni oson hisoblaydi. Masalan  $axb+c$  yozuvi  $(axb)+c$  ni bildiradi,  $ax(b+c)$  ni yoki boshqa kombinatsiyani emas, chunki ko'paytirishning ustunligi yuqori. Ko'chirish amalining usutnligi VA amalidan yuqorimi? Teskari yozuv shunday tushunmovchiliklarni bartaraf etadi (5.5-rasm).

Инфиксная запись	Обратная польская запись
$A+B \times C$	$ABCx+$
$A \times B + C$	$ABxC+$
$A \times B + C \times D$	$ABxC Dx+$
$(A+B)/(C-D)$	$A B+C D-/$
$A \times B / C$	$ABxC/$
$((A+B) \times C + D) / (E + F + G)$	$AB+CD+ EF + G +/$

5.5-rasm. Ba'zi matematik yozuvlarning teskari yozuvda ifodalanishi

Har ikkala yozuvda operandlarning yozilish ketma-ketligi bir xil. Amallarning yozilishi esa teskari yozuvda qaysi tartibda bajarilsa, shu tartibda yoziladi.

### 5.2. Teskari yozuvda formulalar hisob-kitobi

Bu yozuv stekli kompyuterlar uchun juda qulay hisoblanadi. Formula  $n$  ta belgidan iborat bo'lib, har bir belgi yoki operand, yoki amal. O'qilishi tartibi quyidagicha amalga oshiriladi: formula chapdan o'ngga qarab o'qib boriladi. Agar operand uchrasa u stekka joylashtiriladi, agar amal uchrasa u amalni stekdagi operandlar bilan bajariladi (5.6-rasm).

Matematik yozuv  $(8+2 \times 5) / (1+3 \times 2-4)$

Infiks yozuv  $825x+132x+4-/$

5.6-rasm. Matematik yozuvning infiks yozuvda ifodalanishi

Stek cho'qqisidagi son o'ng tarafdagi operand (chapdagi emas, chunki chapdagi operand keyingi operand kelishi bilan bir pastga surilgan). Bu ayirish va bo'lish amallari uchun juda muhim, chunki operandlarning stekda joylashish tartibi bu yerda muhimdir. Bo'lish amali quyidagicha hisoblanadi. Dastlab stekka surat joylashadi, keyin esa maxraj joylashadi.

8	2	5	x	+	1	3	2	x	+	4	-	/			
8	2	5	$5 \times 2 = 10$		10	8	18	1	3	2	$2 \times 3 = 6$	$6 + 1 = 7$	4	$7 - 4 = 3$	$18 / 3 = 6$
	8	2	8			18	1	3	1	18	7	18			
		8				18	1	18		18					

5.7-rasm. Stekda bajariladigan amallarning bajarilish ketma-ketligi

Teskari yozuvni matematik yozuvga o'zgartirish oson. Buning uchun teskari yozuvni chapdan o'qib borish va har bir operanddan keyin bir amal belgisini qo'yib borish lozim. Agar kelgan simvol konstanta yoki o'zgaruvchi bo'lsa, uni stekka joylashtirish lozim. Agar simvol arifmetik amal bo'lsa, stekning yuqorigi 2 elementi bilan shu amalni bajarish lozim.

### 5.3. O'tish komandasi uchun adreslash usullari

Mashinada o'tish komandalari maqsadli adreslash uchun maxsus adreslash usullari zarur. Buning bir varianti to'g'ridan-to'g'ri adreslash, bunda maqsadli adres komandaga qo'shilib ketadi. Adreslashning boshqa usullari ham mavjud. Bilvosita registrli adreslash dasturga maqsadli adresni hisoblash, uni registrga joylashtirish va unga o'tishni amalga oshiradi. Bu usulning ahamiyati uning maksimal egiluvchanligidir, chunki u bevosita dastur bajarilishi davomida amalga oshadi. Lekin buning natijasida xatolarning paydo bo'lish ehtimoli yuqori bo'ladi.

Registr ko'chishli indeksli adreslash ham maqbull yechim hisoblanadi. Yana bir yechim komandalar hisoblagichiga nisbatan adreslash. Bunda maqsadli adresni olish uchun komandada joylashgan ko'chish amali dastur hisoblagichiga qo'shiladi. Mohiyatan bu registr sifatida PC ishlatiladigan indeksli adreslashning o'zidir.

Qadam	Bajariladigan amallar	Komanda	Stek
1	$8\ 2\ 5\ x\ +\ 1\ 3\ 2\ x\ +\ 4\ -\ /$	BIPUSH 8	8
2	$25x+132x+4-/$	BIPUSH 2	8,2
3	$5x+132x+4-/$	BIPUSH 5	8,2,5
4	$x+132x+4-/$	IMUL	8,10
5	$+132x+4-/$	IADD	18
6	$132x+4-/$	BIPUSH 1	18,1
7	$32x+4-/$	BIPUSH 3	18,1,3
8	$2x+4-/$	BIPUSH 2	18,1,3,2
9	$x+4-/$	IMUL	18,1,6
10	$+4-/$	IADD	18,7
11	$4-/$	BIPUSH 4	18,7,4
12	$-/$	ISUB	18,3
13	$/$	IDIV	6

5.8-rasm. Postfiks yozuvni stekda bajarilish jarayoni

### 5.4. PentiumII protsessorida adreslash usullari.

Pentium II da adreslash usullari doimiy emas va hosil bo'layotgan komanda qanday formatda (16, 32 yoki 64 bitli) ekanligiga bog'liq

bo'ladi. Bu yerda 32 bitli komandalari ko'rib chiqamiz. Adreslash turlaridan bevosita, to'g'ridan-to'g'ri, bilvosita registr indeksli va maxsus adreslashlar ishlatiladi. Muammo shundaki, komandalarga nisbatan barcha usullar qo'llanila olmaydi, yana hamma adreslash usullarida hamma registrlar ham ishlatilavermaydi. Bu esa kompilyator yozuvchining ishini juda murakkablashtiradi.

### 5.5. Buyruq (komanda) lardagi manzillash usullari, uzilishlar.

Mashina komandalarini turli usullarda guruhlariga ajratish mumkin. Ular turli mashinalarda takrorlanishi, detallarda farqlanishi mumkin. Har xil kompyuterlarda alohida komanda ham mavjud bo'lib, asosan avvalgi tizimlar bilan moslashish uchun qo'llaniladi.

### 5.6. Ma'lumotlarni ko'chirish komandalari

Ma'lumotlarni bir joydan ikkinchisiga ko'chirish eng keng qo'llaniladigan amaliyotdir. Ko'shirish deganda bir joydagi ma'lumotlarning ikkinchi joyda asl nusxasini paydo qilish tushuniladi. Bunda "ko'chirish" termini o'rniga "nusxalash" ishlatilsa aniqroq bo'ladi. Ko'chirishda ikkinchi joyda shu ma'lumot hosil qilinishi bilan avvalga joydagi ma'lumot ma'lum vaqt saqlanib turadi.

Ikkita sababga ko'ra ma'lumotlar ko'chirilishi mumkin. 1-sabab fundamental tushuncha: o'zgaruvchilarga qiymat berish. Mas  $A=B$  kabi. Bunda B o'zgaruvchi joylashgan yacheykadagi ma'lumotlar A o'zgaruvchi joylashgan yacheykaga o'tadi. 2-sabab ma'lumotlarga murojaat qilishni osonlashtirish (tezlashtirish). Ma'lumotlarning faqat ikkita joylashadigan joyi (registr va xotira) mavjud ekan, ularning ko'chiriladigan joyi ham ikkita (xotira va registr), shundan kelib chiqib 4 xil ko'chirish uslubi mavjud. (reg-reg, reg-xotira, xotira-reg, xotira-xotira) Ba'zi kompyuterlarda har bir xolat uchun alohida, jami 4 ta komanda ishlatiladi. Masalan LOAD – o'qish, STORE – registrdan xotiraga ko'chirish, MDE – registrlararo ko'chirish. Nusxalash komandasi esa mavjud emas.

Ko'chirish komandalari qancha miqdordagi ma'lumot ko'chirilishini aniqlab berishi lozim. Ba'zi komandalar turli sondagi axborot birliklarini ko'chira oladi (1 bit dan toki xotiraning hammasini). So'z uzunligi belgilab qo'yilgan mashinalarda odatda bir harakatda bir so'z ko'chiriladi. Boshqa har qanday ko'chirishlar dasturiy boshqaruv orqali bajariladi. Ko'plab so'zlarni ko'chirish xavfli, chunki bu ko'p vaqtni talab qilishi va oraliqda uzilib qolishi mumkin. Ba'zi so'z

uzunligi o'zgaruvchi mashinalarning komandalari boshlang'ich va oxirgi adreslarni belgilaydi, ma'lumotlar sonini emas. Ma'lumotlarni ko'chirish to'xtash komandasiga yetguncha bajarilaveradi.

**Binar amallar.** Bu shunday amallarki, ikkita operanddan bitta natija hosil bo'ladi. Kompyuter arxitekturasi butun sonlarni qo'shish va ayirish, ko'paytirish va bo'lish komandalaridan iborat.

Binar komandalarning keyingisi mantiqiy komandalardir. Ikkita o'zgaruvchi bilan ishlovchi 16 ta funksiya mavjud. Lekin ushbu 16 ta komandani hammasini ishlatadigan mashina mavjud emas. Asosan VA, YOKI, YO'Q amallari ishlatiladi. Ba'zan INKOR-YOKI, YO'Q-YOKI, YO'Q-VA ishlatiladi.

Har biri 8 bitli simvolli bo'lgan 32 bit uzunlikdagi so'zlar bilan ishlaydigan mashinani qaraymiz. Aytaylik, ikkinchi so'zni boshqalardan ajratib olish kerak bo'lsin. Oddiy aytganda shunday so'z hosil qilish kerakki, o'ng tomonda 8 bitda joylashsin va chap tomonda 24 ta nollar bo'lsin. (o'ng bitga tenglanish texnologiyasi)

Kerakli simvolni olish uchun bu so'z va konstanta VA amali bilan bog'lanadi. Natijada keraksiz bitlarning barchasi nollarga aylanadi. Keyin natija kerakli simvol o'ng tomonda joylashishi uchun 16 bit o'ngga suriladi.

YOKI komandasini qo'llashda so'zga bitlarni kiritish nazarda tutiladi. 32 bitli so'zning faqat oxirgi 8 bitini o'zgartirish uchun shu 8 bitni dastlab 0 larga aylantirib olamiz, hosil bo'lgan simvolni YOKI amali bilan qo'shish kerak.

Hozirgi ko'pchilik kompyuterlar qo'zg'almas vergulli son bilan ishlovchi komandalarni qo'llaydi. Mashinalar bunday sonlarning 2 variantini saqlaydi: tezlikni oshirish uchun qisqa formatdagi va hisoblash aniqliklarini oshirish uchun uzun so'zlar.

### 5.7. Tenglash va shartli o'tish.

Deyarli hamma dasturlar o'z ma'lumotlarini tekshirib, uning natijasiga ko'ra bajariladigan komandalarni ketma-ketligini o'zgartirishi lozim. Ba'zan komandadagi ma'lum bir bit shartli o'tishni ko'rsatadi. Ko'pincha maqsadli adres absolyut emas, nisbiy bo'ladi.

Shartli o'tishda shartning bajarilishiga qarab LABEL belgisiga o'tish yoki undan shartli operatoridan keyingi amallarni bajarish mumkin. Ko'p mashinalarda alohida shartlarni ko'rsatuvchi shart kodi biti mavjud. Masalan to'lganlik biti 1 ga teng bo'ladi, agar arifmetik amal natijasi xatolikka olib kelsa. Ko'chirish biti va to'lganlik bitini

adashtirmaslik kerak. Ko'chirish biti hisoblashlar aniq bo'lishi uchun kerak. O'ga tekshirish esa sikllarni bajarishda zarur.

So'zlar va simvollarni tenglash ko'p xollarda, xususan saralash algoritmidan muhim. Tenglashni bajarish uchun uch adres zarur bo'ladi: 2 ta ma'lumotlar uchun, 3-si shart bajarilganda o'tish uchun. Komandalari 3 adresli bo'lgan kompyuterlar uchun bu muammo tug'dirmaydi, aks holda qandaydir boshqa yo'llarni topish zarurati tug'iladi.

Mumkin bo'lgan yechimlardan biri tenglikni tekshiruvchi va natijani shart bitiga yozib qo'yuvchi komandani kiritishdir. Keyingi komanda shart bitlarini tekshirish va o'tkazishni bajarishi mumkin. Bunday yondashuv PentiumII va UltraSPARCII larda qo'llaniladi.

Sonlarni taqqoslashda bir qancha nozik jihatlari mavjud. Tenglash qo'shish yoki qyirish kabi oddiy amal emas. Agar katta musbat son katta manfiy son bilan taqqoslansa, ayirish amali to'lib ketishga olib kelishi mumkin. Tenglashda esa to'lib ketish holatlari sodir bo'lmaydi. Tanglashda sonlarni ishorali yoki ishorasiz ekanligini hal qilish lozim.

### 5.8. Protseduralarni chaqirish komandalari

Protsedura – ma'lum bir vazifani bajaruvchi va asosiy dasturning istalgan joyidan chaqirilishi mumkin bo'lgan dastur. Assembler tilida uning o'miga podprogramma termini ishlatiladi. Protsedura bajarilib bo'lganidan keyin chaqiruvchi komandaga qaytadi, bundan qaytish adresi protseduraga uzatilishi yoki qayerdadir saqlanishi kerakligi kelib chiqadi.

Qaytish adresi 3 ta joyda bo'lishi mumkin: xotirada, registrda yoki stekda. Optimal yechim adresni protseduraning birinchi so'zida saqlash. Protsedura ikkinchi so'zdan bajarilishni boshlaydi. Protsedura boshqa protseduralarni chaqira oladi, chunki har birida bitta qaytish adresi uchun makon mavjud. Agar protsedura o'zini o'zi chaqirsa (**rekursiya**), bu sxema ishlamaydi, chunki ikkinchi chaqirishda birinchi adres yo'qolib ketadi. Bundan tashqari A protsedura B ni chaqirsa, B esa C ni chaqirsa va C protsedura A ni chaqirsa ham (**zanjirli rekursiya**) yuqoridagi holat ro'y beradi. Yanada maqbulroq yechim qaytish adresini registrga joylash. Rekursiya bo'lganda qaytish adresini har chaqirilganda har joyga ko'chiradi. Eng yaxshi yechim qaytish adresini stekka joylashtirish. Protsedura yakunlanganda qaytish adresini stekdan chiqarib yuboradi. Protseduralarni bunday chaqirishda rekursiya muammo

hosil qilmaydi: qaytish adresi avtomatik ravishda oldingi adresni yo'qotishmaydigan bo'lib saqlanadi.

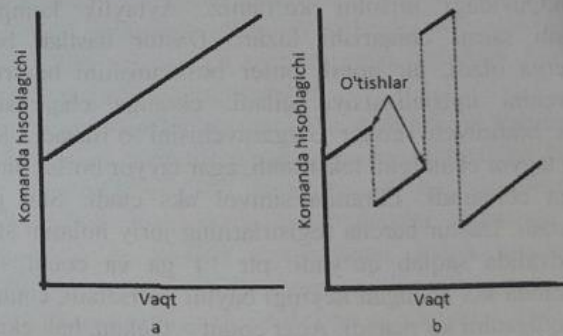
### 5.9. Sikllarni boshqarish

Sikllarni boshqarishni osonlashtirish uchun ba'zi mashinalar maxsus komandalarga ega bo'ladi. Barcha komandalari hisoblagichga ega bo'lib, uning qiymati siklning har bir bajarilishida mos ravishda kamayadi yoki ortadi va qiymat tekshiriladi. Qiymat qo'yilgan shartni bajarganda sikl bajarilishi to'xtaydi.

**Boshqarish potoki.** Boshqarish potoki – komandalari dasturning bajarilishi davomida dinamik bajarilishidir. Komandalari xotiradan ketma-ket o'qib bajariladi. Protsedurani chaqirish yoki o'tish komandasi boshqarish potokini o'zgarishiga olib keladi, ya'ni dasturning asosiy qismi bajarilishi to'xtaydi, boshqaruv boshqa qismga o'tadi.

### 5.10. Ketma-ket boshqarish potoki va o'tishlar

Ko'p komandalari boshqarish potokini o'zgartirmaydi. Bir komandadan keyin undan keyingi komanda bajariladi. Har komanda bajarilgandan keyin komanda ko'rsatkichi komandaning uzunligiga mos ravishda o'zgaradi. Komanda hisoblagichi o'rtacha vaqt davomida o'rtacha miqdorga o'zgaruvchi vaqtga bog'liq chiziqli funksiyadir.



5.9-rasm. Komanda hisoblagichining vaqtga bog'liqligi: (a) o'tish komandasiz; (b) o'tish komandali

O'tish mavjud bo'lganda komanda hisoblagichi chiziqli funktsiya shaklini yo'qotadi.

**Protseduralar.** Dasturlarni strukturalashda (o'qilishini yengillashtirishda) protseduralar muhim ahamiyat kasb etadi. Bir



tomondan protsedutani chaqirish o'tish komandasi sifatida baholanadigan bo'lsa, boshqarish potokini o'zgartiradi, lekin, o'tish komandasidan farqli ravishda chaqirilgan komandaga qaytmaydi. Boshqa tarafdan protsedura tanasini yangi, yuqori darajadagi komanda deb qabul qilish mumkin. Bunday qarashda protsedurani chaqirishni o'zi alohida komanda sifatida qaraladi, protsedura qanchalik murakkabligi esa ahamiyatsiz. **Rekursiv protsedura** alohida o'rganishni talab qiladi. Bu o'zini-o'zi chaqiruvchi protsedura bo'lib, uni o'rganish protsedurani chaqirish tartibini va lokal o'zgaruvchilarni aniqlab beradi. Bu yerda Xanoy minorasini misol qilib keltirish mumkin.

**Uzilishlar.** Uzilishlar – odatda o'qish-yozish yoki boshqa jarayonlar bilan bog'liq bo'lgan, bevosita dasturning aloqasi bo'lmagan boshqarish potokidagi o'zgarishlardir. Masalan, dastur diskka ma'lumotlarni uzatishni boshlash haqida buyruq berishi mumkin va uzatish yakunlanishi bilan uzilishni buyurishi mumkin. Uzilish dasturning bajarilishini to'xtatadi va boshqaruvni **uzilishlarni qayta ishlash dasturiga** uzatadi. Bu dastur tegishli harakatlarni bajargandan keyin boshqaruvni asosiy dasturga uzatadi. U dasturning qaysi holatda to'xtatilgan bo'lsa, shu xolatdan boshlashi kerak, boshqacha aytganda barcha registrlarning holati uzilishdan oldingi holatiga to'liq qayta tiklanishi lozim. Quyidagi misolni ko'ramiz. Aytaylik kompyuter terminalga matnli satrni chiqarishi lozim. Dastur dastlab barcha simvollarini buferga oladi, ptr nomli bufer boshlanishini bildiruvchi global o'zgaruvchini initsializatsiya qiladi, ekranga chiqariladigan simvollar sonini bildiruvchi count o'zgaruvchisini o'rnatadi. Keyin dastur ekranning tayyor ekanligini tekshiradi, agar tayyor bo'lsa birinchi simvolni ekranga chiqaradi. Ekranda simvol aks etadi. Shu joyda uzilishlar boshlanadi. Dastur barcha registrlarning joriy holatini stekda yoki sistema jadvalida saqlab qo'yadi. ptr +1 ga va count -1 ga orttiriladi. ptr ekranda aks etadigan keyingi baytni ko'rsatadi, count esa yana necha bayt qolganini ko'rsatadi. Agar count > 0 ekan, hali ekranga chiqadigan simvollar mavjud. ptr ko'rsatib turgan simvol bufer registriga o'tkaziladi. Barcha saqlangan registrlar qayta tiklanadi va shu tarzda jarayon davom etadi. Uzilishlar bilan bog'liq tiniqlik tushunchasi mavjud. Uzilish sodir bo'lganda qandaydir amallar bajariladi, qandaydir dasturlar yuklanadi va h.k., lekin hammasi tugagandan keyin kompyuter avvalgi holatiga qaytishi kerak. Bunday xususiyatga ega dastur tiniqlikka ega dastur deyiladi.

## 6. Kompyuterlarning asosiy ko'rsatkichlari: massiv-parallel tizimlar (Massiv-parallel protsessorlar), simmetrik multiprotsessorli tizimlar (SMP), xotiraga bir xil bo'lmagan ruxsatlilik tizimlari (NUMA)

### 6.1. Kompyuterlarning asosiy ko'rsatkichlari

Parallel harakat tizimli kompyuterlarni tanlashda quyidagi ko'rsatkichlarga e'tibor qaratiladi:

- 1) Protsessor tipi, razmeri, elementlar soni;
- 2) Xotiraning tipi, razmeri, modullar soni;
- 3) Xotira va protsessor elementlari qanday o'zaro munosabatda bo'lishi.

Protsessor elementlari turli tipli bo'lishi mumkin – kichik arifmetik-mantiqiy qurilmadan to to'liq mikroprotsessorgacha. Razmeri jihatdan esa mikrosxemaning kichik qismidan to  $1 \text{ m}^3$  gacha. Kichik hajmli elementdan ko'p sondagilarini kompyuterga joylashtirish mumkin. Katta bo'lsa aksincha. Parallel harakatli kompyuterlar seriyali ishlab chiqariladigan qismlardan tuziladi.

Xotira tizimi bir nechta protsessorlar xotiraga murojaat qila olishi uchun o'zaro mustaqil modullarga bo'linadi. Bu modullar bir necha KB dan bir necha MB gacha bo'lishi mumkin. Protsessor yonida yoki alohida platada joylashishi mumkin. Dinamik operativ xotira markaziy protsessor ga qaraganda sekinroq ishlaydi, shu sababli xotiraga tezroq murojaat qilish uchun turli kesh xotiralar ishlaydi. Ko'p darajali kesh xotiralar ham qo'llanilishi mumkin.

Turli protsessorlar va xotiralar mavjud bo'lishidan qat'iy nazar parallel harakat tizimlari o'zaro qanday ulanganiga ko'ra farqlanadi. O'zaro bog'lanishni ikki kategoriyaga bo'lish mumkin: statik va dinamik. Statik sxemalarda komponentlar qandaydir tartibga ko'ra ulanadi, masalan yulduz topologiyasi, halqa topologiyasi va h.k. Dinamik sxemalarda hamma komponentlar o'tkazgich sxemaga ulanadi, u esa ma'lumotni tegishli yo'nalishga mos holda uzatadi.

Parallel harakatli kompyuterlarda aynan qanday jarayonlar parallel bajariladi? Buning bir necha variantlari mavjud. Ba'zi mashinalar bir vaqtda o'zaro bir-biriga bog'liq bo'lmagan bir necha masalalarni bajaradi. Ikkinchi variant bir necha jarayondan iborat masalani yechish. Misol shaxmat partiyasi. Keyingi yurishdan oldin yurish mumkin bo'lgan barcha variantlar olinadi, har bir variant parallel ravishda alohida hisob-kitob qilinadi (yutuqqacha). Bunda parallelik ko'p sonli

foydalanuvchilar bilan ishash uchun emas, balki bitta masalani yechimini tez topish uchun ishlatiladi.

Keyingi bosqichda ko'p konveyerli yoki ko'p arifmetik-mantiqiy qurilmali mashinalar keladi. Ular potokli komandalar bilan ishlaydi. Masalan vektorli ma'lumotlar bilan ishlovchi maxsus texnik ta'minotli superkompyuterlar (kinostudiyalarda maxsus effektlarni ishlatishda yoki mavjud bo'lmagan personajlarni jonlantirishda). Bunda ham bitta masala hal qilinadi, mashinaning turi qismlari birgalikda ishlaydi.

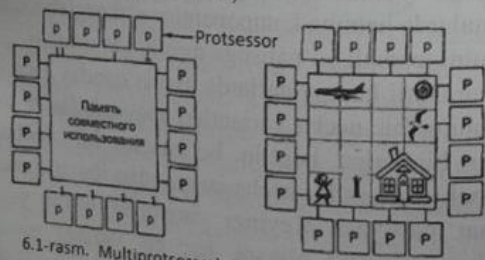
Yuqoridagilardan tizimlar o'zaro **detalizatsiya darajasi** bo'yicha farqlanadi. Vaqtda bo'linishli ko'pprotsessorli tizimlarda parallellik bloki yetarlicha katta, deyarli foydalanuvchi dasturiga teng. Dasturiy ta'minotning ko'p qismlarining o'zaro aloqasiz parallel ishlashi **yirik birliklar darajasidagi parallellik** deyiladi. Qarama-qarshi xolat esa (vektorli ma'lumotlarni ishlashda) **kichik birliklar darajasidagi parallellik** deyiladi.

**Detalizatsiya darajasi** termini faqat dasturiy ta'minotga nisbatan ishlatiladi, lekin uning analogi texnik ta'minotda mavjud. Kam sonli protsessorlarga ega, ma'lumotlarni sekin o'tkazuvchi sxemalar bilan ishlaydigan tizimlar **bilvosita (bo'sh) aloqali tizimlar** deyiladi. Uning qarama-qarshisi **bevosita (mustahkam) aloqali tizimlar** bo'lib, uning komponentlari nisbata kichik, o'zaro yaqin joylashgan va maxsus yuqori o'tkazuvchanlikka ega tarmoq orqali bog'lanadi.

### 6.2. Informatsion modellar

Har bir parallel tizimlarda masalaning turli qismlarini yechuvchi protsessorlar axborot almashish maqsadida o'zaro muloqotga kirishishlari zarur. Buni qanday tashkil qilish kerak? Ikki xil yechim mavjud: multiprotsessor va multikompyuter.

Bunday tizim multiprotsessor deyiladi yoki hamkorlikda ishlatiladigan tizim deyiladi (6.1-rasm).



6.1-rasm. Multiprotsessorlar. Barcha protsessorlar umumiy xotiradan foydalanadi.

Multiprotsessorli tizim dasturiy ta'minotga mo'ljallangan. Multiprotsessorda bajariladigan barcha jarayonlar umumiy xotiradagi yagona virtual maydonda ishlaydi. Har qanday jarayon xotiradan so'zni o'qish va/yoki yozishi mumkin (*LOAD* va *STORE* komandalari yordamida). Ikkita protsessorning biri ma'lumotni o'qisa va biri yozadigan bo'lsa ular axborot almasha oladi.

Shunday o'zaro munosabat sababli multiprotsessorlar keng tarqalgan. Uning tushunariligi va keng turdagi masalalarni hal qila olishi ahamiyatga ega. 16 ta protsessorning har biri bir jarayonni boshqaradi. Qaysidir jarayonning obyektlari seksiya chegarasidan chiqsa, jarayon obyektning izidan keyingi seksiyaning so'zlarini o'zlashtirib, unga o'tib ketadi. Masalan SunEnterprise 10 000, Sequent NUMA-Q, SGI Origin 2000, HP/Convex Exemplar.

### 6.3. Multikompyuterlar

Birinchi yondashuvda parallel arxitekturaning ikkinchi usulida har bir protsessor faqat o'zigagiga tegishli bo'lgan o'zining xotirasiga ega bo'ladi. Bunday tizim multikompyuter yoki tarqoq xotirali tizim deyiladi.

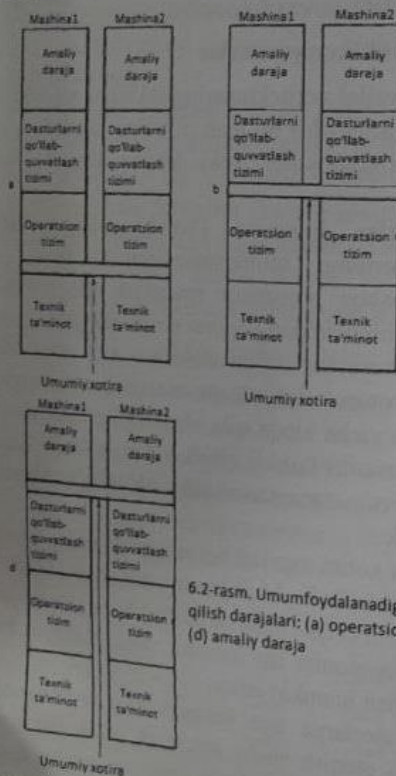
Odatda multikompyuterlar bilvosita (bo'sh) aloqali tizim hisoblanadi. Multikompyuterning multiprotsessordan farqi bir protsessorning xotirasiga boshqa protsessor murojaat qila olmaydi. Multiprotsessorlar umumiy ishlatiladigan yagona xotira maydoniga ega, multikompyuterlar har bir protsessor uchun alohida xotira maydoniga ega. Multikompyuterdagi protsessorlar o'zaro munosabatda umumiy xotiraga yozish va o'qish bo'yicha aloqa qila olmasligi e'tiboridan bu yerda boshqacha aloqa tuzilmasini tashkil qilish lozim bo'ladi. Bunda o'zaro ulanish tarmog'i (сеть межсоединений) ishlatiladi. Masalan *IBM PS/2*, *Intel/Sandia* va h.k

Umumiy foydalanadigan xotira mavjud bo'lmaganda uning o'rniga maxsus dasturiy ta'minot ishlatiladi. Multikompyuterlarda *LOAD* va *STORE* komandalarini ishlatish orqali hamma jarayonlar ma'lumotlarni o'qiy oladigan va hamma jarayonlar ma'lumotni yoza oladigan bitta virtual adresli muhit hosil qilish mumkin emas.

Bir xil sondagi protsessorlarga ega bo'lgan multiprotsessordan ko'ra multikompyuter ishlab chiqish ancha arzon va oson. Bir necha yuzlab protsessorlar hamkorlikda ishlatadigan umumiy xotira qurish juda murakkab ish, 10 000 va undan ko'p protsessorli multikompyuter ishlab chiqish oson.

Dilemma: multiprotsessorlarni qurish qiyin, dasturlash oson, multikompyuterlarda teskarisi. Shu sababli dasturlash va qurish nisbatan oson bo'lgan gibrud tizimlar ustida izlanish olib borilgan. Buning natijasida umumiy xotirani turli usullarda ishlatish mumkinligi aniqlandi. Arxitekturani qurish bo'yicha deyarli barcha izlanishlar har ikkala arxitekturaning yutuqlarini o'zida maksimal jamlagan tizimni topish yuzasidan olib borilgan. Bunda kengaytirish imkoniyati mavjud bo'lgan tizimlarni qurish (yangi protsessorlarni qo'shganda ham ishlashda davom etadigan) ahamiyatlidir.

Bir yondashuv shundayki, bunda kompyuter tizimi bir butun emas, balki turli darajalardan iborat deb qaraladi. Bu turli darajadagi xotirani tashkil qilish imkonini beradi (6.2-rasm).



6.2-rasm. Umumfoydalanadigan xotiradan foydalanishni tashkil qilish darajalari: (a) operatsion tizim; (b) dasturiy ta'minot; (d) amaliy daraja

Ikkinchi yondashuv sahifalarga bo'lingan, yagona virtual adres makonini ishlatadigan ajratilgan xotira modelini beruvchi operatsion

tizim va texnik ta'minotni ishlatish. Bunday yondashuvda (DSM- **Distributed Shared Memory – umumiy foydalanadigan ajratilgan xotira**) har bir sahifa alohida xotira bloklarida joylashadi. Har bir mashina o'zining virtual xotirasiga ega bo'ladi. Agar mavjud bo'lmagan sahifadan *LOAD* yoki *STORE* komandalari chaqirilsa operatsion tizimning uzilishi ro'y beradi. Keyin operatsion tizim kerakli sahifani topadi va shu sahifani ishlatayotgan protsessorga uni tegishli shaklda o'zgartirib, o'zaro bog'lanish orqali yuborishni talab qiladi. Bu talab bajarilgandan keyin uzilgan komandaning bajarilishi davom etadi (yangilanadi). Demak, protsessorga yetishmayotgan sahifalarni diskdan emas, xotiradan chaqiriladi. Tashqaridan qaraganda mashina umumiy taqsimlangan xotiraga egadek ko'rinadi.

Uchinchi yondashuv umumiy taqsimlangan xotirani dasturiy ta'minot darajasida amalga oshirish. Bunda taqsimlangan xotirani dasturiy ta'minotda tashkil qilish. Bu yondashuvda taqsimlangan xotirani dasturlash tili yaratadi, kompilyator uni tashkil qiladi. Umumiy foydalanadigan xotirani tashkil qilishning usuli Orca sistemasidagi umumiy obyektlar modelidir. Orca modelida jarayonlar obyektlarni taqsimlab oladi va ular bilan ishlaydi. Agar protsedura obyektning ichi holatini o'zgartirsa, operatsion tizim ushbu obyektning barcha mashinalardagi nusxalari o'zgartirilganligini nazorat qiladi. Bunda obyekt fizik emas, dasturiy tushuncha bo'lgani sababli, uni OT yoki dasturiy ta'minotning aralashuvsiz tashkil qilish mumkin.

#### 6.4. SIMD kompyuterlari

SIMD (Single Instruction Stream Multiple Data Stream-bir potok komanda, bir necha potok ma'lumotlar) kompyuterlari vektorlar va massivlar bilan bog'liq ilmiy va texmik masalalarni yechishga mo'ljallangan. Bunday mashinada bitta boshqarish bloki mavjud. U komandalarni bitta bittadan ijro etadi, har bir komanda esa bir necha ma'lumotlar elementlarini boshqaradi. SIMD tipidagi 2 ta asosiy kompyuter – massiv-parallel protsessorli (*array protsessors*) va vektor protsessorli (*vector protsessors*).

#### 6.5. Massiv parallel protsessorlar

Massiv-parallel protsessorli ilk mashina *ILLIAC IV*. Bunday protsessorida ko'plab ishlanuvchi elementlarni ishga tushirish uchun signallarni uzatuvchi bitta boshqaruv blokiga ega. Bunda har bir

element protsessor, arifmetik-mantiqiy qurilma va lokal xotira elementidan tashkil topadi.

Barcha massiv-parallel protsessorlar ushbu umumiy modelga mos kelsada ular o'zaro farqlanadi. Bu farqlar esa quyidagilarda ko'zga tashlanadi. 1) qayta ishlovchi element strukturasi bo'yicha. U juda oddiy ham bo'lishi mumkin, juda murakkab ham bo'lishi mumkin. Eng oddiy qayta ishlanuvchi element 1 bitli arifmetik-mantiqiy qurilma. Bunday mashinada har bir arifmetik-mantiqiy qurilma 2 ta 1 bitli operandni o'z lokal xotirasidan o'qiydi. Hisoblash natijasida 1 bit ma'lumot va bir necha bayroq bitlari hosil bo'ladi. Qayta ishlovchi element 8 bitli, 32 bitli yoki boshqa qo'zg'almas vergulli son bilan ishlovchi qurilma bo'lishi mumkin. Ba'zi hollarda qayta ishlovchi element turini tanlash mashinaning mo'ljallanishiga qarab tanlanadi. Qo'zg'almas vergulli sonni ishlatish turli matematik hisoblashlarda zarur, oddiyroq amallar, masalan, qidiruvda shart emas. 2) qayta ishlovchi elementlarning o'zaro bog'liqligi. Bunda quyidagi barcha topologiyalar tadbiq qilinishi mumkin. Ko'pincha to'rtburchakli panjara ishlatiladi, chunki ular matritsali masalalarni yechishda qulay, o'z tizimiga yangi protsessorlar qo'shish bilan o'tkazuvchanlik qobiliyati ortishi natijasida katta hajmli masalalar uchun oson moslashadi. 3) qayta ishlovchi elementlar qanday lokal mustaqillikka egaligi. Qaysi komandani bajarish kerakligini boshqaruv bloki ko'rsatadi, lekin ko'pincha Massiv-parallel protsessorlarda har bir qayta ishlovchi element bir necha lokal ma'lumotlar asosida ushbu komandani bajarish yoki bajarmaslikni hal qiladi va bu protsessorga egiluvchanlikni beradi.

### 6.6. Vektorli protsessorlar

SIMD mashinasining ikkinchi tipi **vektorli protsessor**. Ular keng tarqalgan, chunki bir necha o'n yillar davomida yetakchi bo'lib kelgan (masalan Cray-1, C90, T90 va h.k mashinalar). Katta hajmli raqamli ma'lumotlarni tez qayta ishlovchi vositalar

$for(i=0; i<n; i++) a[i]=b[i]+c[i];$

shaklidagi tuzilmalar bilan to'lgan. Bunda  $a, b, c$  lar Qo'zg'almas vergulli sonli vektorlar (yoki sonli massivlar). Siklda  $b$  va  $c$  massiv elementlarini yig'indisini  $a$  da saqlashni buyuradi. Odatda elementlar ketma-ket qo'shilib borishi nazarda tutiladi, lekin bunday tartib bo'lishi hech qanday ahamiyatga ega emas.

Bunda vektorli arifmetik-mantiqiy qurilma keltirilgan. Bunday mashina kirishda 2 ta  $n$  elementli vektorni o'qiydi va  $n$  ta element bilan

bir vaqtda ishlovchi vektorli arifmetik-mantiqiy qurilma yordamida ularni parallel qayta ishlaydi. Hisoblash natijasi ham vektor bo'ladi.

Vektorli kompyuterlar skalyar (vektorli bo'lmagan) amallar uchun ham qo'llaniladi, shuningdek aralash (vektor-skalyar) amallar uchun ham ishlatiladi. Vektorli amallarning asosiy tiplari quyidagi jadvalda keltirilgan:

Vektorli amallarning asosiy tiplari

6.1-jadval.

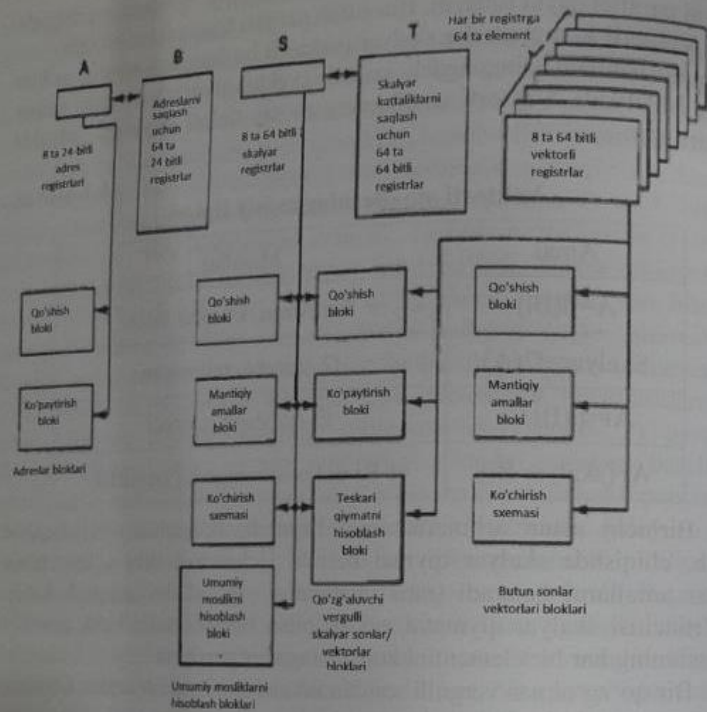
Amal	Misollar
$A=f_1(B_i)$	$f_1$ -cosinus, kvadrat ildiz
Skalyar= $f_2(A)$	$f_2$ -summa, minimum
$A_i=f_i(B_i, C)$	$f_3$ -qo'shish, ayirish
$A_i$ (skalyar $B_i$ )	$f$ - $B_i$ ni konstantaga ko'paytirish

Birinchi ustun arifmetik amal bajaradi. Ikkinchisi vektor qabul qilib, chiqishda skalyar qiymat beradi. Uchinchisi ikki vektor bilan binar amallarni bajaradi (mos elementlar yig'indisini aniqlash kabi). To'rtinchisi skalyar qiymatni vektor bilan birlashtiradi. Tipik misol – massivning har bir elementini konstantaga ko'paytirish.

Bir qo'zg'almas vergulli sonidan ikkinchisini ayirish uchun ularning eksponentialari bir xil bo'lishi kerak.

### 6.7. Cray-1 vektorli superkompyuteri.

Superkompyuterlar tarkibida har biri alohida hisoblashni bajaruvchi, parallel ishlaydigan arifmetik-mantiqiy qurilmalar bo'ladi. Cray-1 kompyuteri RISC tipli oddiy arxitekturaga ega. Bu mashina registr tipli, ko'p komandalari 16 bitli. Registrnlarning 5 tipi mavjud.



6.3-rasm. Cray-1 hisoblash mashinasining registrlari va funksional bloklari

8 ta 24 bitli registrlar (A) xotiraga murojaat uchun mo'ljallangan. 64 ta 24 bitli registrlar (B) A registrlarning qiymatini kerakmas bo'lgunicha xotiraga yubormasdan o'zida saqlaydi. 8 ta 64 bitli (S) registrlar skalyar qiymatlarni (butun sonlar, Qo'zg'almas vergulli son) saqlashga mo'ljallangan. Bu registrlarning qiymatlaridan operand sifatida butun sonlar bilan ishlaganda, Qo'zg'almas vergulli son bilan ishlaganda foydalanish mumkin.

64 ta 64 bitli registrlar (T) S dagi registrlar qiymatlarini (xotiraga yubormasdan) saqlab turuvchi registrlardir.

Qiziqarli registrlar to'plami bu 8 ta 64 bitli registrlardir. Bunday har bir registr 64 elementi Qo'zg'almas vergulli son bilan ishlovchi vektorga ega. Bitta 16 bitli komandada ikkita vektorni qo'shish, ayirish, ko'paytirish mumkin. Bo'lish amali bajarilmaydi, lekin teskari qiymatni hisoblash mumkin. Vektorli registrlar xotiradan yuklanishi mumkin,

xotiraga saqlanishi mumkin, lekin bu ko'p vaqt talab qilgani sababli bunday amallar sonini kamaytirishga harakat qilish lozim. Barcha vektorli amallarda registrli operandlar ishlatiladi.

Superkompyuterlarda hamma holatda ham operandlar registrlarda joylashishi shart emas. Masalan CDC Cyber205 mashinasi vektorlar bilan ishni xotirada olib boradi. Bunday uslubda ixtiyoriy uzunlikdagi vektorlar bilan ishlash imkoniyati mavjud, lekin u mashinaning tezigi pasayishiga olib keladi.

Cray-1 12 ta turli funksional bloklardan tashkil topgan. Ularning 2 ta si 24 bitli adreslar bilan arifmetik amallar bajarishga mo'ljallangan. 4 tasi 64 bitli butun sonlar bilan ishlashga mo'ljallangan. Cray-1 butun sonli ko'paytirish amalini bajaruvchi blok bilan ta'minlanmagan. Qolgan 6 ta konveyerlashgan blok vektorlar bilan ishlaydi.

### 6.8. Xotiraga bir xil bo'lmagan ruxsatlilik tizimlari (NUMA)

UMA (Uniform Memory Access – xotiraga bir turli murojaat). Bir shinali UMA tipli multiprotsessorlar razmeri bir necha 10 protsessor bilan cheklangan bo'ladi. Ulanadigan protsessorlar sonini 100 ta va undan ortiq qilish uchun boshqacha yondashuv talab etiladi. Bunda barcha xotira modullari bir xil murojaat vaqtiga ega bo'lishi kerak. Bunday vaziyatda NUMA (NonUniform Memory Access – xotiraga bir xil bo'lmagan murojaat) qo'l keladi. UMA multiprotsessorlari kabi u barcha jarayonlar uchun yagona adres muhitini ta'minlaydi, lekin ulardan farqli ravishda lokal xotira modullariga murojaat tezroq amalga oshiriladi. Bundan kelib chiqadiki, barcha UMA dasturlar NUMA mashinalarida hech qanday o'zgarishsiz ishlaydi, lekin takt chastotasi bir xil bo'lgan NUMA ga qaraganda UMA da unumdorlik yuqori bo'ladi.

NUMA mashinalari boshqalardan farqlanuvchi 3 ta asosiy xarakteristikaga ega

- 1) Barcha protsessorlarga ko'rinadigan yagona adres muhiti mavjud
- 2) Tashqi xotiraga murojaat qilishda LOAD va STORE ni ishlatish mumkin
- 3) Tashqi xotiraga murojaat vaqti lokal xotiraga qaraganda sekinroq kechadi

Agar tashqi xotiraga murojaat vaqti yashirin bo'lmasa (kesh-xotira yo'qligi sababli), bunday tizim NC-NUMA (No Caching NUMA – keshsiz NUMA) deyiladi. Agar kelishilgan kesh mavjud bo'lsa, bu tizim CC-NUMA (Coherent Cache NUMA – kelishilgan keshli NUMA)

deyiladi. Ba'zi hollarda bu tizimni apparatli DSM (Distributed Shared Memory – tarqoqlangan umumfoydalaniladigan xotira) deyiladi, chunki uning ishlashi DSM dastur kabidir.

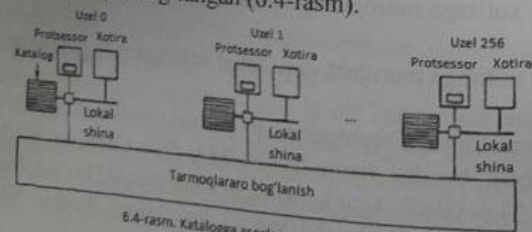
NC-NUMA mashinada kesh-xotiraning yo'qligidan xotiraning kelishilganligi qat'iy ta'minlanganligi kelib chiqadi. Har bir so'z xotirada qat'iy belgilangan pozitsiyada joylashadi shu sababli eski ma'lumotlarni o'chib ketishidan xavfsiramaslik mumkin.

### 6.9. CC-NUMA multiprotsessorlari

Yuqoridagi slaydda keltirilgan multiprotsessorlar kesh-xotira bo'lmaganligi sababli kengaytirish imkoni juda kichik. Keshsiz har gal xotiraning o'ziga so'z o'qish uchun murojaat qilish unumdorlikni pasayishiga olib keladi. Kesh xotiraning qo'shilishi bilan keshni mutanosiblashtirish usulini ham qo'shish kerak bo'ladi. Buning bir usuli – sistema shinasini kuzatish. Texnik jihatdan buni tashkil qilish oson, hatto to'rtta kuzatuvchi shina va yuqori tezlikli 16 bayt kenglikdagi koordinatali kommutator bilan ham. Katta hajmli multiprotsessor tashkil qilish uchun boshqacha yondashuv talab etiladi.

CC-NUMA tipidagi katta multiprotsessorlarni tashkil qilishning keng tarqalgan shakli katalogga asoslangan multiprotsessoridir. Bunda asosiy g'oyasi – kesh xotiraning har bir satri qayerda joylashadi va holati qandayligini aniqlovchi ma'lumotlar bazasini saqlashdir. Ma'lumotlar bazasiga murojaat har komandada mavjud ekan, ma'lumotlar bazasi zaprosga javobni qisqa vaqtda beruvchi yuqori tezlikli maxsus qurilmada joylashishi zarur bo'ladi.

Katalogga asoslangan multiprotsessorni tushunish uchun misol sifatida 256 ta uzeldan tashkil topgan, har bir uzelda bir protsessor+16mb operativ xotira bo'lgan, protsessor bilan lokal shina orqali bog'langan tizimni qaraymiz. Xotira hajmi  $2^{32}$  baytni tashkil qiladi. U har biri 64 baytli bo'lgan  $2^{26}$  ta satrga bo'lingan. Xotira uzellarga bo'lingan: 0-uzelda 0-16 MB, 1-uzelda 16-32 MB, va h.k. uzellar tarmoq orqali bog'langan (6.4-rasm).

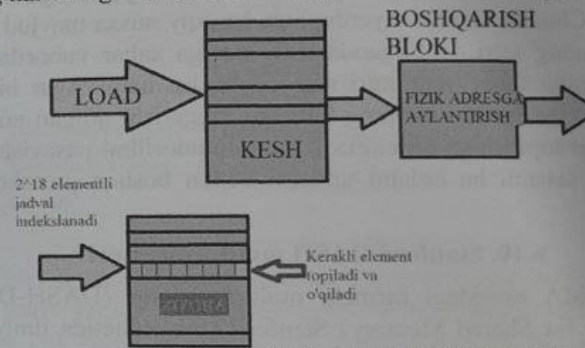


6.4-rasm. Katalogga asoslangan multiprotsessor

Tarmoq esa tashkil qilinishi bo'yicha panjara, giperkub va/yoki boshqa shaklda bo'lishi mumkin.

Katalog qanday ishlashini o'rganish uchun LOAD komandasi ishini kuzatamiz. U keshlangan satrga murojaat qiladi. Dastlab komandani yuborgan protsessor xotirani boshqarish blokiga yuboradi, u esa uni fizik adresga aylantiradi, mas  $0x24000108$ . Xotirani boshqarish bloki bu adresni uch qismga bo'ladi.

10 lik sanoq sistemasida ko'rsatilgan bu uch qism 36-uzel, 4-satr, 8-kochish. Xotirani boshqarish bloki murojaat qilinayotgan xotira so'zi 36-uzelda ekanligidan, so'rovni shu uzelda yuboradi, keshda 4-satr bor-yo'qligini tekshiradi, bor bo'lsa aynan qayerda ekanligini aniqlaydi. So'rov 36-uzelga yetib kelgach katalogning texnik ta'minotiga yo'naltiriladi. Texnik ta'minot qismi 218 elementli jadvalni indekslaydi (keshning har satriga bitta elementdan), va 4-elementni o'qiydi.



6.5-rasm. LOAD komandasining ishlashi

Kataloglar qancha xotirani egallaydi? Har bir uzeldan 16 MB operativ xotira va bu operativ xotirani kuzatish uchun  $2^{18}$  ta 9 bitli elementlardan iborat. Demak,  $9 \cdot 2^{18}$  bit, bu esa 16 MB ning 1.76% ni tashkil qiladi. Agar kesh xotiraning satr uzunligi 32 bayt bo'lgan taqdirda ham bu ko'rsatkich 4% dan oshmaydi. Agar kesh xotiraning satr uzunligi 128 bayt bo'lsa, bu ko'rsatkich yana kamayib, 1%dan past bo'ladi.

Buning kamchiligi shundaki, satrlar faqat bitta uzelda keshlanishi mumkin. Satrlarni bir necha uzellarda keshlash uchun ularni qandaydir topish usulini bilish lozim bo'ladi. Turli variantlar mavjud. Birinchisi – maydon katalogining har bir elementiga boshqa uzellarni aniqlashni topshirish. Bu har bir satrni bir necha kesh xotira bloklarida saqlash imkonini beradi. Yana bir imkoniyat uzeldan raqamini bitli shaklga o'tkazish, har uzeldan bir bit. Har bir 64 baytli kesh xotira satri uchun

256 bitni saqlovchi katalog 50%dan ortiq ishlab chiqarishga kirmaydigan xarajatlarni (непроизводительные затраты) talab qiladi. Yana bir varianti – katalogning har bir elementida 8 bitli maydon saqlash va bu maydonni bog'langan ro'yxatning sarlavhasi sifatida ishlatish. Bu esa kesh xotiraning barcha satrlari nusxalarini bog'laydi. Bunday strategiyada bog'langan ro'yxatni ko'rsatkichi uchun har bir uzelda qo'shimcha muhit talab etiladi. Bundan tashqari barcha nusxalarni topish zarurati bo'lganda bog'langan ro'yxatni tekshirib chiqish lozim bo'ladi. Bu yondashuvlarning har biri o'z yutuq va kamchiliklariga ega. Amaliyotda har uchala yondashuv ham ishlatilaveradi.

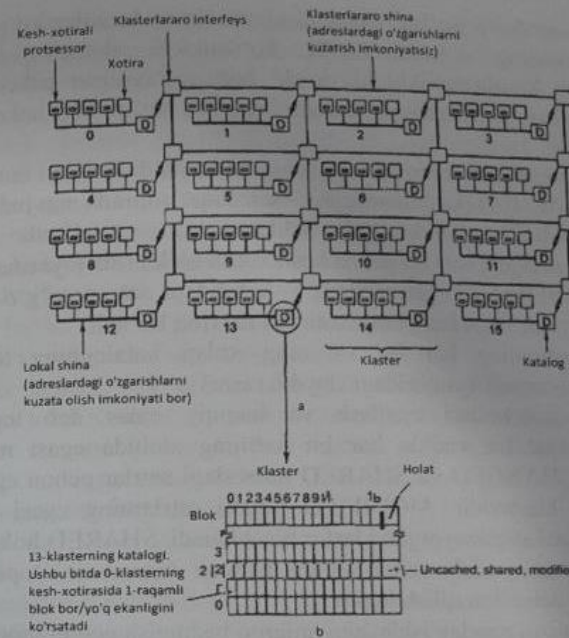
Yana bir muammo joriy xotira holati yangilanganmi yoki yo'qmi? O'zgarishga uchramagan kesh xotira satrini o'qish kerak bo'lganda bunday so'rov kesh xotiraning kerakli satri joylashgan uzeligga yuboriladi. Chunki faqat shu yerdaginga haqiqiy nusxa mavjud bo'ladi. Kesh xotiraning satri o'zgarganda joriy uzelda xabar yuboriladi, agar kesh xotiraning faqat bitta satri mavjud bo'lsa ham. Agar bir necha nusxalar mavjud bo'lsa, ularning bittasini o'zgarishi qolgan nusxalarni haqiqiy emas topilishiga olib keladi. Bu unumdorlikni pasayishiga olib keladi. Shu sababli bu holatni tuzatish uchun boshqa protokol kerak bo'ladi.

### 6.10. Stanford DASH multiprotsessori

CC-NUMA asosidagi birinchi multiprotsessor (DASH-Directory Architecture for Shared Memory) Stenford Universitetida ilmiy loyiha sifatida ishlab chiqilgan (6.6-rasm).

U 16 ta klaster, 4 ta protsessor, 16MB global xotira va kiritish-chiqarish qurilmalaridan tashkil topgan. Har bir protsessor faqat o'zining lokal shinasini kuzatib turadi. Lokal mutanosiblik kuzatish yordamida qo'llab-quvvatlanadi. Global moslashuv uchun esa boshqa mexanizm zarur, chunki global kuzatuv imkoniyati mavjud emas. 612-sahifa

Bu tizimda adres muhitining to'liq hajmi 256 mb. Adres muhiti har biri 16 mb bo'lgan 16 bo'lakka bo'lingan. Kesh-xotiraning bir satri 16 baytga teng. Ma'lumot uzatish 16 baytli (bir satri) tartibda bo'ladi. Har bir klaster 1M satr saqlaydi.



6.6-rasm. DASH arxitekturasi (a), DASH katalogi (b).

Har bir klaster o'zining nazorat klasteriga ega bo'lib, u qaysi klasterlar joriy vaqtda nusxaga egaligini kuzatib boradi. Har bir klaster 1M satrga ega ekan, katalogda 1M element mavjud bo'ladi. Har bir element har bir klasterda bir bitga ega bo'ladi. Bu bit o'zi joylashgan satrni kesh xotirada bor yoki yo'qligini aniqlab beradi. Bundan tashqari element 2 bitli maydonga ega bo'lib, u satr holati haqida ma'lumot beradi.

Har biri 18 bitdan bo'lgan 1M element katalog hajmini 2MB dan katta ekanligini bildiradi. 16 klaster mavjud bo'lganda katalogning umumiy hajmi 36 MB ni tashkil etadi. Bu esa jami 256 MB hajmning 14%ni tashkil etadi. Klasterdagi protsessorlar soni oshadigan bo'lsa ham katalogning xotira hajmi o'zgarmaydi. Har bir klasterda ko'p sonli protsessorlarning bo'lishi katalog xotirasi qiymatini kamaytirishga olib keladi.

DASH dagi har bir klaster boshqa klasterlar bilan axborot almashinuvchi interfeys bilan ta'minlangan. Interfyeslar klasterlararo kanal orqali to'g'ri burchakli panjara shaklida bog'langan. Tizimda

qancha klaster ko'p bo'lsa, shuncha klasterlararo kanallar ko'payadi, demak tizimning o'tkazuvchanlik ko'rsatkichi shuncha oshadi. Klasterlararo kanalning ikki ko'rinishi bor: so'raydigan paketlar va javob paketlari. Klasterlararo kanalni kuzatib borishning imkoniyati mavjud emas.

Kesh xotiraning har bir satri quyidagi 3 ta xolatda bo'lishi mumkin:

- 1) UNCACHED (keshlanmagan) – satr faqat xotirada mavjud
- 2) SHARED (hamkorlikda ishlanuvchi) – xotirada yangi ma'lumotlar mavjud; satr bir necha kesh xotira bloklarida joylashadi
- 3) MODIFIED (o'zgartirilgan) – xotiradagi satr noto'g'ri (yoki eskirgan); bu satr faqat bitta kesh xotirada mavjud bo'ladi

Kesh xotiraning har bir satrining xolati katalogning tegishli elementida ko'rinadi. (yuqoridagi slayd b rasm)

DASH protokollari egallash va haqiqiy emas deb topishga asoslangan. Har bir vaqtda har bir satrning alohida egasi mavjud bo'ladi. UNCHANGED va SHARED holatidagi satrlar uchun ega shu satrning o'z klasteridir. MODIFIED holatli satrlarning egasi ushbu satrning nusxasi saqlanayotgan klaster hisoblanadi. SHARED holatidagi satrga ma'lumot yozish uchun uning hamma nusxalarini topish va haqiqiy emas deb e'lon qilish lozim.

Bu mexanizm qanday ishlashini aniqroq tushunish uchun protsessor xotiradan so'zni qanday o'qishini qaraymiz. Dastlab u o'zining kesh xotirasini tekshiradi. U yerda so'z topilmasa, klasterning lokal shinasiga qaysi klasterdagi protsessor shu so'z mavjudligi so'rovi yuboriladi. Agar so'z mavjud bo'lsa, satr o'sh joydan kerakli joyga ko'chiriladi. Agar satr SHARED xolatida bo'lsa, uning nusxasi hosil qilinadi. Agar satr MODIFIED xolatida bo'lsa katalogni ushbu satr SHARED xolatiga o'tganligi haqida ogohlantiriladi. Har qanday xolatda ham so'z qaysidir kesh xotiradan o'qiladi, lekin bu kataloglarning bitli ko'rinishiga ta'sir qilmaydi (chunki katalog har klasterga 1 bit saqlaydi, har protsessorga 1 bit emas)

Agar kerakli satr joriy klasterning hech bir keshida mavjud bo'lmasa, so'rov paketi shu satrni saqlab turgan joriy klasterga yuboriladi. Bu klaster xotira adresining oldingi 4 ta biti bilan aniqlanadi. Texnik qism joriy klasterdagi barcha jadvallarni tekshiradi va satr qanday xolatda ekanligini aniqlaydi. Agar u UNCACHED yoki SHARED xolatida bo'lsa, uni global xotiradan chaqirib, so'ralgan klasterga jo'natadi. Keyin texnik qism o'zini yangilaydi (обновление), joriy satrni kesh xotirada saqlanganligini belgilab qo'yadi.

Agar kerakli satr MODIFIED xolatida bo'lsa, texnik qism ushbu satr saqlanayotgan klasterni topadi, shu joyga so'rov yuboradi. Bu klaster uni kerakli klasterga jo'natadi va o'zidagi nusxani SHARED deb belgilab qo'yadi, chunki bu satr bittadan ortiq klasterda mavjud bo'ldi. Shuningdek xotirani yangilash uchun va satr xolatini SHAREDga o'zgartirish uchun satr nusxasini joriy klasterga yuboradi.

Yozish esa boshqacha uslubda kechadi. Yozishdan oldin protsessor ushbu satr kesh xotirada yagona nusxada ekanligini tekshiradi. Agar protsessorning keshida satr avvaldan mavjud bo'lsa va MODIFIED xolatida bo'lsa yozish mumkin. Agar satr mavjud bo'lsa va SHARED xolatida bo'lsa, dastlab shu satrning boshqa hamma nusxalarini haqiqiy emas deb topish uchun signal yuboriladi.

Agar kerakli satr protsessorning keshida mavjud bo'lmasa, bu protsessor lokal shinaga yondosh protsessorlarda shu satrni mavjudligini tekshirish signalini yuboradi. Satr topilsa, u bir keshdan ikkinchisiga ko'chiriladi. Agar xolati SHARED bo'lsa, boshqa barcha nusxalar haqiqiy emas deb belgilanadi.

Agar satr yana qayerdadir mavjud bo'lsa, paket joriy katalogga yuboriladi. 3 xil variant bo'lishi mumkin: 1) agar satr UNCACHED bo'lsa, xolati MODIFIED ga o'zgartiriladi va so'rayotgan protsessorga yuboriladi. 2) Agar xolati SHARED bo'lsa barcha nusxalari haqiqiy emas deb belgilanadi, va MODIFIED qilib, so'rayotgan protsessorga uzatiladi. 3) Agar satr MODIFIED bo'lsa, so'rov shu satrni saqlab turgan klasterga yuboriladi. Klaster so'rovni bajaradi va o'zining nusxasini haqiqiy emas deb belgilaydi.

### 6.11. COMA multiprotsessorlari

NUMA va CC-NUMA mashinalari asosiy xotiraga murojaat vaqti kattaligi bilan xarakterlanadi. CC-NUMA da kesh xotira ishlatilishi hisobiga bu birmuncha bartaraf etilgan. Lekin, so'ralgan ma'lumotning hajmi keshning hajmidan kattaroq bo'lsa, bu muammo baribir qaytadi.

Har bir asosiy xotirani kesh sifatida qarash bilan bu muammoni hal qiladigan protsessor mavjud va COMA (Cache Only Memory Access) deyiladi. Adres muhiti satrlarga bo'linadi. Ular esa zarur bo'lganda tizimda ko'chib yura oladi. Xotira bloklari alohida mashinalarga ega emas. Satrlarni kerakli vaqtda chaqiradigan bunday xotira **attraction memory** deyiladi. Asosiy xotirani katta kesh xotira sifatida ishlatish muvaffaqiyatli murojaatlar chastotasini oshiradi, demakki unumdorlikni ham.



Bunda 2 ta muammo paydo bo'ladi: 1) Kesh xotira satrlari qanday joylashadi? 2) Agar satr xotiradan o'chirilsa va u oxirgi nusxa bo'lsa, keyin nima bo'ladi?

Ikkinchi muammoni hal qilishda mashina asosiy katalogga qaytib, satrning boshqa nusxalari borligini tekshiradi. Yo'q bo'lsa, nusxasini hosil qilib, saqlab qo'yadi, yana bir yondashuv – kesh xotiraning har bir satrini bir nusxasini asosiy deb belgilab qo'yish va hech qachon o'chirmaslik. Bunda katalogni tekshirish talab qilinmaydi.

## 7. Parallel vektorli tizimlar (PVP), klasterli tizimlar

### 7.1. Parallel ishlash turlari

Masalani parallel yechishning usullari ko'p. Masalan, axborotni o'qish-yozish amalini xotiraga murojaat qilish imkoni bo'lgan qurilmalarning o'zi, markaziy protsessor ning ishtirokisiz bajaradi, xususan, asosiy xotira va tashqi xotira interfeysi DMA (Direct Memory Access – markaziy protsessor ni aylanib o'tib, xotiraga bevosita murojaat texnologiyasi) kontrolleri yordamida markaziy protsessor ishi bilan parallel bajariladi. Tashqi xotiradan asosiy xotiraga axborot uzatishni markaziy protsessor tashkil qiladi, DMAga egishli komandani beradi, DMA bajaradi. DMAga komanda bergandan keyin markaziy protsessor bu jarayonga aralashmaydi, boshqa hisoblashlarni bajaradi. Bu vaqtda DMA ko'chirishni mustaqil bajaradi, ishni yakunlab, bu haqda uzilish signali bilan markaziy protsessor ga xabar beradi. Xabarni olgan markaziy protsessor ko'chirilgan axborot bilan tegishli amallarni bajarishni boshlaydi.

Bu misolda parallellikning 2 ta asosiy xususiyati namoyon bo'ladi.

**1-dan**, bitta masalada (jarayonda) bu usulni qo'llash uchun bu masala (jarayon) mashinaning turli qurilmalarida parallel bajarilishi mumkin bo'lgan qismlardan tashkil topishi kerak. Bu misolda masala (jarayon) ning qismlari sifatida markaziy protsessor ning hisoblash jarayonlari, tashqi xotiradan asosiy xotiraga ko'chirish bor. **2-dan** parallel qayta ishlashni tashkil qilish va nazorat qilish uskunalari mavjud bo'lishi kerak. Yuqoridagi misolda parallellikni markaziy protsessor tashkil qiladi, markaziy protsessor va DMA ishini koordinatsiya qilish DMA ning ishini yakunlaganlik haqidagi uzilish signali bilan bajariladi. Bu signalga javob qilib, markaziy protsessor hisoblashlarni boshlaydi.

Bu parallellikka eng sodda misol, unda 2 tagina masala (jarayon) bajariladi. Umumiy holatda ko'p sonli hisoblashlarni ko'p sonli parallel

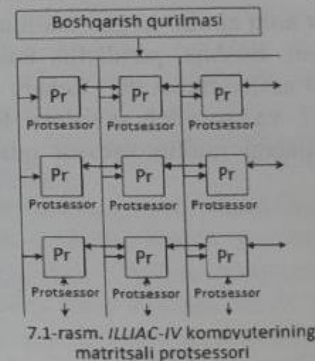
bajariladigan jarayonlarga bo'lish mumkin. Buni amalga oshirish uchun bir necha apparat tizimlari zarur bo'ladi.

### 7.2. Parallel qayta ishlash tizimlari klassifikatsiyasi

Yagona protsessorli kompyuter tizimi *bir potok komanda* – *bir potok ma'lumot* (Single Instruction stream, Single Data stream SISD) deyiladi. Protsessor bajarayotgan dastur komandalar potoki hisoblanadi, qayta ishlayotgan ma'lumotlar ketma-ketligi esa ma'lumotlar potoki deyiladi. *Bor potok komanda ko'plab protsessorlarda ishlanadigan tizim* ham mavjud, bu arxitektura SIMD (Single Instruction stream, Multiple Data stream) deyiladi. Ko'p potokli ma'lumotlar bilan ko'p protsessor yordamida tashkil qilingan arxitektura MIMD (Multiple Instruction stream, Multiple Data stream) deyiladi. Bunday tizimlarda yagona ma'lumotlar potoki turli ishlarni bajaradigan turli protsessorlarda qayta ishlanadi (bajariladi). Bunday arxitektura amaliyotda kam qo'llaniladi.

### 7.3. Ma'lumotlarni matritsali qayta ishlash

SIMD ni matritsali qayta ishlash deyiladi. 1970-yillarda bu arxitekturada ILLIAC-IV ishlab chiqilgan (7.1-rasm).



Ikki o'lchamli protsessorlar matritsasi markaziy protsessor dan kelgan komandani barcha protsessorlarda parallel bajaradi. Har bir protsessor yonidagi protsessor bilan axborot almashadi. Massiv ikki o'lchamli ma'lumotlarni qayta ishlaydi. Har bir element (protsessor+xotira elementi) registrlarga va xotiraga, shuningdek qo'shni element bilan axborot almashish uchun tarmoq registriga ega bo'lishi zarur. Markaziy protsessor hamma elementga tarmoq registrlari qiymatini bir daraja chap, o'ng, past, yuqoriga o'tkazish komandasini yuboradi. Har bir element axborotni qayta ishlash uchun o'z arifmetik-

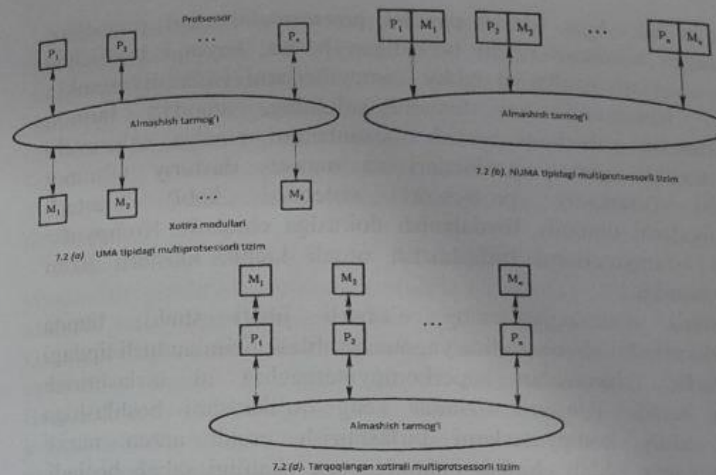
mantiqiy qurilmasiga ega bo'lishi lozim. Hisoblashlar bajarilgandan keyin holat biti 1 ga tenglanadi. Bu esa hisoblashlar yakunlanganini bildiradi.

Muammoli vaziyat: bir necha kuchli protsessor ishlatgan ma'qulmi yoki bir nechta yuzlagan kuchsiz protsessor ishlatgan ma'qulmi? 1-yondasuvni tashkil etishga ILLIAC-IV superkompyuteri misol bo'ladi. U 46 razryadli komandalarni qayta ishlovchi 64 ta protsessoridan tashkil topgan. 2-yondashuvga 80-yillarda ishlatilgan matritsali protsessorlar misol bo'ladi. CM-2 tizimida 65536 ( $2^{16}$ ) ta protsessor joylashtirish mumkin, lekin ularning razryadliliigi 1 ga teng bo'lishi lozim bo'ladi.

Matritsali protsessorlarni qo'llash sohasi juda kam. Asosan matritsalar va vektorlar bilan bog'liq masalalar (jarayonlar) ni qayta ishlashga mo'ljallangan. Bunday masalalarni superkompyuterlar ham bajara oladi. Matritsali protsessorning vektorlidan farqi, vektorlarda yuqori unumdorlik faol konveyerlash orqali hal qilinadi, matritsali esa parallellik darajasini oshirish bilan. Ta'kidlash lozimki, bu ikki texnologiya ham unumdorlikni keskin ravishda oshira olmaydi.

#### 7.4. Umummo'ljallangan multiprotsessor tizimlari arxitekturasi

Matritsali tizimlar aniq ajralib turgan ma'lumotlar parallelligi bilan ishlashga mo'ljallangan. Boshqa, parallellik aniq ajralib ko'rinmagan tizimlar uchun MIMD arxitekturasi qo'l keladi, chunki bunda ko'plab protsessorlar mustaqil va parallel ravishda turli dastur qismlarni bajaradilar. MIMD tipidagi multiprotsessor qurishning 3 ta sxemasi (7.2-rasm)



7.2-rasm. MIMD tipidagi multiprotsessor qurishning 3 ta sxemasi

7.2 (a) rasmda eng oddiy texnologiya keltirilgan. Unda protsessorlar, xotira modullari, va ularni bog'lash uchun kommunikatsion tarmoqlar mavjud. Xotiraga murojaat qilishda bu tarmoq sekinlashishga olib keladi. Agar bu sekinlashish xotiraga murojaatning barcha turlarida bir xil bo'lsa, umumiy xotirali tizim (UMA-Uniform Memory Access), yoki xotiraga bir xil murojaatli multiprotsessorli tizim deyiladi.

Yuqori tezlikni olish uchun protsessorlarni har birini xotira moduli bilan bog'lash mumkin. (7.2 d). Bunda har bir protsessor nafaqat o'zining lokal xotirasiga, balki boshqa protsessorlar xotirasiga ham murojaat qila oladi. Bunday tizim xotiraga bir xil bo'lmagan murojaatli tizim (NUMA-Non Uniform Memory Access) deyiladi.

7.2 (b) va 7.2 (d) dagi sxemalarda global xotira ishlatilishi ko'rsatilgan. Bunda har bir xotira moduliga har bir protsessor murojaat qila oladi. 12.4 rasmda har bir xotira moduli unga bevosita ulangan protsessorlarga tegishli hisoblanadi. Hech bir protsessor xotiraga o'zicha murojaat qila olmaydi. Buning uchun xotirani tegishli qismini egallab turgan protsessor bilan muloqot qilishi zarur bo'ladi.

#### 7.5. Klasterli tizimlar

Klasterli texnologiyalar massiv-parallel protsessorlar (massiv-parallel tizimlar) ning arxitekturasiga asoslangan g'oyaning mantiqiy

davomi bo'ldi. Agar Massiv-parallel protsessorlar dagi protsessor yakunlangan hisoblash tizimi bo'ladigan bo'lsa, keyingi bosqichda bunday hisoblash uzellarida oddiy kompyuterlarni ishlatish mumkin emasmi? Kommunikatsion texnologiyalarning, standart tarmoq protokollaridan xabarlarini uzatish mexanizmini amalga oshiruvchi yuqori tezlikli tarmoq qurilmalari va maxsus dasturiy ta'minot yaratilishi (markaziy protsessorli sistemasi kabi) klasterli texnologiyalarni umumiy foydalanish doirasiga chiqardi. Kompyuter sinfidagi kompyuterlarni birlashtirish orqali kichik klasterli tizim yaratish mumkin.

Klasterli texnologiyalarning e'tiborli jihati shuki, bunda unumdorlikni oshirish maqsadida yagona hisoblash tizimiga turli tipdagi kompyuterlar (shaxsiydan superkompyutergacha) ni birlashtirish imkonini beradi. Klasterli tizimlar keng qo'llanishni boshlashiga ularning oddiy kompyuterlarni birlashtirish orqali arzon narxli superkompyuter ishlab chiqish imkoniga ega ekanligi sabab bo'ladi. Birinchi bunday tizim COCOA bo'lib, 25 ta ikki protsessorli personal kompyuterlardan 48 protsessorli Cray T3D dan kuchliroq tizim ishlab chiqilgan. Ularning narxi 100.000\$ va 2.000.000\$

Lekin bunday tizimlarning ekvivalentligi mos emas. Tarqoqlangan xotirali tizimlarning unumdorligi aloqa muhitining o'tkazuvchanligiga to'g'ridan-to'g'ri bog'liq bo'ladi. Aloqa muhiti ikkita parametri bilan ko'zga tashlanadi: 1) *latentlik* – xabar jo'natishdagi vaqt to'xtalishi. 2) *o'tkazuvchanlikqobiliyati* – axborotni uzatish tezligi. Cray T3D kompyuteri uchun bu ko'rsatkichlar 1 mks va 480 Mb/s, axborot muhiti sifatida Fast Ethernet tarmog'i ishlatilgan klasterda 100 mks va 10 Mbit/s ga teng. Superkompyuterlarning narxi nima uchun balandligini shundan bilish mumkin. Shu sababli yuqoridagi ta'kidlanganga o'xshash klasterlarda yechiadigan masalalar juda oz sonlidir.

Qisqa tarif beradigan bo'lsak, *klaster bu* – yagona hisoblash resursi sifatida ishlatish uchun mo'ljallangan va bog'langan kompyuterlarning to'plamidir. Klaster tizimi alohida kompyuterlarga qaraganda yaxshidir. **Birinchi**dan masalalarni paketli qayta ishlovchi ko'plab dispetcherlik dasturlari mavjud. Bu dasturlar masalani klasterga umumiy qilib yuboradi. Bu dasturlar masalani bo'sh turgan mashinalar orasida mustaqil taqsimlaydi yoki bo'sh mashina bo'lmaganda masalani buferlaydi, bu esa kompyuterlarning bir tekis va effektiv yuklanishini ta'minlaydi. **Ikkinchi**dan bitta masalani yechish uchun bir necha kompyuterlarning resurslaridan foydalaniladi.

Klasterni hosil qilish uchun asosan bir protsessorli shaxsiy kompyuterlar yoki 2-4-protsessorli SMP-serverlar (simmetrik multiprotsessorli tizimlar) ishlatiladi. Bunda mashinalar arxitekturasi va tarkibiga hech qanday cheklovlar qo'yilmaydi. Har bir mashina o'z operatsion tizimida ishlashi mumkin. Ko'pincha Linux, Windows NT, Solaris kabi standart operatsion tizimlar ishlatiladi. Agar mashinalar har xil operatsion tizimlarda ishlasa bunda klaster *heterogen klaster* deyiladi.

Klasterlarni hosil qilishda ikkita yondashuv mavjud. Birinchi yondashuv kichik klasterlarni yaratishda qo'llaniladi. Klasterga har biri alohida ishlaydigan kompyuterlar bog'lanadi masalan o'quv sinfi kompyuterlari yoki ishchi stansiya laboratoriyalari kabi. Ikkinchi yondashuv kuchli hisoblash tizimini ishlab chiqishda qo'llaniladi. Bunda kompyuterlarning tizim bloklari maxsus joylarda yonma-yon joylashtiriladi, ularni boshqarish uchun esa xost-kompyuter deb nomlanuvchi bir yoki bir necha kompyuterlar ajratiladi. Bu yondashuvda har bir kompyuterni monitor, klaviatura kabi qurilmalar bilan ta'minlash shart emas, shu sababli narx arzonlashadi.

Kompyuterlarni klasterlarga birlashtirishning ko'plab texnologiyalari ishlab chiqilgan. Eng ko'p qo'llaniladigani Fast Ethernet. Chunki uni ishlatish sodda va tashkil qilish uchun harajat ko'p talab qilinmaydi. Lekin, bu yutuq bilan birgalikda unumdorligi past ekanligini ham ta'kidlash joiz. Bu klaster 10 MB/s o'tkazuvchanlikka ega, operativ xotira qurilmasi esa 250 MB/s o'tkazuvchanlikka ega.

Kommunikatsion amallar ulushi ko'p bo'lgan chiziqli algebra masalalarini ko'pprotsessorli tizimlarda yechishga mo'ljallangan ScaLAPACK dasturiy paketi ishlab chiqaruvchilari ko'pprotsessorli tizimga quyidagi talablarni shakllantirishgan: **“Tizimda (MB/s tezlikda o'lchanadigan) ikkita mashina orasidagi almashinuv tezligi (Mflopsda o'lchanadigan Floating-point Operations Per Second) bitta mashinaning tezligidan 1/10 miqdordan kam bo'lmasligi kerak”**. Masalan, Pentium III 500 MHz (500 Mflops) bazasida tashkil qilingan klasterda Fast Ethernet talab qilinadigan tezlikning faqat 1/5 qismini beradi.

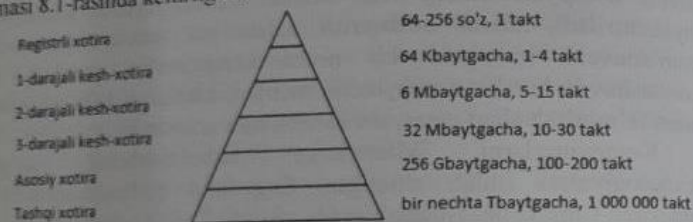
Ba'zi xollarda yuqori tezlikli maxsus klasterlardan foydalanish mumkin, masalan Scali Computer firmasining SCI klasteri (taxm.100MB/s) yoki MiryNet (taxm.120MB/s) kabi. Shuningdek klaster texnologiyasini ishlab chiqishga SUN, HP kabi firmalar ham qiziqish bildirgan va ishlab chiqarmoqda.

## 8. Xotira ierarxiyasi, asosiy xotira, sinxron va asinxron xotira qurilmalari. Operativ va doimiy saqlash qurilmalari. Stekli va assotsiativ xotira

### 8.1. Xotiraning tipik ierarxiyasi

Birinchi ierarxik xotira fon Neyman prinsipli kompyuterlarda ishlatilgan. Unda 2 darajali ierarxiya bo'lgan: 1-tezkor operativ xotira, 2-sekinroq ishlovchi tashqi xotira. Dastur qismlarini bu darajalar orasida o'tkazish uchun dasturchi dastur matnida ko'rsatma berishi zarur bo'lgan.

Zamonaviy kompyuterlarda xotira ierarxiyasi birmuncha murakkablashgan: darajalar soni ko'paygan, ma'lumotlar almashish avtomatik va hisoblashlar fonida bajariladi. Xotira ierarxiyasining tipik sxemasi 8.1-rasmda keltirilgan.



8.1-rasm. Xotirani tashkil qilishning ierarxik sxemasi

Ierarxiyadagi har bir daraja o'zining murojaat vaqti, hajmi bilan xarakterlanadi va ma'lum tipdagi xotira bilan tashkil qilinadi (tezkor va qimmat narxli xotira, sekin va arzon dinamik xotira, tashqi xotira). Markaziy protsessor bilan yaqinroq bog'langan xotira darajalari qisqa murojaat vaqti va kichik hajmi bilan xarakterlanadi. Markaziy protsessoridan uzoqlashgani sayin xotiraning hajmi va murojaat vaqti oshib boradi. Ierarxiyada dastrular, ularning qismlari, ma'lumotlar xotiraning turli darajalari orasida ko'chib yuradi.

Ierarxiyaning yuqori darajasida registrlar joylashadi (registrlar fayl). Registrlar boshqa darajalarga nisbatan eng kichik murojaat vaqtiga ega. Shu sababli ularda ma'lumotlarni saqlash maqsadga muvofiq. Dastlabki kompyuterlarga qaraganda hozirgi kompyuterlar markaziy protsessorning yadrosi arxitekturada nazarda tutulganiga nisbatan ko'plab apparat registrlariga ega. Natijada bitta dasturiy registrdan foydalanuvchi alohida-alohida komandalar uchun markaziy protsessor resurslarga bog'liqlikni kamaytirish maqsadida registrlar

xotiraning turli yacheykalarini ishlatishi mumkin. Bu katta ahamiyatga ega, ayniqsa parallel hisoblashlarda.

Zamonaviy mashinalarda saqlanadigan ma'lumotlarning turiga qarab bir necha turli registr guruhlarini farqlanadi, masalan butun sonli, ratsional sonli, vektorli registrlar. Bunday farqlanish mikroarxitektura bilan to'g'rida to'g'ri bog'liq, chunki registrlarning mos guruhlarini ular bilan ishlovchi qurilmalarga yaqin joylashishi lozim.

### 8.2. Registrlarning asosiy xarakteristikalari

- 1) registr razmeri (bitlar soni)
- 2) shu tipli registrlar soni (dasturiy yoki texnik arxitekturada)
- 3) ushbu registrlar parametr sifatida ishlatuvchi markaziy protsessor komandalari to'plami

**Ierarxiyaning 2-darajasi** kesh xotira. Unda komandalar nusxalari va operativ xotiradan olingan ma'lumotlar saqlanadi. Kesh xotira adresli muhitga tegishli emas va uning tarkibini bevosita o'qish yoki o'zgartirish mumkin emas. Kesh xotirani kesh kontroller deyilvchi maxsus qurilma boshqaradi. Zamonaviy mashinalar 2 ta, 3 ta va h.k darajali kesh xotiraga ega bo'ladi. Kesh xotira markaziy protsessor kristallida yoki undan tashqarida bo'lishi mumkin. Barcha darajadagi keshlar tezkor murojaatni ta'minlash uchun markaziy protsessor kristallida joylashadi. Bundan tashqari ko'pchilik markaziy protsessorlar 1-darajali keshni ajratilgan holatda, komandalar va ma'lumotlar uchun alohida tashkil qiladi, va ularning har biri alohida tashkil qilinish xususiyatiga ega bo'ladi.

**Ierarxiyaning 3-darajasida** operativ xotira qurilmasi joylashadi. Kesh xotiradan farqli ravishda adresli xotira hisoblanadi. Unda joriy vaqtda markaziy protsessor da bajarilayotgan dasturlar va ma'lumotlar saqlanadi. Texnik jihatdan operativ xotira qurilmasi markaziy protsessoridan alohida bo'lib, asosiy platada joylashadi.

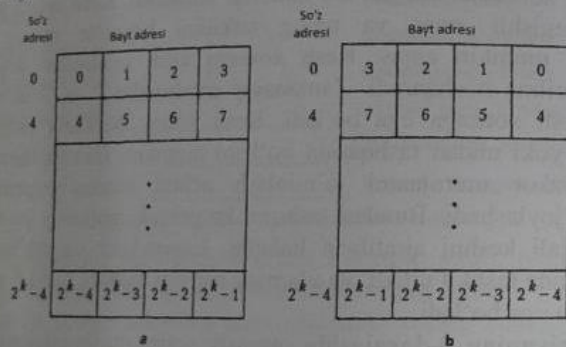
**Ierarxiyaning quyi darajasi** tashqi xotiradir. Bu xotiraning tezligi eng sekin va hajmi eng katta shaklidir. Virtual xotira tashkil qilish uchun ishlatiladi. Bundan tashqari, ierarxiyaning boshqa darajalaridan farqli ravishda holati energomustaqildir, ya'ni uning holati kompyuter o'chirilganda ham saqlanadi. Odatda bu qattiq diskdir.

Zamonaviy mashinalarda xotira darajalari orasida dastur kodlari va ma'lumotlarni o'tkazish qoidalarini belgilovchi ierarxik xotirani boshqarishning ikki xil mexanizmi mavjud. **1-mexanizm** keshlash mexanizmi deyilib, kesh xotiraning barcha darajalarini va operativ

xotirani bog'laydi. Keshlash to'liq kesh kontrollerlar vositasida bajariladi. **2-mexanizm** esa virtual xotira mexanizmi deyilib, operativ va tashqi xotiralarni bog'laydi. Tashqi ko'rinishda juda katta hajmli ko'rinishni beradi. Virtual xotirani tashkil qilish markaziy protsessor va OT hamkorligida amalga oshiriladi.

Kompyuterning maksimal xotira hajmi uning adreslash tizimidan kelib chiqadi. Masalan 16 razryadli adreslarni generatsiya qiladigan 16 razryadli kompyuter,  $2^{16}$  Kbayt (64 Mbayt) adreslaydi, 32 razryadli kompyuter  $2^{32}$  Kbayt (4Gbayt) adreslaydi, 40 razryadli kompyuter  $2^{40}$  Kbayt (1 terabayt) adreslaydi, 64 raryadli kompyuter  $2^{64}$  (18 446 terabayt) adreslaydi. Adreslanadigan xotira degani uning hajmini bildiradi.

Zamonaviy mashinalarning ko'pchilgi xotirani baytli adreslaydi (8.2-rasm).



8.2-rasm. Baytlarni va so'zlarni adreslash: baytlarning teskari tartibi (a), baytlarning to'g'ri tartibi (b)

Bu slaydda baytli adreslanadigan 32 razrydli kompyuter keltirilgan. Intel protsessorlarida baytlashning to'g'ri ketma-ketligi, Mototola-68000 da esa teskari ketma-ketlik ishlatiladi. ARM protsessor arxitekturasida har ikkala uslub ishlatiladi.

Odatda xotirani qurishda undagi ma'lumotlarni o'qish/yozish baytli amalga oshirilishi nazarda tutiladi. So'z uzunligi xotiraga bir murojaatda o'qiladigan/yoziladigan bitlar sonini bildiradi.

Xotira va markaziy protsessor orasida ma'lumot almashinish MAR (Memory Adress Register – adres registri) deb nomlanuvchi 2 ta registr yordamida bajariladi. Xotiradan ma'lumotlarni o'qish uchun markaziy protsessor dastlab adresni MAR registriga yuklaydi va R/W (read/write) ni holatini 1 ga o'zgartiradi. Bunda xotira ma'lumotlarni ma'lumotlar

liniyasiga o'tkazadi va MFC signalini hosil qiladi (ma'lumotlarni yuklanganlik signali). MFC ni o'qigan markaziy protsessor ma'lumotlarni adres liniyasidan MDR (Memory Data Register) ga o'tkazadi. Ma'lumotlarni xotiraga yozish uchun markaziy protsessor adresni MARga yuklaydi, ma'lumotlarni MDR ga yuklaydi, R/W ni 0 ga o'tkazadi, bunda yozish jarayoni borayotganligi bilinadi.

**Xotiraning tezkorligi** amalni aniqlash va uni bajarish vaqti bilan belgilanadi. Yoki, bu vaqtni **xotiraga murojaat vaqti** deyiladi. Xotira tezkorligining yana bir xarakteristikasi **xotira sikli** bo'lib, ikkita ketma-ket keladigan operatsiyalar orasidagi vaqtni bildiradi. Odatda xotira sikli murojaat vaqtdan ko'proq bo'ladi.

Tasodifiy murojaatli xotira (Random Access Memory – RAM) da ixtiyoriy adresga murojaat uchu bir xal vaqt sarflanadi.

Odatda markaziy protsessor komandalarni xotirada shakllanganiga nisbatan tez ishlaydi. Shu sababli xotiraga kirish sikli tizimning nozik joyi hisoblanadi. Siklni kamaytirishning usullaridan biri kesh xotiradan foydalanishdir.

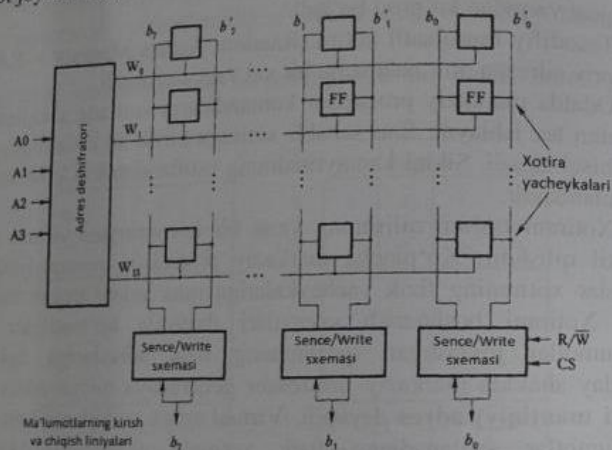
Xotirani tashkil qilishning yana bir konsepsiyasi virtual xotirani tashkil qilishdir. Ko'pincha markaziy protsessor generatsiyalaydigan adreslar xotiraning fizik yacheykalariga mos keladi degan tushuncha bor. Xotirani boshqarish sxemalari dasturda ko'rsatilgan adresni ma'lumotlar joylashgan qurilmaning fizik adreslariga aylantiradi. Bunday shaklda markaziy protsessor generatsiya qilgan adres **virtual (yoki mantiqiy) adres** deyiladi. Virtual adres muhiti ma'lum shaklda ma'lumotlar saqlanadigan fizik xotirada akslanadi. Akslantirish xotiraning maxsus boshqaruvchi sxemalar yordamida bajariladi. Ularni esa **xotirani boshqarish bloki** deyiladi. Dasturni bajarilishi jarayonida akslantirish funksiyasi tizim talablariga ko'ra o'zarib borishi mumkin.

Virtual xotira mashinaning ko'rishi mumkin bo'lgan fizik hajmini oshirishga mo'ljallangan. Virtual adres muhiti va unda joylashgan ma'lumotlar hajmi qo'llanilayotgan protsessorning adreslash imkoniyatlariga qarab katta bo'lishi mumkin. Joriy vaqtda bu adres muhitining bir qismigina fizik xotirada akslanadi. Dastur adresi real xotirada bo'lmagan ma'lumotlarni o'qishi kerak bo'lganda, xotirani boshqarish bloki tasvirlash funksiyasini o'zgartiradi va ma'lumotlarni diskdan asosiy xotiraga yuboradi. Shu sababli xotiraga har bir murojaat siklida adreslarni qayta ishlash tizimi adreslanayotgan ma'lumot mashinaning asosiy xotirasida bor-yo'qligini tekshiradi. Mavjud bo'lsa, u o'qiladi va dastur bajarilishi davom etadi. Agar yo'q bo'lsa, diskdan

xotiraga kerakli soʻz mavjud boʻlgan sahifa yuboriladi. Bu sahifa xotiradagi hozircha kerak boʻlmagan makondagi joyni egallab turadi. Bunday tartibda, sahifalarni xotiradan diskka va aksincha uzatish jarayoni tez-tez takrorlanib turishi mashina unumdorligini pasayishiga olib keladi. Agar almashtiriladigan sahifalarni tanlash rejalashtirib amalga oshirilsa, jarayonni tezlatish mumkin.

### 8.3. Xotira mikro sxemalari

Xotiraning har bir (massivlarga birlashuvchi) yacheykasida 1 bit maʼlumot joylashadi (8.3-rasm).



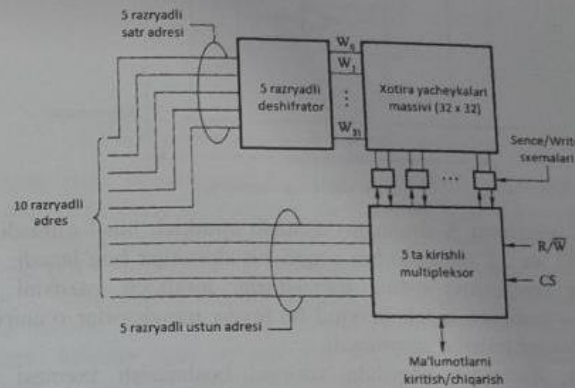
8.3-rasm. Xotira mikro sxemasida bit yacheykalarining joylashuvini tashkil qilish

Bu massivda har bir yacheykalar satri soʻzni hosil qiladi, satrdagi hamma yacheykalar soʻz shinasini deb ataluvchi umumiy liniyaga birlashadi. Ustunlardagi yacheykalar bit shinasiga birlashadi. Bit liniyasi yozish va oʻqish shinasiga ulanadi. Oʻqish jarayonida bit shinasini soʻz shinasidan oʻqiydi va oʻqilgan maʼlumotni chiqish shinasiga uzatadi. Yozish jarayonida esa bit shinasini kiruvchi axborotni qabul qilib, soʻz shinasini orqali yacheykalarga yozadi.

8.3-rasmda 8 bitli 16 ta soʻzdan tashkil topgan kichik xotira mikro sxemasi keltilgan. Sxemalarda yozishda 16x8 deb yoziladi. Har bir sxemaning kirish va chiqishi bitta ikki yoʻnalishli axborot shinasiga ulanadi, u esa kompyuterning shinasiga ulanadi. Adres va axborot shinasidan tashqari ikkita, RW (Read/Write) va CS (Chip Select – xotira elementini tanlash) deb nomlangan boshqarish shinasini mavjud. RW ning

kirish qiymati bajariladigan amalni, CS ning kirish qiymati xotirani tashkil etuvchi mikro sxemalardan birini koʻrsatadi.

8.3-rasmdagi xotira sxemasi 128 bit axborot saqlaydi va adres, axborot, boshqaruv uchun 14 ta shina talab qiladi. Yana 2 ta – elektr toki va “заземление” shinasini ham kerak.



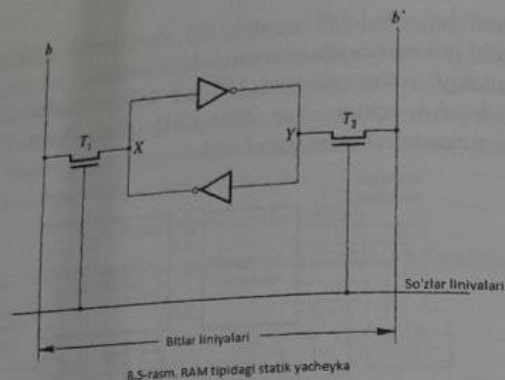
8.4-rasm. 1024 x 1 xotira mikro sxemasi

8.4-rasmda 1K (1024) yacheykali xotira mikro sxemasi keltirilgan. Bu 128x8 massivida, 19 ta tashqi chiqish bilan tashkil qilinadi. Shu sonli yacheykalarni 1Kx1 massivi bilan tashkil qilinsa, 10 razryadli adres, bitta axborot shinasini, 15 ta chiqish kerak boʻladi. Bunda 10 razryadli adres satrlari va ustunlar adresini aniqlovchi 2 ta 5 razryadli boʻlinadi. Adres satri signali 32 ta yacheykali satrni adreslaydi. Ustun adresi xususiyatiga koʻra bu yacheykalarining faqat bittasi chiquchi multipleksor yoki kiruvchi demultipleksor orqali tashqi axborot shinasiga bogʻlanadi.

Zamonaviy xotira mikro sxemalari rasmdagiga qaraganda juda koʻp yacheykalardan tashkil topadi. Rasmdagi sxemalar xotira tuzilishini tushintirish uchundir. Katta mikro sxemalar ham xuddi shunday tuziladi, faqat katta razmerli massiv va koʻp sonli tashqi ulanishlarga ega. Masalan, 4 megabaytli mikro sxema 512Kx8 strukturali boʻlishi mumkin va 19 ta adres, 8 ta axborot almashinish shinalariga ega.

### 8.4. Statik xotira

Elektr toki mavjud boʻlganda ishlaydigan xotira mikro sxemasi statik xotira deyiladi (8.5-rasm).



Ikki invertorni X-simon joylashtirib ajratkich hosil qilinadi. Bu ajratkich T1 va T2 tranzistorlari orqali bit shinasiga bog'lanadi. So'z shinasining boshqaruvi ostida tranzistorlar ajratkich vazifasini ham bajaradi. So'z shinasida tok mavjud bo'lmasa tranzistorlar o'chirilgan va ajratkichining holati o'zgarmaydi.

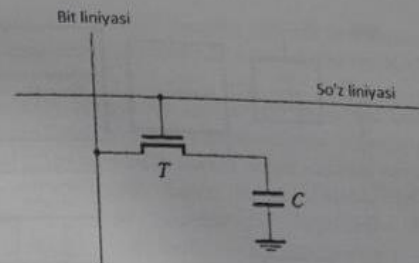
**O'qish.** Xotiradan o'qishda xotirani boshqarish sxemasi so'z shinasini faollashtiradi, bunda T1 va T2 yopiladi. Agar yacheykaning qiymati 1 ga teng bo'lsa,  $b$  da yuqori,  $b'$  da past kuchlanish bo'ladi. Agar yacheyka 0 ga teng bo'lsa aksincha. Bit shinasini oxiridagi sxemalar  $b$  va  $b'$  xolatini tekshiradi va mos chiqish signaligini o'rnatadi (hosil qiladi, uzatadi).

**Yozish.** Yacheyka xolatini aniqlash uchun tegishli qiymat  $b$  ga, qo'shimchasi esa  $b'$  ga joylanadi va so'zshinasi faollashtiriladi. Kerakli signallar bit shinasida hosil bo'ladi.

### 8.5. Asinxron xotira

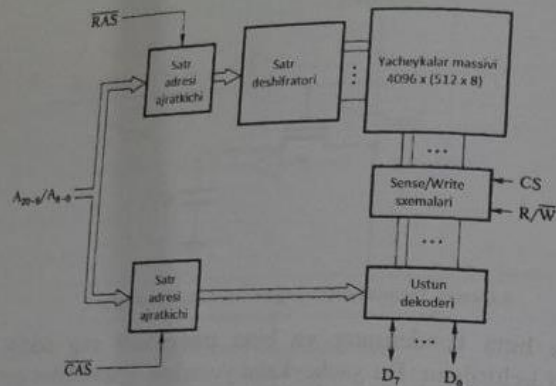
Statik RAM (xotira) tez ishlaydi, lekin har bir yacheykasi bir nechta tranzistorli bo'lgani sababli qimmat baholanadi. Shu sababli nisbatan oddiyroq tuzilmali, nisbatan arzonroq bo'lgan xotira qurilmalari ishlab chiqariladi. Bunday yacheykalar o'z xolatini uzoq muddat saqlab tura olmaganligi uchun *dinamik xotira (RAM)* deyiladi.

Dinamik xotira yacheykasida axborot kondensatordagi zaryad shaklida saqlanadi, va bu zaryad bir necha millisekund turishi mumkin. Yacheykadagi axborotni ko'proq muddat saqlab turish zaruriyati kondensatordagi zaryadni to'ldirish bilan ta'minlanadi (8.6-rasm).



8.6-rasm. Dinamik xotiraning bir tranzistorli yacheykasi

Bunda bitta kondensator va bitta tranzistorli eng oddiy xotira yacheykasi keltirilgan. Bu yacheykaga yozishda T tranzistor qo'shiladi va bit shinasiga kuchlanish uzatiladi. Natijada kondensatorda zaryad hosil bo'ladi. Tranzistor o'chirilganda kondensator razryadlanib boshlaydi. Kondensator zaryadi chegara qiymatdan pasaymasdan oldin olingan axborot to'g'ri bo'ladi. O'qish jarayoni yacheyka tranzistori qo'shilgan holatda sodir bo'ladi. Bit shinasiga ulangan kuchaytirgich kondensator zaryadi belgilangan chegara qiymatdan yuqori yoki pastligini o'lchaydi. Yuqori bo'lsa, bit shinasiga yuqori kuchlanish (1 ni) uzatadi va kondensator qiymatni saqlab turish uchun qayta zaryadlanadi. Agar kuchlanish chegara qiymatdan past bo'lsa kuchaytirgich bit shinasiga past kuchlanish (0 ni) uzatadi va kondensator kuchlanishi pasayishda davom etaveradi. Shunday qilib, o'qish jarayonida yacheyka tarkibi avtomatik yangilanadi. Tanlangan satrning hamma yacheykalari bir vaqtda o'qiladi, natijada butun satr qiymati yangilanadi.



8.7-rasm. 2M x 8 hajmi dinamik xotira mikrosxemasining ichki tashkil qilinishi

8.7-rasmda 2Mx8 konfiguratsiyali 16 megabitli DRAM mikrosxemasi keltirilgan. Unda har bir satrning 4096 ta yacheykalari 8 ta yacheykadan iborat, har biri 512 bayt axborot hajmi 8 yacheykali 512 guruhga bo'lingan va 4Kx4K massiviga birlashgan. O'z-o'zidan, satrni tanlash uchun 12 adres razryadi kerak bo'ladi. Yana 9 razryad satrdagi 8 bitli guruhni tanlash uchun zarur. Demak, bunday mikrosxemada 1 baytni o'qish/yozish uchun 21 razryadli adres kerak bo'ladi. Yuqori 12 va quyi 9 razryadlar satr va ustun bayt adreslarini tashkil etadi. Mikrosxemaning chiqishlari sonini ixchamlashtirish maqsadida satr va ustun adreslari 12 chiqishga multiplekslanadi. O'qish yoki yozish jarayonida mikrosxemaning chiqishlariga dastlab satr adreslari uzatiladi. Kiruvchi RAS (Row Address Strobe – satr adresi strobi) signaliga javob qilib uzatilgan adreslar satrlarga yoziladi. 334

#### 8.6. Operativ va doimiy saqlash qurilmalari. Stekli va assotsiativ xotira

**Asinxron xotira.** Ma'lumotlar satrlarga yuklangandan keyin adres chiqishlariga ustun adreslari yuboriladi. CAS (Column Address Strobe) signali hosil qilinadi. Bunda axborot dekodlanadi va SWdan tegishli 8 talik guruh ajratiladi. Agar boshqaruvchi RW signali o'qishni ko'rsatsa, kiruvchi qiymatlar axborot shinasiga ( $D_{7-0}$  ga) yuboriladi. Yozish amali uchun  $D_{7-0}$ dan axborot sxemaga uzatiladi. Keyin u mos ravishdagi 8 ta ustundagi ko'rsatilgan yacheykalardagi ma'lumotlarni yangi; ash uchun ishlatiladi. Ko'p mikrosxemalarda RAS va CAS signallarini faollashtirish

past kuchlanish bilan ko'rsatiladi, bundan adresni stroblash tok kuchi kamayayotganda bajarilishi tushuniladi. sxemalarda bu RAS va CAS deb belgilanadi.

O'qish yoki yozish jarayonida satr adresi uzatilishi shu satrdagi barcha yacheykalarni o'qilishi va yangilanishiga (обновление) olib keladi. DRAMning holatini saqlab turish uchun doimiy ravishda uning har bir satriga murojaat qilib turish kerak. Odatda bu avtomatik regeneratsiya sxemasi deyiladi. Regeneratsiya sxemalarini foydalanuvchiga ortiqcha murakkablik tug'dirmaslik uchun odatda dinamik xotiraga integratsiya qilinadi.

Bu turdagi dinamik xotiralar *asinxron rejimda* boshqariladi. U esa xotirani boshqaruvchi maxsus sxemaning generatsiyasi orqali, RAS va CAS signallari bilan amalga oshiriladi. Bunda markaziy protsessor xotiradan javob kelishini kutishga moslashishi kerak. Bunday xotira *asinxron DRAM* deyiladi. O'zining xotira hajmi kattaligi va mikrosxemaning arzonligi sababli DRAMlar kompyuterlarda keng ishlatiladi. Mashinalarda mikrosxemalarning sonini kamaytirishga bo'lgan talab natijasida DRAMlar shunday tashkil qilinadiki, o'qish/yozish jarayonlari parallel bajariladi. Mikrosxemalar turli tashkil qilinishi sababli turli tipagi xotirani komponovka qilish mumkin. Masalan, 64 mb xotirani 16M x 4, 8M x 8, 4M x 16 shaklida joylashtirish mumkin.

**Tezkor sahifali rejim.** 5.7-rasmda keltirilgan DRAMga murojaat qilish vaqtida barcha 4096 yacheykalar o'qiladi, lekin axborot qatori  $D_{7-0}$  da faqat 8 bit joylashadi. Bu bayt ustun adresi  $A_{7-0}$  da tanlanadi. Bu sxemani modifikatsiya qilish mumkin, shunda u satrning baytlariga qayta o'qimasdan, balki qayta murojaat qila oladi. Buning uchun har bir satrning chiqish kuchaytirgichiga ajratkich ulanadi. Yangi adres satrni tanlanganda uning barcha bitlari uchun ajratkichlar o'rnatiladi. Bunda kerakli baytni chiqish liniyasiga joylash (uzatish) uchun kerakli ustun adresini ko'rsatish kifoya qiladi.

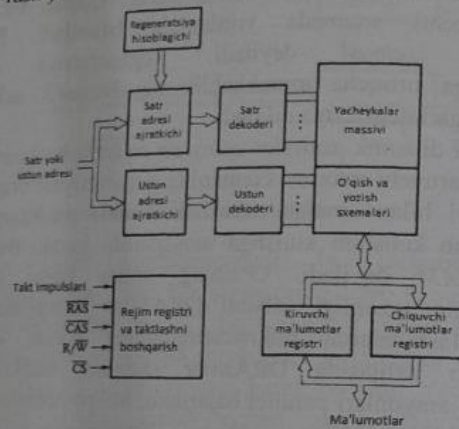
Odatda baytlar ketma-ket, adreslari o'sib borish tartibida uzatiladi. Uzatishni tezroq amalga oshirish maqsadida CAS boshqaruvchi signali yordamida ustunlarning ketma-ketligini generatsiya qilish lozim. Bunday blokli uzatish rejimi **Tezkor sahifali rejim (FPM – Fast Page Mode)** deyiladi. Baytlarning kichik bo'laklari **bloklar**, katta bo'laklari **sahifalar** deyiladi.



Axborotlarni tezkor blokli uzatish grafik terminallarda keng qo'llaniladi. Umumfoydalanishdagi kompyuterlarda esa asosiy xotira va kesh xotira orasida axborot almshishda ishlatiladi.

### 8.7. Sinxron xotira

Xotiraning takt signali bilan teng ishlashi jarayoni *sinxron DRAM* deyiladi (8.8-rasm).



8.8-rasm. Sinxron DRAM

Ko'rinib turganidek, yacheykalar massivi asinxron DRAMniki bilan bir xil. Axborot va adres shinalari registrlar bilan buferlanadi. Har bir o'qish kuchaytirgichining chiqishi ajratkich bilan bog'langan. O'qish jarayonida bu ajratkichlarga tanlangan satrning barcha yacheykalari yuklanadi. Agar satrga murojaat o'qish uchun emas, balki faqat axborotlarni regeneratsiya qilish uchun bo'lsa, ajratkichning holati o'zgarmaydi. Tanlangan ustunlarga mos bo'lgan ajratkichlardagi axborotlar chiquvchi registrlarga yuboriladi.

SDRAM xotira (sinxron xotira) bir necha rejimlarda ishlashi mumkin. Bu rejimlar boshqaruvchi axborot bilan tanlanadi. SDRAMda ustunlarni ketma-ket tanlashda CAS liniyasidan keluvchi signallarni ishlatish shart emas. Boshqarish signallarini bevosita sxemaning ichida hosil qilish mumkin. Bunda takt signali va ustunlar hisoblagichi ko'rsatkichi ishatiladi. Buu holatda yangi axborotlar axborot shinasiga har bir takttda joylanib boriladi.

SDRAMda ichki generatsiya sxemasi mavjud. Uning tarkibidagi adres satri hisoblagichi generatsiyani bajaradi. odatda SDRAMda axborotlarhar 64 ms (millisekund) da generatsiyalanadi.

Zamonaviy SDRAMlar bir necha GHz (gigagers) chastotada (takt hastotasida) ishlaydi.

**Kutish vaqti.** Xotira va markaziy protsessor orasida ma'lumotlar so'zlar yoki so'zlar bloki shaklida uzatiladi. Sahifali shakldagi axborotlar esa xotira va disklar orasida uzatiladi. Bunday amallarning tezligi va natijadorligi (эффективность) mashinaning umumiy unumdorligiga bevosita ta'sir qiladi. Unumdorlik ikkita ko'rsatkish: kutish vaqti va o'tkazuvchanlik qobiliyati bilan xarakterlanadi.

Kutish vaqti termini (yoki *latentlik*) xotiradan/ga bir so'zni uzatishga sarflangan vaqtdir. Agar axborotlar so'zma-so'z uzatilganda edi, kutish vaqti (latentlik) va xotira unumdorligi bir tushuncha bo'lib qolar edi. Lekin axborotlarni blokli yoki sahifali uzatishda uzatishga ketgan vaqt alohida so'zlarni uzatish tezligi va axborot bloki hajmiga teng bo'ladi. Shu sababli blokli uzatishda kutish vaqti deganda birinchi so'zni uzatish vaqti tushuniladi. Odatda birinchi so'zni uzatish boshqalariga qaraganda ko'proq vaqt talab qiladi.

Bunda xotiraga murojaat sikli RAS signalini faollashtirishdan boshlanadi. 5 takt o'tgandan keyingina birinchi so'z uzatiladi. Bunda kutish vaqti 5 takt ga teng deyiladi. 100 MHH chastotada bu 50 ns (nanosek) ga teng. Qolgan 3 ta so'z ketma-ket taktlarda uzatiladi.

Bloklar turli uzunliklarga ega ekanligidan unumdorlikni bir sekundda uzatiladigan bit/bayt soni bilan o'lchash qabul qilingan. Bu ko'rsatkich *o'tkazuvchanlik qobiliyati* deyiladi. Bir yoki bir nechta mikrosxali xotira blokining o'tkazuvchanlik qobiliyati xotirada saqlanayotgan axborotlarga murojaat tezligi va parallel chaqiriladigan bitlar sonidan kelib chiqadi. Lekin, mashinaning real unumdorligi faqatgina xotira unumdorligiga bog'liq emasligi ayon. U asosan xotira va markaziy protsessor orasidagi shinaning o'tkazuvchanlik qobiliyatiga bog'liq. Odatda xotira mikrosxemalari shinaning o'tkazuvchaligiga qarab ishlab chiqiladi. O'tkazuvchanlik qobiliyati ma'lumotlari uzatish tezligi (ularga murojaat tezligi) va axborot shinasiga kengligiga bog'liq bo'ladi.

### 8.8. DDR SDRAM

Texnologiyalar rivojlanishi kompyuter tizimini yanada tezroq ishlashini talab qiladi. Bunda xotira mikrosxemalarini ham ish tezigini

o'shish zarurati paydo bo'lmoqda. Standart SDRAMlar takt signali chastotasi bilan ishlaydi. Undan keyingi xotira qurilmalarida yacheykaga murojaat shakli saqlanib qolib, axborot uzatish ikki yoqlama qilingaligi bilan farqlanadi. Odatiy holatda kutish vaqti bu xotirada odatiy xotiradan farq qilmaydi, lekin katta paketli amallarda o'tkazuvchanligi deyarli 2 barobar yuqori bo'ladi. Bunday xotira qurilmasi *ikkilangan o'tkazuvchanlikli SDRAM (Double Data Rate SDRAM - DDR SDRAM)* deyiladi.

Bunda axborotlarga murojaatni tezlatish uchun yacheykalar massivi 2 ta o'zaro mustaqil massivga bo'lingan. Ma'lumotlar blokining ketma-ket so'zlari turli massivlarda saqlanadi. Bunday uslub bir vaqtda xotiradan ikkita so'zni o'qish imkonini beradi. Ularning birinchisi takt signali boshida, ikkinchisi oxirida uzatiladi.

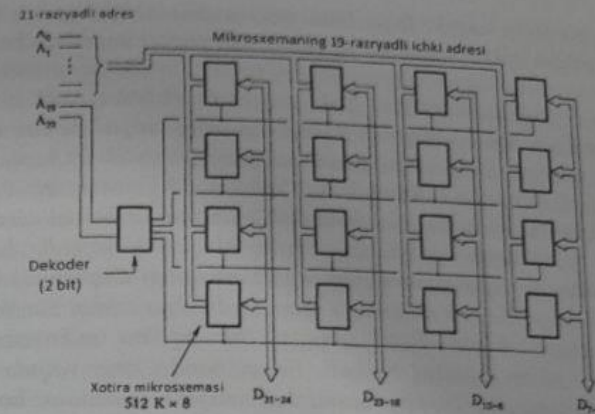
DDR SDRAM va standart SDRAM asosan axborotlar blokli uzatiladigan mashinalarda yaxshi natija beradi. Bunga misol qilib umummo'ljallangan kompyuterlar misol bo'la oladi. Shuningdek, blokli uzatish yuqori sifatli videodispleylarda ishlatiladi.

### 8.9. Katta hajmli xotira strukturasi

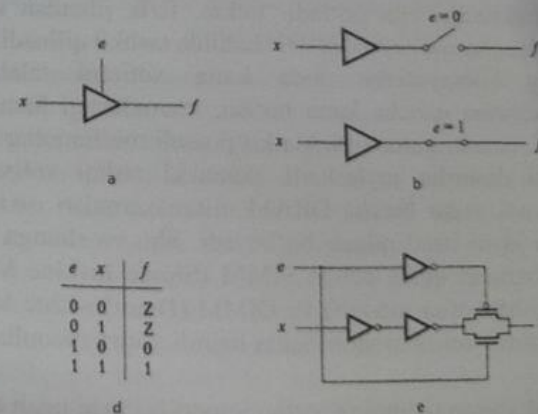
Yuqorida ko'rilgan xotira turlari primitiv, ularni bitta mikrosxemada tashkil qilish mumkin. Katta hajmli xotiralarni tashkil qilish usullari quyida ko'riladi.

**Statik xotira tizimlari.** Har biri 32 bitli bo'lgan 2 M dona ( $2097152 \text{ ta} = 2^{21}$ ) so'zli xotirani qaraymiz (8.9-rasm).

Bunda (8.9-rasm)da) 512K x 8 statik mikrosxema asosida tashkil qilinish uslubi ko'rsatilgan. Har bir ustun so'zning 8 bitini saqlovchi 4 ta mikrosxemadan tashkil topgan. 4 ta shunday to'plam 2Mx 32 xotirani beradi. Har bir mikrosxemaning boshqaruvchi elementi mavjud, u CS (Chip Select - mikrosxemani tanlash) deyiladi. Bu elementga 1 uzatilganda (yuqori kuchlanish berilsa) mikrosxema ma'lumotlarni qabul qiladi yoki ularni o'z shinasiga joylaydi. Mikrosxema chiqishi 3 xolatda bo'lishi mumkin (8.10-rasm):



8.9-rasm. 512K x 8 statik xotira asosida 2M x 32 modulli xotirani tashkil qilish



8.10-rasm. Uch xil holatli buferlar. (a)-ramziy tasvirlanishi; (b)-ekvivalent sxemasi (d)-rostlik jadvali; (e)-amalga oshirilishi

Har bir joriy vaqtda faqat bitta mikrosxema chiqishga ma'lumot uzatadi, qolgan mikrosxemalarning chiqishlari yuqori impedans holatida bo'ladi.

(Yuqori impedans holati – yoki yuqori Omlil holat, yoki Z-holat, yoki o'chirilgan holat – raqamli mikrosxemaning chiqish holati, ushbu chiqishga ulangan ichki va tashqi sxemalarning qarshiliklari keskin farq qilishi holati. Mantiqiy elektronika "mantiqiy 1", "mantiqiy 0" dan tashqari uchinchi holat. Bunda shina tipli ulanishni tashkil qilish uchun chiqishda o'rnatilgan qo'shimcha kalit chiqishni yopadi, boshqa sxemalar bilan yopiq tranzistorning yuqori qarshiligi (impedans) orqali ulanadi. Bunday ulanish sistemaga ulangan boshqa tashqi zanjirlarga ta'sir qilmaydi, demak bitta zanjirga bir nechta mikrosxemalar chiqishlarini ulash mumkin bo'ladi. Faqat bunda joriy vaqtda faqat bitta mikrosxema fol bo'lib, boshqalari yuqori impedans holatida bo'lishini nazorat qilib turish lozim).

Bunday xotiradan 32 razryadli so'zni tanlab olish uchun 21 adresli razryad zarur bo'ladi. Yuqori 2 razryad CS ning qaysi 4 ta boshqaruvchi signalini faollashtirish kerakligini aniqlaydi, qolgan 19 ta razryad belgilagan satrning aniq baytiga murojaat qilishda ishlatiladi. Barcha mikrosxemalarning RW chiqishlari birlashtiriladi, natijada yagona umumiy chiqish hosil bo'ladi.

### 8.10. Dinamik xotiralar tizimi

Katta dinamik xotiralar tizimi ham yuqoridagi (5.10) slyadda ko'rsatilgan strukturaga ega bo'ladi. Lekin, fizik jihatdan ko'pincha qulayroq hisoblangan xotira modullari shaklida tashkil qilinadi.

Zamonaviy kompyuterlar juda katta xotirani talab qiladi. Mashinaning xotirasi qancha katta bo'lsa, unumdorligi ham shuncha yuqori deyiladi, chunki katta xotirada ko'p sonli ma'lumotlar va ularni qayta ishlovchi dasturlar joylashadi, demakki tashqi xotiraga talab shuncha kamayadi. Agar barcha DRAM mikrosxemalari *ona plata* da joylashsa, ko'p joyini band qilgan bo'lar edi. Shu va shunga o'xshash muammolarni bartaraf qilish uchun SIMM (Single In-Line Memory – bir qatorli chiqishli xotira moduli) va DIMM (Dual In-Line Memory – ikki qatorli chiqishli xotira moduli) katta hajmli xotira modullari tashkil qilindi.

Statik RAM asosan tizimning tezligi yuqori bo'lishi talab etiladigan tizimlarda qo'llaniladi. Uning bazaviy yacheykalarini tashkil qilish sxemasi murakkabligi uchun narxi va hajmi katta bo'ladi. Asosann kesh

xotirani tashkil qilishda ishlatiladi. Dinamik RAM ko'p kompyuterlarda ishlatiladi.

**Xotira kontrolleri.** Tashqi aloqalar sonini kamaytirish maqsadida mikrosxemaning adres chiqishlari multipleksalanadi (*multiplekslash – ing multiplexing, muxing kanalni siqish, bir necha potok ma'lumotni bir kanaldan uzatish.* **Telekommunikatsiyada** bir necha mantiqiy kanallardagi ma'lumotlarni bitta fizik kanalda uzatish, bu optik tolali yoki misli kabel va h.k. **informatsion texnologiyalarda** bir necha ma'lumotlar potokini (virtual kanallarni) birlashtirish. Misol: videofayl uzatilishida bir vaqtda bir necha audio ham uzatiladi). Adres 2 qismga bo'linadi. Dastlab yuqori adres bitlari aniqlanadi, ular yacheykalar massividan satrni tanlaydi. RAS signali bilan bu bitlar mikrosxemadagi ajratkichlarda saqlanadi, keyin shu adres kirishlariga kichik adresli bitlar uzatiladi, ular esa CAS signali bilan ajratkichlarga joylanadi.

Odatda markaziy protsessor adresni butunicha uzatadi, shinaga hamma razryadlarini joylaydi. Uni multiplekslashni *xotira kontrolleri* bajaradi. odatda kontroller markaziy protsessor va dinamik xotira orasida joylashadi.

Xotiraga murojaat qilish signalini olgan kontroller markaziy protsessor dan to'liq adresni RW signalini so'rab oladi. Keyin naybat bilan xotiraga satr va ustun adreslarini yuboradi va RAS va CAS signalini shakllantiradi. RW va CS signallarini ham yuboradi. CS ning faol holati past kuchlanish deb olinadi.

### 8.11. Operativ va doimiy saqlash qurilmalari, stekli va assotsiativ xotira

**Rambus xotira.** Dinamik xotiraning unumdorligi kutish vaqti va o'tkazuvchanlik qobiliyati bilan o'lanishini bilamiz. Dinamik xotiralarning barchasida yacheykalar massivining qurilishi bir xil bo'lar ekan, deyarli barcha mikrosxemalar (xotira mikrosxemalari) bir xil kutish vaqtiga ega bo'ladi. O'tkazuvchanlik qobiliyati esa nafaqat mikrosxema strukturasi, balki mikrosxema va markaziy protsessor ning aloqasini tashkil qilinish uslubiga ham bog'liq. DDR SDRAM va SDRAM markaziy protsessor shinasiga bevosita ulanadi. Shu sababli ma'lumot almashish mikrosxemaning tezkorligidan tashqari shinning o'tkazuvchanligiga ham bog'liq bo'ladi. Masalan 133 MHz takt chastotali shina har 7.5 ns (nanosekund) dabitta (yoki ikkita) ma'lumot almashishni bajaradi (agar takt signalining ikki fronti ham ishlatilsa

ikkita). Shinadan yanada ko'proq ma'lumotni uzatishning yagona usuli uni kengaytirishdir.

Lekin shinaning juda kengayib ketishi uning narxi oshishiga va ona platani ko'p joyini band qilishga olib keladi. Buning o'rniga tor lekin tezkorligi yuqori shinani ishlatish mumkin. Bu texnologiya Rambus shinalarida qo'llanilgan (**Rambus firmasi**). Rambus texnologiyasining alohida xususiyati deb mikrosxemalar orasidagi axborot almashishda tezkorsignalizatsiya usulini aytish mumkin. Odatda mikrosxemalarda 0 va 1 ni hosil qiluvchi minimal va maksimal qiymatlar ishlatiladi. Rambus texnologiyasida esa bazaviy kuchlanishdan kichik qiymatga farq qiladigan kuchlanishlar ishlatiladi. Bazaviy kuchlanish 2 V ga teng, 0 esa  $2-0,3=1,7$  V, 1 esa  $2+0,3=2,3$  V ga teng. Bunday signallar tizimi *differensial* deyiladi. Kichik kuchlanishlar farqi tranzaksiya vaqtini qisqartiradi va ma'lumotlar uzatilishini tezlashtiradi. (*Tranzaksiya* – ing transaction ma'lumotlar bazasi bilan bajariladigan ketma-ket amallar guruhi. Tranzaksiya butun holicha va muvaffaqiyatli bajarilishi mumkin yoki umuman bajarilmasligi mumkin, bunda hech qanday foyda/zarar bo'lmaydi. **Tranzaksiyaga misol.** 5 raqamli hisob egasidan 7 raqamli hisob egasiga 10 birlik summani o'tkazish talabi qo'yilgan. Amallar bajarish ketma-ketligi quyidagicha:

1. 5 ning hisobini tekshirish
2. 5 ning hisobini 10 birlikka kamaytirish
3. Yangi balansni 5 ning hisobiga yozib qo'yish va 5 ni hisobini yopish
4. 7 ning hisobini tekshirish
5. 7 ning hisobini 10 birlikka ko'paytirish
6. Yangi balansni 7 ning hisobiga yozib qo'yish va 7 ni hisobini yopish)

Bu yerdagi barcha harakatlar "hisob raqamlar orasida mablag' ko'chirish" deb nomlanuvchi bitta mantiqiy butunlikni tashkil etadi va tranzaksiya hisoblanadi. Agar bu tranzaksiyani o'rtasida uzadigan bo'lsak va bajarilgan amallarni bekor qilmasak, 5 ning hisobi 10 birlikka kamayadi, 7 esa hech narsa olmaydi.

Differensial signal tizimini ishlatishda kommunikatsiya tarmoqlarini qurishning alohida texnologiyasi zarur. Bu esa keng shinalarni hosil qilishni murakkablashtiradi. Bundan tashqari differensial signallar bilan ishlovchi alohida interfeys sxemalar kerak bo'ladi. Rambus bunday aloqa liniyalarini qurishning to'liq komplektini

taqdim etadi. Ular Rambus kanalai deyiladi. Rambus shinalari 400 MHz chastotada ikki tomonlama ishlaydi, umumiy unumdorlik 800 MHz.

Rambus xotira mikrosxemalari DRAMning standart texnologiyalari asosida qurilsa ham, alohida tuzilish xususiyatlariga ega bo'ladi. Ular bir nechta so'zlarga parallel kira oluvchi bir necha xotira banklariga birlashadi. Mikrosxema tarkibiga Rambus shinasini uchun interfeys sxemasi ham kiradi. Bunday sxemalar Rambus DRAM (RDRAM) deyiladi.

RDRAM va markaziy protsessor, RDRAM va boshqa qurilmalar o'rtasidagi aloqa aloqa liniyalari orqali uzatiluvchi paketlar vositasida bajariladi. Paketlarning 3 xil turi mavjud: so'rov paketlari, tasdiq paketlari, ma'lumot paketlari. So'rov paketida bajariladigan amalning tipi ko'rsatiladi. Bu paket xotira adresi va uzatiladigan baytlar sonini aniqlovchi 8 razryadli hisoblagichga ega. Amal tipi xotiraga o'qish/yozishni va RDRAMning turli boshqaruvchi registrlaridan o'qish/yozishni bajaradi. Shinadan so'rov kelishi bilan so'rovni qayta ishlovchi qurilma 2 xil javob varianti beradi: 1) ijobiy javob – agar so'ralgan ma'lumotni shu zahoti uzatish mumkin bo'lsa; 2) salbiy javob – ma'lumot tayyor emas, qayta urinish lozim bo'lsa.

So'rovdagi bitlar soni ma'lumot liniyasidan ko'p bo'ladi, shu sababli so'rov bir necha taktida bajariladi. Buni kompensatsiyasi ma'lumotlarni uzatishning yuqori tezligi bilan qoplanadi. RDRAM mikrosxemalarini SIMM va DIMM kabi yirik modularga birlashtirish mumkin (RIMM). Bitta shunday modulda 16 tagacha RDRAM mikrosxemasi bolishi mumkin.

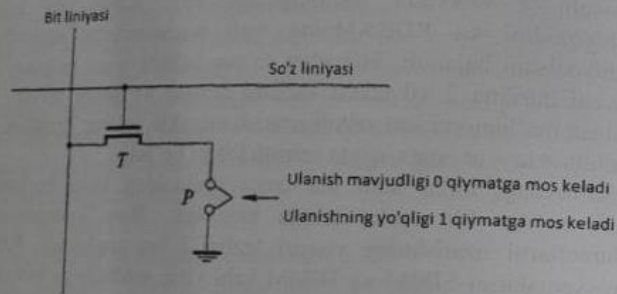
Rambus texnologiyasi DDR SDRAM bilan raqobatda bo'la oladi. Har birining o'ziga xos yutuqlari va kamchiliklari mavjud.

## 8.12. Faqat o'qish uchun mo'ljallangan xotira.

SRAM va DRAM mikrosxemalari elektrga bog'liq bo'lib, elektr uzilishi bilan barcha ma'lumotlar o'chib ketadi. Lekin ko'plab qurilmalar va komponentlar (tashkil etuvchilar) borki, ularga axborot har doim zarur. Masalan qattiq disk (vinchester). Katta hajmli ma'lumotlarni, operatsion tizimni saqlashi zarur. Kompyuter yoqilganda dasturiy ta'minot (DT) qattiq diskdan asosiy xotiraga yuklanadi. Buni maxsus yuklovchi dastur bajaradi. Yuklovchi dasturning kodi ham ancha katta bo'lganligi sababli, uning bir qismi ham qattiq diskda saqlanadi. markaziy protsessor DT ni asosiy xotiraga yuklashda maxsus

komandalarni bajaradi. Agar barcha xotiralar energobog'liq bo'lganda markaziy protsessor yukalshni bajara olmas edi. Shu sababli kompyuterda kichik hajmli energomustaqil xotira mavjud. Unda kompyuter ishga tushirilganda birinchi bajarilishi kerak bo'lgan komandalar va DT ni asosiy xotiraga ko'chiruvchi komandalar joylashadi.

Energomustaqil xotiraning turlari ko'p. Ulardan ma'lumot o'qish oddiy tartibda, xuddi SRAM yoki DRAMdan o'qish kabi bajariladi. Ularga ma'lumot yozish uchun esa alohida protsedura zarur. Ishchi rejimda bu xotiradan ma'lumotlar faqat o'qiladi, shu sababli bunday xotira faqat o'qish uchun mo'ljallangan xotira – ROM (Read Only Memory) deyiladi.



8.11-rasm. PROM yacheykasi

**ROM.** 8.11-rasmda ROMni tashkil qilishning uslubi ko'rsatilgan. Agar P dagi tranzistor "yerga ulangan" bo'lsa, bu yacheykada mantiqiy 0 hosil bo'ladi, aks xolda mantiqiy 1. Bit liniyasi rezistor orqali tok manbaiga ulangan. Yacheyka xolatini o'qish uchun so'z liniyasini faollashtirish kerak. Bunda tranzistor kaliti yopiladi va agar tranzistor va "yerga ulanish" aloqasi mavjud bo'lsa, bit liniyasidagi kuchlanish 0 gacha pasayadi. (mantiqiy 0 hosil bo'ladi) Agar "yer" bilan ulanish bo'lmasa bit liniyasida yuqori kuchlanish mavjud bo'ladi (mantiqiy 1 hosil bo'ladi). Bit liniyasi oxiridagi sxema to'g'ri chiqishli bo'lib, to'g'ri qiymat uzatadi (invers emas).

**PROM.** Ba'zi ROM mikrosxemalarining ma'lumotlarini foydalanuvchi dasturlash yo'li bilan o'zgartira olishi mumkin. Bunday mikrosxemalar dasturlanuvchi ROM (*programmable ROM*) deyiladi (PROM).

Bu mikrosxemani dasturlashda (unga ma'lumot yozishda) suzuvchan ulanish deb ataladigan texnologiya ishlatiladi (плавающее соединение). Bu sxemada P bilan belgilangan. Dasturlanishgacha barcha yacheykalarda 0 lar joylashadi. 1 ni joylashtirish uchun suvuvchan ulanishni yuqori kuchlanish bilan kuydiriladi. O'z-o'zidan ayonki, bu jarayon faqat 1 marta ishlatiladi, qayta dasturlash mumkin emas.

PROM xotira ROMga nisbatan qulayroq va egiluvchanroq. Uning qulayligidan ko'p sonli bir xil mikrosxemalarni ishlab chiqarishda o'zgarmaydigann va doimiy saqlanadigan ma'lumotlarni saqlashda foydalaniladi.

**EPROM.** ROM mikrosxemalarining yana bir tipi ma'lumotlarni qayta-qayta yozish imkonini beruvchi EPROM (*Erasable Programmable ROM*) deb nomlanuvchi mikrosxemadir. EPROM dagi ma'lumotlar uzoq vaqt saqlanishi e'tiboridan uni ROMning o'rniga vaqti-vaqti bilan yangilanib turadigan dasturlarni saqlashda ishlatiladi.

EPROM strukturasi ROM strukturasi bilan bir xil (yuqoridagi rasm). Faqat farqli jihati unda yerga ulanishning o'rniga tranzistor qo'yiladi. Bu tranzistor ba'zon ochiq, ba'zan esa yopiq holatda bo'ladi. EPROMning asosiy xususiyati ham shu tranzistordir. Tranzistorga yuqori kuchlanish berib, uni ochiq kalit sifatida ishlatish mumkin. Shunday qilib, EPROM va ROMning strukturasi birxilligidan ulardan bir xil strukturali xotira qurilmalarini qurish mumkinligini bilib olamiz.

EPROMning eng asosiy xususiyati undagi ma'lumotlarni o'chirib tashlab, qayta dasturlash mumkin. O'chirish uchun ultrabinafsharang yorug'lik bilan tranzistorlardagi barcha zaryadlar yo'qotiladi, buning uchun esa ranzistorlar shaffof korpuslarda bo'lishi talab etiladi.

**EEPROM.** EPROMning kamchiligi – undagi ma'lumotlarni yangilash uchun xotira chipini platadan olish kerak, qolaversa ultrsbinafsharang yorug'lik mikrosxemalagi hamma ma'lumotlarni o'chirib yuboradi. ROMning dasturlanuvchi boshqa shakli ham mavjud bo'lib, beunda qayta dasturlash elektr toki bilan bajariladi. Bunday mikrosxemalar *elektron dasturlanuvchi doimiy xotira (Electrically Erasable Programmable ROM, EEPROM)* deyiladi. Bunday mikrosxemani dasturlashda uni pladan olish shart emas, yana ularni tanlov asosida, ba'zii qismlarinigina ham dasturlash mumkin. EEPROMning kamchiligi sifatida ma'lumotlarni o'chirishda, o'qishda va yozishda turli kattalikdagi kuchlanishlar kerak bo'lishini ta'kidlash mumkin.

**Flesh xotira.** Bunday xotira mikrosxemasining ishlashi xuddi EEPROM kabi bo'lib, asosiy farqlanishi shundaki, EEPROMda o'qish va yozish bitta yacheykadan bajariladi, flesh xotira esa o'qishda bitta yacheykadan o'qiydi, yozishda esa bloklab yozadi. Yozish oldidan yozilayotgan blokda barcha ma'lumotlar o'chiriladi. Flesh xotira yuqoridagi darajadagi siqilganlikka ega, shu sababli EEPROM ga nisbatan kattaroq hajmli (axborot jihatdan) bo'ladi. Shuningdek, bir xil kuchlanishli tok bilan yozish/o'chirish/o'qishni bajara oladi.

**Flesh kartalar.** Oddiy flesh xotirada kam sonli ma'lumotlar saqlanishi e'tiboridan ko'p sonli ma'lumotlarni saqlash uchun flesh xotira mikrosxemalari bitta kartaga montaj qilinadi. Bunday kartalar standart interfeysga ega ekanligidan, ularni turli qurilmalarda ishlatish mumkin. Zamonaviy flesh kartalar 1 TB gacha ma'lumot saqlaydi.

**Flesh disklar.** Kompyuterning qattiq diskini almashtira oladigan darajadagi katta hajmli flesh xotiramodullari ham mavjud. Bunday har bir modul qattiq diskni o'rniga sistemaga o'rnatilishi mumkin. Flesh diskning asosiy xususiyati shundaki, uni qo'shimcha ko'chma elementlarsiz olib yurish mumkin. Unga murojaat qilish vaqti qattiq disknikidan kam, javob vaqti ham. Yana qattiq diskka qaraganda kam energiya talab qiladi, bu esa batareyada ishlovchi (smartfon, noutbuk kabi) qurilmalarda katta ahamiyatga ega. Bulardan tashqari vibratsiyaga o'ta chidamli.

Kamchiliklari sifatida shuni aytish mumkinki, uning axborot hajmi qattiq diskniki kabi katta bo'lolmayi. Yana har bir yacheykaga 1 million marta yozib-o'chirish bajarilgandan keyin ishdan chiqa boshlaydi.

**Tezkorlik, hajm va narx.** Ideal xotira tezkor, katta hajmli va arzon bo'lishi shart. Tezkor xotirani SRAMdan hosil qilish maqbul, lekin bu xotira qurilmasini narxi baland bolishiga olib keladi. Chunki bunda har bir yacheyka 6 ta tranzistordan tashkil topadi. Arzonroq variant uchun oddiy yacheykali RAMni qurish mumkin, lekin uning tezligi nisbatan past bo'ladi. Bundan shuni xulosa qilish mumkin: faqat qattiq disk yuqoridagi talablarni maksimal darajada qanoatlantiradi. Bunda RAM texnologiyasi qo'l keladi. SRAM mikrosxemalarini esa tezkor qurilmalarda, masalan flesh xotiralarda ishlatish mumkin.

Eng tezkor xotira – registrlar. Ular esa juda kichik hajmlidir. Keyingi darajada kesh xotira joylashadi. Uni markaziy protsessor ning sxemasiga joylashtiriladi. Odatda kompyuterlarda ikki darajali kesh ishlatiladi. 1-darajali kesh markaziy protsessor mikrosxemasida joylashadi. 2-darajali kesh esa 1-darajali kesh va asosiy xotira orasida

joylashadi. SRAM mikrosxemalari bilan tashkil qilinadi. Ba'zi hollarda esa markaziy protsessor umuman kesh xotiraga ega bo'lmasligi yoki har ikki darajadagi kesh ham markaziy protsessor sxemasida joylashgan bo'lishi mumkin.

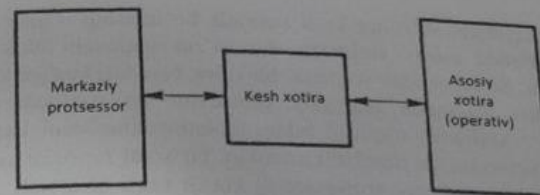
1-darajali kesh xotira keyingi darajasida asosiy xotira joylashadi. Uning hajmi kattaroq va dinamik xotira mikrosxemalari asosida SIMM, DIMM, RIMM modullari shaklida tashkil qilinadi. Tipik kompyuterlarda 1-darajali keshga murojaat vaqti bilan asosiy xotiraga murojaat vaqti 10 baravarga farqlanadi.

### 9. Kesh xotira. Kesh xotira xususiyatlari, asosiy va kesh xotiralarga murojaat qilish algoritmlari. Aralash va bo'lingan kesh xotira. Bir bosqichli va ko'p bosqichli kesh xotira

Agar markaziy protsessorning tezligi bilan asosiy xotiraning tezligini taqqoslasa juda katta farq ko'rsatkichi kelib chiqadi. Lekin markaziy protsessor asosiy xotiradan komandalar va ma'lumotlar kelishini uzoq vaqt kutib tura olmaydi. Shu sababli kerakli ma'lumotni olishga murojaat qilish vaqtini qisqartirish mexanizmlari zarur bo'ladi. Asosiy xotiraning tezligi kam ekanligidan, bu yerda arxitekturali yechim foyda beradi. Bu kesh xotira sifatida namoyon bo'ladi.

Keshlash mexanizmining samaradorligi kompyuter dasturlarining murojaatlarni lokallashtirish (локализация ссылок) xususiyatiga asoslanadi. Turli dasturlarni bajarilishini tahlil qilish shuni ko'rsatdiki, sarflanadigan vaqtning asosiy qismi takrorlanuvchi jarayonlarni bajarishga ketadi. Masalan oddiy va murakkab sikllar, protsedura va funksiyalar, va h.k. Bunda komandalarning bajarilish ketma-ketligi ahamiyatli emas. Muhimi ma'lum vaqt oralig'ida dastruning lokallashtirish sohasida ko'p bora takrorlanishidir. Dasturning qolgan qismiga murojaat kamdankam bajariladi, asosan 1-2 marta. Shu xolat murojaatlarni lokallashtirish deyiladi. Murojaatlarni lokallashtirish ham zamonda ham makonda ro'y beradi. Vaqt bo'yicha lokalizatsiya yaqinda bajarilgan komandalar yaqin vaqtda yana bajarilishini bildiradi. Makon bo'yicha lokalizatsiya esa hozir bajarilayotgan komandalardan keyin ularga adresi yaqin joylashgan komandalar bajarilishi mumkinligini bildiradi.

Agar dasturning faol segmentlarini tezkor kesh xotiraga joylashtirilsa, ularning bajarilish vaqti sezilarli darajada kamayadi. Komandalarni keshlash sodda tuzilishli bo'lib, xotirani boshqarish sxemalari shunday ishlab chiqiladiki, bunda murojaatlar lokalizatsiyasi xususiyatlaridan foydalanish ko'zda tutiladi. Vaqt lokalizatsiyasi bo'yicha markaziy protsessor murojaat qilayotgan har bir element (komanda, ma'lumot, va h.k) keshda unga yana bir marta murojaat qilingunicha saqlanib turadi. Makon lokalizatsiyasi bo'yicha esa, kesh xotiraga nafaqat so'ralayotgan adresdagi ma'lumotlar, balki atrofidagi bir necha ma'lumotlar ham uzatiladi. Ketma-ket adresli elementlar to'plami blok yoki kesh satri deyiladi.



9.1-rasm. Kesh xotira

Markaziy protsessor xotiradan o'qish komandasini uzatsa, xotira bloki berilgan adresdan bittadan so'zni keshga o'tkazadi. Qachon markaziy protsessor shu adresli so'zni o'qishga buyruq bersa, bu so'z keshdan qidiriladi. Odatda kesh xotirada bir necha blok ma'lumotlar mavjud bo'ladi va so'ralgan so'zni keshdan topilish ehtimoli juda yuqori hisoblanadi. Asosiy xotiradagi va keshdagi bloklarning mosligini akslanish funksiyasi aniqlaydi. Kesh xotira to'la bo'lsa va markaziy protsessor so'ragan so'z keshda emas, asosiy xotirada bo'lsa, keshni boshqaruvchi ta'minot keshga markaziy protsessor so'ragan so'z joylashgan blokni joyashtirish uchun qaysii blokni keshdan o'chirish kerakligini aniqlaydi. Bunday yechimni tanlash qoidalari to'plami almashtirish algoritmi deyiladi.

Markaziy protsessor keshni mavjudligini sezmaydi ham. U kerakli so'zni adresini o'qish/yo'zish uchun xotiraga yuboradi. Bunda keshni boshqarish sxemasi shunday so'z keshda mavjudligini tekshiradi. Mavjud bo'lsa, so'z uzatiladi. So'zni o'qishda asosiy xotiraga umuman murojaat qilinmaydi. Agar so'zni xotiraga yozish talab etilsa, tizim 2 xil yo'l tutishi mumkin. **1-dan** bevosita yozish protokolidagi kesh va asosiy xotira bir vaqtda yangilanadi deb nazarda tutiladi. **2-dan** ma'lumotlar faqat keshda yangilanadi va maxsus bayroq bitida belgilanadi. Bu bit o'zgarish bayroqchasi yoki modifikatsiya bayroqchasi deyiladi. Unga mos asosiy xotiradagi so'z keyinroq, shu so'z saqlanayotgan blok keshdan o'chirilayotganda yangilanadi. Bu texnologiya *teskari yozuv* yoki *teskari nusxalash* deyiladi. Bevosita yozish protokoli sodda, lekin unda ortiqcha, takroriy amallar ko'p bajariladi. Masalan keshdagi so'zni har gal yangilaganda, asosiy xotiradagi shu blokni saqlovchi blok ham yangilanadi. Teskari yozuvda ham shunga o'xshash xolat yuz beradi, chunki keshdagi blokda bir dona so'z o'zgartirgan bolsa ham, uni asosiy xotiraga yozishda blokni barchasini qayta yozish kerak bo'ladi.

Qidirilayotgan soʻzning kesh xotirada boʻlmasligi oʻqish xatoligi deyiladi. Bunda asosiy xotiradan shu soʻzni saqlovchi blok keshga koʻchiriladi. Soʻralayotgan soʻz esa, blokning barchasi keshga koʻchirib boʻlingandan keyingina markaziy protsessor ga uzatiladi. Baʼzan tegishli soʻz xotiradan oʻqilishi bilan, blokni yuklanishini kutmasdan markaziy protsessor ga uzatilishi mumkin, bu xolat *bevosita yuklanish* deyiladi, bunda markaziy protsessor ni kutish vaqti qisqaradi. Bunday sxemani amalga oshirish murakkabligi uchun amaliyotda qoʻllanilmaydi. Xotiraga yoziladigan soʻning keshda mavjud boʻlmasligi *yozish xatoligi* deyiladi. Bunday xolatda bevosita yozish protokoli bilan soʻz bevosita asosiy xotiraga yoziladi. Agar teskari yoziv protokoli ishlatilayotgan boʻlsa, keshga shu soʻz mavjud blok koʻchiriladi, keyin bu soʻz keshda qayta yozilib, oʻzgargan (MODIFIED) deb belgilanadi.

### 9.1. Akslanish funksiyasi

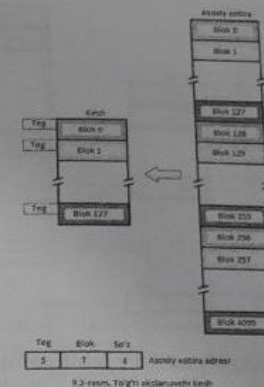
Asosiy xotiradan keshga maʼlumotlar bloki qanday koʻchirilishini misolda koʻramiz. Faraz qilaylik, bizda har biri 16 ta soʻzli boʻlgan 128 blokdan iborat, umumiy 2048 (2 K) soʻzli kesh va 16 razryadli adreslanuvchi asosiy xotira bor. Asosiy xotira hajmi 64 K soʻzli boʻlib, biz uni har biri 16 soʻzli 4 K blok sifatida qaraymiz. Hozircha ketma-ket adreslar ketma-ket soʻzlarni koʻrsatadi deb hisoblaymiz.

### 9.2. Toʻgʻri akslanish

Asosiy xotira va keshdagi bloklarni taqqoslashning eng oddiy usuli toʻgʻri akslanishdir. Bu texnologiyada asosiy xotiraning  $j$  bloki keshning  $j$  blokida akslanadi (9.2-rasm).

Bunda asosiy xotiraning 0, 128, 256, ... bloklari keshning 0 blokiga yoziladi. Asosiy xotiraning 1, 129, 257, ... bloklari keshning 1 blokiga yoziladi. Keshning har bir blokiga asosiy xotiraning bir necha bloklari yozilar ekan, hali toʻla boʻlmagan keshda ham pozitsiya uchun musobaqalashuv paydo boʻlishi mumkin. Masalan, dasturning 1-blokidagi komandalari oʻtish algoritmidan keyin 129-blokda ishini davom ettirishi mumkin. Dasturning bajarilishi mobaynida har ikkala blok keshning 1-blokiga koʻchirilishi shart boʻlib qoladi. Bu oson bajariladi: qaysi blok eskiroq boʻlsa, yangiroqʻi uning oʻrniga yoziladi.

Shunday qilib, blokning keshdagi joylashuvi uning asosiy xotiradagi adresidan kelib chiqadi. Xotiradagi adres 3 ta maydonga boʻlingan boʻlishi mumkin (5.15 rasm).



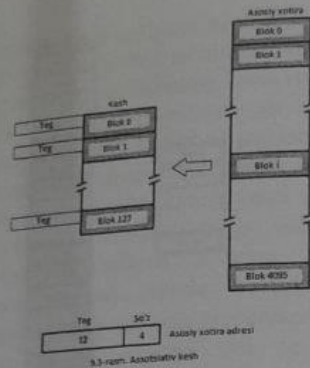
4 ta kichik razryad blokning 16 soʻzini beradi. Keshga yangi blok yozilganda blok nomerining 7 razryadli maydoni uning joylashuvini aniqlaydi. Yuqori 5 ta razryad blokning asosiy xotiradagi adresini aniqlaydi. Blokni keshga yozishda ular maxsus teg maydoni (deskriptorda) koʻrsatiladi. Bunday maydon keshning har bir bloki uchun mavjud boʻlib, u keshning joriy blokida saqlanayotgan 32 blokning qaysi biriga saqlanayotgan axborot mos kelishini aniqlaydi.

Dasturning bajarilish jarayonida markaziy protsessor adreslarni generatsiyalaydi, ularning har birida 7 razryadli blok nomeri maydoni undagi aniq blokni koʻrsatadi. Bu blokning tegi adresning yuqori 5 razryadi bilan taqqoslanadi, agar ular mos boʻlsa, kerakli soʻz keshda ekanligi maʼlum boʻladi. Aks holda kerakli soʻz mavjud blokni asosiy xotiradan keshga nusxalash lozim boʻladi. Bu toʻgʻri akslantirish texnologiyasi boʻlib, oʻzi soddagina lekin egiluvchanligi yoʻq.

### 9.3. Assotsiativ akslantirish

9.3-rasmda egiluvchan (moslashuvchan) akslantirish sxemasi keltirilgan. Bunga koʻra asosiy xotiraning bloki keshning ixtiyoriy blokiga koʻchirilishi mumkin. Bunda keshdagi blokni identifikatsiyalash uchun 5 emas, balki 12 bit zarur boʻladi. Dasturini bajarilishida markaziy protsessor generatsiya qilgan adresning teg bitlari navbatma-navbat keshning har bir blokining teg bitlari bilan taqqoslanadi



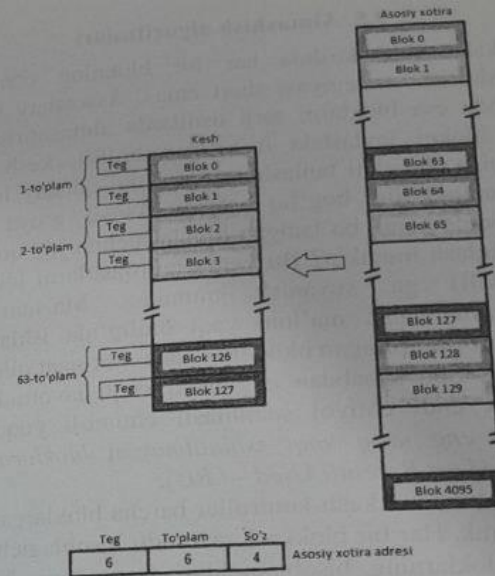


Moslik topilsa, qidirilayotgan so'z keshda mavjudligini anglatadi. Bunday texnologiya **assotsiativ akslanish** deyiladi. U blokka keshning ixtiyoriy joyida joylashish imkonini beradi, bu esa kesh maydonidan unumli foydalanishga imkon yaratadi. Eski blokni keshdan o'chirish uchun kesh to'lgan bo'lishi talab qilinadi va bu maqsad uchun o'chiriladigan blokni tanlasj algoritmi ishlatiladi. Bunday algoritmlar ancha ko'p. (mavzuning davomida ko'riladi). Keshdan blokni qidirish jarayoni **assotsiativ qidiruv** deyiladi. Uning tezligini oshirish uchun esa teglarni parallel qarab chiqish mexanizmi zarur.

#### 9.4. To'plam-assotsiativ akslanish

To'g'ri akslanish va assotsiativ akslanish birgalikda ishlatilishi ham mumkin. Bunda kesh bloklari to'plamlarga birlashadi, va asosiy xotiraning har bir bloki to'plamning ixtiyoriy blokida joylashishi mumkin. To'g'ri akslanishdagi konflikt (munozara?) lar soni keskin kamayadi. **To'plam-assotsiativ akslanish** deb nom olgan bu kesh to'liq assotsiativ keshdan ko'ra arzon, chunki unda assotsiativ qidiruv maydoni kichik. To'plam-assotsiativ akslanishga misol. Aytaylik har birida 2 ta blok bo'lgan 64 ta to'plamli kesh bo'lsin (9.4-rasm).

0, 64, 128, ..., 4092 xotira bloklari 0 to'plamda akslanadi va to'plamning ixtiyoriy ikkita dan bir holatini egallaydi. 64 ta to'plam blokining mavjudligi so'z adresidagi 6 razryadli to'plam maydoni keshning qaysi to'plami kerakli so'zni saqlashini aniqlaydi. Adres tegi maydoni assotsiativ ravishda topilgan to'plamning ikkita bloki bilan taqqoslanadi va mos kelsa mos blok keshda mavjudligini bildiradi. Bunday qidiruv mashina uchun juda sodda hisoblanadi.



To'plamdagi bloklar soni har bir kompyuterning talabiga mos bo'ladi. To'plamda  $n$  ta blok mavjud bo'lsa, bunday kesh  **$n$ -kanalli to'plam-assotsiativ kesh** deyiladi.

Har bir blok uchun keshda yana bitta boshqaruv biti mavjud bo'lishi kerak, u **ishonchlilik biti** deyiladi. U mazkur blok ishonchli axborotni saqlashini aniqlaydi. Uni o'zgarish biti bilan adashtirmang, o'zgarish biti blokda ma'lumotlar MODIFIED bo'lganligini aniqlaydi. O'zgarish biti faqatgina to'g'ridan-to'g'ri yozuv (сквозная запись) ishlatilmaydigan tizimlarda kerak. Tizim ishga tushirilganda yoki asosiy xotiraga tashqi qurilmadan dastrular, ma'lumotlar yuklanganda barcha ishonchlilik bitlari 0 ga tenglanadi. Qattiq disk va asosiy xotira orasidagi almashinuv xotiraga to'g'ri murojaat mexanizmi bilan amalga oshiriladi. Odatda bu ma'lumotlar keshni aylanib o'tadi. Asosiy xotiradan blok keshga birinchi marta yuklanganda blokning ishonchlilik biti 1 ga tenglanadi. Agar asosiy xotira bloki keshni ishlatmasdan boshqa manbadan yangilansa, tizim shunday blok keshda mavjudligini tekshiradi. Mavjud bo'lsa, ichonchlilik biti 0 ga tenglanadi, chunki keshda eskirgan ma'lumot saqlanishi kerak emas.

tizimlarda bir necha yuzlagan, minglagan markaziy protsessor lar maxsus mashinalar boshqaruvi ostida ishlaydi. Bu mashinalar masalalarni mashinalarga taqsimlaydi, ularning o'zaro muloqotini va ma'lumot almashinishini nazorat qiladi. Har bir markaziy protsessor oddiy yoki past unumdorlikli bo'lishi mumkin. Bunday tipdagi birinchi mashina CM-1 (Connection Machines-1) bo'lib, 1 bitli arifmetik-mantiqiy qurilma lardan tuzilgan va 32 bitli qo'shish amali 24 mks (mikrosekund) da bajarilgan. arifmetik-mantiqiy qurilmalar markaziy protsessor ga birlashgan, bitta markaziy protsessor ning unumdorligi 2 MIPS ga teng. markaziy protsessor lar soni 64 mingtaga yetgan, ular 4 ta blokk bo'lingan, har bir blok boshqa bloklar bilan ma'lumot almashishni ta'minlovchi mashina tomonidan nazorat qilingan, butun tizimni esa yuqori darajali komandalar potokini mashina tiliga translyatsiya qiluvchi mashina boshqargan. Bunday bitta mashina 2 mln tagacha markaziy protsessor lar ishini nazorat qila olgan. CM-1 mashinasi 100Gflops unumdorlikka ega bo'lsa (1984-yil!), CM-5 esa 256 ming markaziy protsessor bilan 1Tflops unumdorlikni bergan (1991!).

Bunday tizimlarda hisoblashlar unumdorligini oshirishda (umuman barcha parallel hisoblash tizimlarida) markaziy protsessor lar orasida axborot almashishni tashkil qilish jarayoni hall qiluvchi ahamiyatga ega. Bu yerda alohida shuni ta'kidlash lozimki, masalni qismlarga bo'lishdagi "yiriklik darajasi" hisoblashlarni bajarishga va markaziy protsessor lar orasidagi bog'lanishni tashkil qilishga ketadigan iqtisodiy hisob-kitoblardan kelib chiqadi. Ba'zi hollarda yuqori daradagi "maydalik"ni talab etuvchi yuqori darajadagi parallellik bunday "maydalik" ni tashkil etishdagi ko'pxarajatlilik bilan nizolashadi (konflikt), chunki markaziy protsessor lar orasidagi bog'lanishlar ko'payib ketadi. Bunday vaziyatda "yirik" lahgan texnologiyani qo'llash iqtisodiy muvofiqroq hisoblanadi.

### 11.3. Konveyerli hisoblash. Superskalyar hisoblashlar

**Komandalar konveyeri.** markaziy protsessor ning turli komandalarni bajarishga turlicha vaqt sarflashi ma'lum. Agar turli komandalarni bir xil vaqtda bajarilishini sun'iy usulda ta'minaladigan bo'lsa (fiksatorlar kiritish orqali yoki sinxronlashtirish yordamida) bu usul bilan komandalar konveyerini tashkil qilish mumkin.

Bitta komandani bajarilish vaqtini sun'iy yo'l bilan ko'paytilsa, konveyer to'liq bo'lgan holatda unumdorlik  $1/T_k$  ga yaqin bo'ladi.

Bunda  $T_k$  – konveyer takti, bu holatda bir etapni bajarish vaqti. Bu markaziy protsessor ning unumdorligini 5 barobar oshirish imkonini beradi. Bu – nazariy jihatdan qaraganda. Amaliyotda esa bunga erishish qiyin, chunki konveyerlashdagi nizolar (konflikt) bunga yo'l qo'ymaydi.

Komandalar konveyerlashdagi nizo deganda potokdagi keyingi kelayotgan komandani shu komanda uchun ajratilgan takt davomida bajarila olmay qolishiga aytiladi. Nizolar 3 turli bo'ladi:

1. Strukturali yoki resursli nizolar. Barcha komandalar kombinatsiyalarini bir vaqtda konveyerda joylashtirish va bajarishda apparat vositalar bajara olmay qolganda yuzaga keladi. Markaziy protsessor dagi qaysidir qurilma konveyerlanmagan yoki yagona nuxxada mavjud bo'lsa (parallellanmagan bo'lsa) strukturali yoki resursli nizo kelib chiqadi. Masalan kesh xotira bilan bog'liq shunday nizo bo'lishi mumkin: bitta komandani keshdan o'qib olish jarayoni kechayotganda (komandani bajarishning 1-bosqichi) boshqa komanda hisoblash natijasini keshga yozishga uringanda. Bunday resursli nizolarni kelib chiqmasligi uchun 3 xil uslub ishlatiladi:

- Nizo hal bo'lgunicha konveyerni to'xtatib turish (pipeline stall)
- Nizo kelib chiqadigan qurilmani ikkilantirish, masalan kesh xotirani 2 ga: komandalar keshi va ma'lumotlar keshiga ajratish
- Nizo keltirgan uskunani tezlatish (tezkorroq'iga almashtirish) yoki konveyerlash

Nizolarni hal qilishning oxirgi 2 ta usulini juda kam hollarda qo'llaniladi. Shunda ham nizo kelib chiqish chastotasi ortsa. Chunki qo'shimcha harajatlar 1-xolatdagi konveyerning to'xtab-to'xtab ishlashiga qaraganda iqtisodiy nomuvofiqdir.

2. Dasturiy yoki axboriy nizolar. Ikkita guruhga ajratiladi:

- a... ma'lumotlar bo'yicha nizo. Qachonki keyingi komandaning bajarilishi uchun oldingi komandaning bajarilish natihasi zarur bo'lsa
- b... boshqarish nizolari. Komandalarning tabiiy bajarilish tartibi buzilganda yuzaga keladi. Masalan shartli o'tish operatorini ishlatishda. Ma'lumotlar bo'yicha nizo bir necha turi bo'lishi mumkin.

**Ma'lumotlar bo'yicha nizolar, 1-holat.** RAW – Read After Write – oq'ishdan keyin yozish nizosi. Aytaylik A1 va A2 komandalari mavjud va A2 komanda A1 dan keyin keladi. A2 komanda A1 ning natijasini ishlatganda, hali A1 hisoblashni yakunlamagan bo'lsa A2 bo'sh joyga murojaat qiladi va RAW kelib chiqadi. Masalan:

ADD R1, R2, R3  
SUB R4, R5, R1

IF	ID	M	ME	EX	WB
	IF	ID	M	ME	EX

Bu yerda A1 komandaning hisoblash natijasi (1:5) da, A2 ga kearkli komanda (2:4) da. Ko'rinib turibdiki, A2 hali mavjud bo'lmagan natijaga murojaat qilmoqda va shu joyda nizo kelib chiqadi.

**Ma'lumotlar bo'yicha nizolar, 2-holat.** WAR – Write After Read.

A1 komanda hali ma'lumotni o'qimagan holatda A2 komanda hisoblash natijasini yozib qo'yishidam sodir bo'ladi. Bunday nizo faqat shunday xolatda sodir bo'lishi mumkin: tartiblanmagan qayta ishlovchi konveyerlarda A2 komanda ma'lumotni uzatishda yoki ishni yakunlashda A1 dan oldinga o'tib ketadi.

**Ma'lumotlar bo'yicha nizolar, 3-holat.** WAW – Write After Write. A2 komanda natijani A1 dan oldin yozib qo'ysa va bu komandalar orasida ushbu adresni tekshiruvchi yana boshqa komanda ham bo'lsa sodir bo'ladi. Komandalarning tartiblanmagan xolda bajarilishida yuzaga keladi.

Ma'lumotlar nizolarini hal qilish usullari:

Ma'lumotlar bo'yicha nizolar, 1-holat. Konveyerni to'xtatish

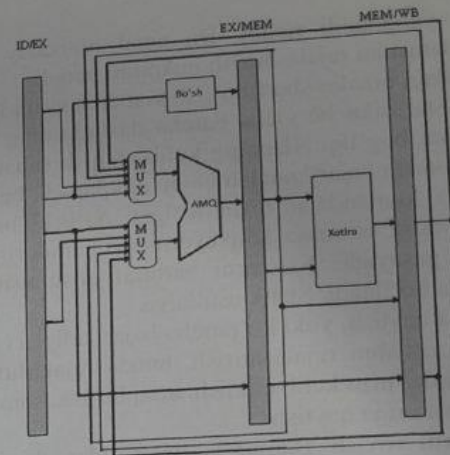
Ma'lumotlar bo'yicha nizolar, 2-holat. Aylanib o'tish va oldinga surish mexanizmini (data bypassing & forwarding) joriy qilish

Ma'lumotlar bo'yicha nizolar, 3-holat. Konveyerni kompilyator orqali yuklash (statik optimallashtirish)

Ma'lumotlar bo'yicha nizolar, 4-holat. Komandalarni markaziy protsessor da tartibsiz bajarish (dinamik optimallashtirish)

Ma'lumotlar bo'yicha nizolar, 5-holat. Registrlarni qayta nomlash

Bulardan Ma'lumotlar bo'yicha nizolar, 2-holat ni qaraymiz. Aylanib o'tish va oldinga surish mexanizmi bir funksional blokning chiqish natijasini bevosita boshqa blokning kirishiga uzatishni bildiradi. 11.1-rasmda natijalar arifmetik-mantiqiy qurilmadan (bufer registri orqali) bevosita uzatilishi ko'rsatilgan. Bunda xotiraga yoki registrga natijalarni yozish zarurati yo'qoladi.



11.1-rasm. Hisoblash natijalari arifmetik-mantiqiy qurilmadan (bufer registri orqali) bevosita uzatilishi

Ma'lumotlar bo'yicha nizolar, 3-holatda dastur kompilyatsiyalanishi vaqtida komandalar bir-biri bilan imkon boricha nizoga kirishmasligi nazarda turuib loyihalanadi. Masalan:

$$A := B + C;$$

$$D := E + F.$$

Bu komandalar potokida RAW tipli nizo 2-,4-,7-,8-komandalardan keyin kelib chiqadi. Xotiraga yozishdagi nizolarni aylanib o'tish mexanizmi bilan bartaraf etish mumkin, o'qishdagi nizolarni esa komandalarni qayta tartiblash orqali hal qilish mumkin.

Ma'lumotlar bo'yicha nizolar, 4-holatda komandalarni dinamik optimallashtirishda nizo keltirib chiqaruvchi komandalar konveyerda qolishi mumkin, bunday komandalarga bog'liq bo'lmagan boshqa komandalar esa uni (konveyerni) aylanib o'tishi mumkin. Nizoli holatlarni tahlil qilish va qayta tartiblanadigan komandalarni tanlash markaziy protsessor dagi konveyerni boshqarish mantiqiga topshiriladi.

Ma'lumotlar bo'yicha nizolar, 5-holatda komandalardagi registrlarning mantiqiy nomlari markaziy protsessor ning fizik registrlarida dinamik akslanadi. Natijada bitta mantiqiy registrga tegishli bo'lgan turli ma'lumotlar turli fizik registrlarga yoziladi. Registrlararo moslik akslanish jadvalida beriladi. U esa har bir komanda

dekodlanganda yangilanib turadi. Bu amal markaziy protsessor ga konveyerni yuklanishini rejalashtirish imkonini beradi.

Boshqarishdagi nizolar shartli o'tish komandalarini konveyerlashda yuzaga keladi. Statistika bo'yicha barcha dasturlarning 15-20% qismi shartli o'tish bilan bog'liq. Nizoning kelib chiqish sababi shuki, sharti o'tish adresi komanda bajarilgandan keyin aniqlanadi, bu vaqtgacha esa konveyer boshqa komandalar bilan to'lgan bo'lishi mumkin. Paydo bo'lgan nizo bartaraf etilmasa konveyerning unumdorligi 2 va undan ortiq barobarga pasayadi. Nizolarni bartaraf etish usullari statik va dinamik usullarga bo'linadi. Statik usullarga

ST1 – orqaga qaytish, yoki ko'pincha bajariladigan (yoki ko'pincha bajarilmaydigan) o'tishni rejalashtirish, bunda rejalashtirish natijasida (prognozlashtirishda) nizo kelib chiqishi aniq bo'lsa, konveyerni tegishli qismini tozalash va ortga qaytish

ST2 – sikllarni yoyish. Barcha sikllar sikldan chiqish sharti bilan bog'liq. Agar sikl soni katta bo'lmasa va takrorlanishlar soni ma'lum bo'lsa, usbu sikli konveyerga takrorlanishlar soniga mos sonda yoyish, bundan maqsad sikl shatini tekshirmaslik

ST3 – to'xtatib turish slotlari. Bunda dasturning shartli komandaning bajarilishidan keyingi o'tishigacha bajarilib bo'ladigan ba'zi qismlari tushuniladi. bu usulning g'oyasi shundaki, to'xtatish slotlarini "foydali" yoki "zararsiz" komandalar bilan to'ldirib turish. Bu komandalar yoki shartning bajarilishiga bog'liq bo'lmaydi, yoki takror bajarilsa ham hech qanday o'zgarish sodir bo'lmaydi.

ST4 – dastur profili asosida o'tishlarni oldindan bashorat qilish. Dasturni ko'p bora ishlatilishi natijasida hosil qilingan statistik ko'rsatkichlar asosida o'tishlarni prognozlashtirish.

Nizolarni bartaraf etish dinamik usullarga quyidagilar kiradi:

DN1 – o'tish adresi ma'lumbo'lunicha konveyerni to'xtatib turish

DN2 – shartli o'tish komandasining o'tish adresini komanda bajarilishi boshlanishi bilan aniqlash

DN3 – markaziy protsessor ning tarmoqlanishini dinamik prognozlash. Bu amal BPB (Branch Predicting Buffer) yordamida bajariladi. Unda asosan  $n$ -razryadli ikkili ko'rinishdagi prognoz hisoblagichi ishlatiladi. Har bir o'tish bajarilganda shu o'tish uchun prognoz hisoblagichi qiymati 1 ga oshadi, o'tish bajarilmasa 1 ga kamayadi. Agar hisoblagichning qiymari  $2^{n-1}$  dan katta bo'lsa, o'tish bajariladi deb prognoz qilinadi, aks holda o'tish bajarilmaydi deb

prognoz qilinadi. Asosan 1 bit yoki 2 bitli hisoblagich ishlatiladi. Bunda to'g'ri prognoz berish 1 bitlida 70%, 2 bitli da 85% ga teng bo'ladi.

Prognozni yanada tezlashtirish uchun BTB (Branch Target Buffer) – maqsadli adresli o'tish buferi ishlatiladi. BTB – assotsiativ kesh xotira bo'lib, teglari sifatida dasturning joriy qismidagi tarmoqlanish komandalari adreslari ishlatiladi, yacheykalarida esa prognoz hisoblagichlari va shart bajarilganda maqsadli o'tish adreslari joylashadi. markaziy protsessor bajariladigan komandani tanlaganda uning adresi BTB da bor-yo'qligini tekshiradi, prognoz hisoblagichini o'qiydi va uning qiymatidan kelib chiqqan holda keyingi komandani keyingi adresdan yoki BTB dagi adresdan o'qishni tanlaydi.

#### 11.4. Superskalyar arxitektura

Komandalar konveyerini ishlatish CPI ning ko'rsatkichini 1 gacha tushirish imkonini beradi, boshqacha aytganda har bir taktida bitta "bajarilgan" komanda chiqadi. Bunday holatda takt uzunligi 10 ns (nanosekund) bo'lgan (takt chastotasi 100 MHz) markaziy protsessor ning unumdorligi 4 barobar oshib, 100 MIPS ga teng bo'ladi. Lekin, 1-dan Celeron markaziy protsessor da 250 MIPS bor, 2-dan CPI=1 ko'rsatkichi konveyerlashdagi nizolar sababli bajarilmaydi, nari borsa CPI=1,5 yoki CPI=2 bo'lishi mumkin. Bunday CPI bilan Celeron va shu kabi arxitekturali markaziy protsessor larning yuqori unumdorligi qanday ta'minlanadi?

Buning uchun ularda *superskalyar qayta ishlash* qo'llaniladi. Bu degani komandalarni ko'p potokli qayta ishlash bo'lib, bir taktida 1 dan ko'p komanda bajariladi (CPI<1).

Aslida superskalyar jarayonda bir nechta komandalar potoki bir nechta qurilmalardan teng o'tadi, boshqa yondashuvlar esa u yoki bu usulda bir potok bilan ishlaydi. Dekodlash, tanlash, RISC-simon ko'rsatma shakliga o'tkazish, qayta tartiblash kabi potok tezliklari va bajaruvchi qurilmalardagi potok tezliklarini moslashtirish uchun shu qurilmalarning ko'rsatma navbatlaridan (FIFO buferidan) foydalaniladi.

Superskalyar markaziy protsessor larda joriy dasturining komandalarining ketma-ket potokini parallel potok triadasiga (*amallar+operandlar+natijani belgilash*) aylantirish va shu bilan bir vaqtda, parallel ravishda ijrochi qurilma komandalar ketma-ketligini bajarishida juda murakkab o'zgartirishlar bajariladi.

## 12. Hyper-Threading texnologiyasi. Multitasking va ko'p oqimli tizimlar. VLIW, SMT texnologiyalari

Hyper-Threading (HT, giperaniqlik) texnologiyasi dastlab 2002-yilda Pentium 4 protsessorlarida ishlab chiqilgan. Shu davrdan keyingi ishlab chiqilgan barcha markaziy protsessorlarda bu texnologiya ishlatilgan deya olmaymiz, chunki Pentium 4 Xeon, Intel Core, Intel Athomlarda bu texnologiya mavjud bo'lsa, ulardan keyin ishlab chiqilgan Intel Core 2 Duo, Intel Core Quadlarda esa ishlatilmagan. Ba'zan bu texnologiyani mashina unumdorligini 2 barobarga oshiradi degan tushuncha mavjud. Aslida, 2 barobar bo'lmasa, unumdorlikni bir qancha (1,5-1,7 barobar) ko'rsatkichga oshirishi aniq.

### 12.1. Texnologiyaning xarakteristikalari

Intel shunday xarakteristika beradi: *Intel Hyper-Threading (Intel HT) har bir yadroga bir necha potok ma'lumotlarni qayta ishlab, markaziy protsessorning resurslaridan yanada unumdorroq foydalanish imkonini yaratadi. Unumdorlik jihatidan bu texnologiya protsessorning o'tkazuvchanlik qobiliyatini oshirib, ko'p potokli ilovalarning umumiy tezligini oshiradi.*

Bu izohda asosiy e'tibor qiladigan jihati - bir yadroga bir necha potok ma'lumotlarni qayta ishlashidir. Qanday amalga oshiriladi? Hyper-Threading texnologiyali protsessor: 1) bir vaqtda bir necha potoklarda bajarilayotgan jarayonlar haqidagi axborotni saqlaydi; 2) har bir mantiqiy protsessor uchun bitta registrlar to'plami va bitta uzilish kontrolleri to'plamiga ega bo'ladi. Registrlar to'plami deganda protsessor ichiga joylangan tezkor xotira bloklarini tushunish va uzilish kontrolleri to'plami deganda turli qurilmalardan tezlik bilan bajarilishi talab etiladigan ma'lumot kelganda shu ishni bajarishni ta'minlovchi integrallangan blok tushunish lozim.

Aytaylik, markaziy protsessor oldida 2 ta masala turibdi. Agar protsessorida 1 ta yadro bo'lsa, bu masalalar ketma-ket bajariladi. Agar 2 ta yadro bo'lsa, parallel ravishda har biri bir yadroga bajariladi va ikkita masalaning bajarilish vaqti ko'p vaqtda bajarilgan masalning vaqtiga teng bo'ladi. Agar protsessor bir yadroli bo'lsa va Hyper-Threading ni bo'lsa jarayon qanday bo'ladi? Ma'lumki, masalani yechish jarayonida markaziy protsessor 100% shu masala bilan band bo'lmaydi, balki markaziy protsessorning qaysidir bloki bu masalada umuman ishlatilmaydi, qaysidir joyda xatolik sodir bo'ladi va qaysidir blok xatolikni qayta ishlaydi va h.k. umuman olganda jarayonda

bevosita masala bilan markaziy protsessorning qandaydir 70% qismi band bo'ladi xolos. Hyper-Threading texnologiyasi esa ana shu bo'sh qolgan protsessor qismlariga ikkinchi masalani yuklaydi va shu usulda unumdorlik paydo bo'lmaydi, chunki juda katta ehtimol bilan shuni aytilish mumkinki, hisoblash bloki har ikkala jarayonga bir vaqtda albatta kerak bo'lib qoladi. Shu joyda ikkinchi jarayon birinchi jarayon yakunlanguncha to'xtab (kutib) turadi.

Natijada ikkita masalani yechish uchun sarflanadigan vaqt shu ikkita masalani parallel yechishga ketadigan vaqtdan ko'ra ozroq va eng qiyin masalani yechishga sarflanadigan vaqtdan ko'proq bo'ladi.

### 12.2. Texnologiyaning yutuqlari va kamchiliklari

Hyper-Threading texnologiyasini qo'llovchi markaziy protsessor uni qo'llamaydigan markaziy protsessor dan hajm jihatidan o'rtacha 5% ga katta bo'ladi, chunki qo'shimcha registrlar va uzilishlar kontrolleri taxminan shuncha joy egallaydi. Hyper-Threading texnologiyasini qo'llash markaziy protsessorni 90-95% ga unumdorligini oshirsa, o'sish ko'rsatkichi  $95\% - 70\% = 20\% \div 30\%$  bo'ladi va bu ancha yuqori ko'rsatkich hisoblanadi.

Lekin, bu hammasi faqat yaxshi degani emas. Ba'zan Hyper-Threading texnologiyasi unumdorlikni umuman oshirmaydi, ba'zan esa sekinlashtiradi ham. Buning sabablari:

a) kesh xotiraning hajmining yetishmasligi. Misol uchun, zamonaviy 4 yadroli i5larda 6 MB li 3-darajali kesh mavjud, har bir yadroga 1,5 MB. 4 yadroli i7larda 8 MB bo'lib, mantiqiy yadrolar 8 ta ekanligidan, har yadroga 1 MB to'g'ri keladi va unumdorlikni pasayishi ko'rinib turibdi.

b) DT ni (dasturiy ta'minotni) optimallashtirilmaganligi. Asosiy muammo, dasturlar mantiqiy yadroni fizik yadro deb hisoblaydi va buning natijasida masalalarni parallel bajarishda bitta yadroning hisoblash blokiga ko'p komandalar murojaat qilishi natijasida kutib qolish ro'y beradi. Bu esa unumdorlikni pasayishiga olib keladi, garchi Hyper-Threading texnologiyasi mavjud bo'lsa ham.

d) Ma'lumotlarning bog'liqligi. Shundayki, ikkinchi komandani bajarish uchun birinchi komandaning bajarilish natijalari kerak, u esa hali bajarilish jarayonida (yakunlanmagan).

**Giper-aniqlik bilan ishlay oladigan dasturlar:** bunday dasturlar juda ko'p, chunki Hyper-Threading texnologiyasi uchun bunday

dasturlar ishlatishga juda qulaydir. Umuman Hyper-Threading texnologiyasi alohida ko'rsatkichlar, xususan protsessor hajmi, elektr energiya va b larni talab qilmas ekan, jarayonni to'g'ri tashkil qilishda 30% gacha o'sishga erishish mumkin. Shu sababli Hyper-Threading texnologiyasini jarayonlarni parallellashda mohir bo'lgan arxivator dasturlar (WinRAR), grafik redaktorlar (Photoshop, CorelDraw), 2D/3D modellash dasturlari (3D Max) tezda o'zlashtirib oldi.

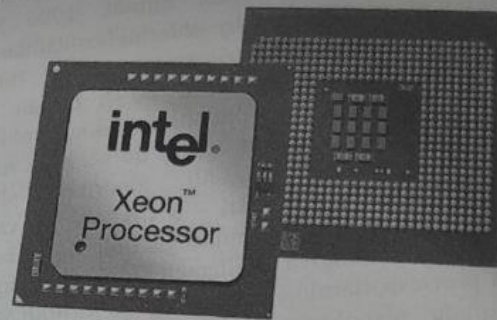
Giper-aniqlik bilan ishlay olmaydigan dasturlar odatda o'yin dasturlari Hyper-Threading texnologiyasini qo'llay olmaydi, chunki ularni parallellashtirish imkoniyati juda kam. Shu sababdan ular uchun yuqori chastotali 4 ta yadro yetarli hisoblanadi. Lekin, o'yin bilan bir vaqtda bajariladigan fon dasturlari ham borki, agar o'yin dasturi 4 ta yadroni to'liq band qilsa, fon dasturlari uchun hisoblashlar yana shu markaziy protsessor yadrolariga tushadi, natijada o'yin tezligi pasayadi. Bunday vaziyat uchun i7 protsessori juda qo'l keladi, chunki mantiqiy 8 ta yadroga bo'lingan mikroprotsessorda barcha fonli jarayonlar ustunlikning quyi darajasi beriladi, shuning uchun ham o'yinlar o'ynash uchun kompyuter tanlaganda i7 protsessorli Hyper-Threading texnologiyasiga asoslangan mashinani tanlash lozim.

Agar kompyuterda o'rnatilgan protsessor tegishli standartni qo'llab-quvvatlasa, Hyper-Threading odatda, avtomatik ravishda faollashadi. Ammo ba'zi hollarda siz Hyper-Threading texnologiyasining ishlashi uchun kerakli shartlarni qo'lda bajarishingiz kerak.

Buning uchun dastlab BIOS interfeysiga kirish kerak. BIOS interfeysida Hyper-Threading elementini topish kerak: uni qo'llab-quvvatlovchi I/O tizimining versiyalarida odatda u aniq joyda joylashgan. Tegishli variantni tanlagandan so'ng, Enter tugmasini bosib va uni yoqilgan deb belgilab faollashtiring. Agar ushbu rejim allaqachon o'rnatilgan bo'lsa, u holda Hyper-Threading ishga tushadi. Foydalanuvchi uning barcha afzalliklaridan foydalanishi mumkin. Sozlamalarda texnologiyani faollashtirgandan so'ng, "O'rnatishni saqlash va chiqish" (Save changes and exit) elementini tanlab, barcha o'zgarishlarni BIOS-da saqlashingiz kerak. Shundan so'ng, kompyuter Hyper-Threading yordami bilan protsessor ishlaydigan rejimda qayta boshlanadi. Hyper-Threading-ni o'chirib qo'yish ham shunga o'xshash tarzda amalga oshiriladi.

**Hyper Threading protsessorlari.** Kompaniyaning bu konsepsiyasini amalga oshirgan birinchi protsessor, ba'zi manbalarga ko'ra, Intel Xeon markaziy protsessor dir. Bir qator arxitekturalarda

ushbu mikrosxemalar Pentium 4 ga o'xshaydi, keyinchalik ushbu texnologiya qolgan texnologiyalarda ham amalga oshirildi. Keyinchalik, ko'p qirrali hisoblash funksiyasi Prestonia yadroli Xeon server protsessorlarida amalga oshirildi (12.1-rasm).



12.1-rasm. Prestonia yadroli Intel Xeon server protsessori (old va orqa tomondan ko'rinishi)

Hozirgi kunda Hyper-Threading tarqalishi eng mashhur mikrosxemalar orasida Core, Xeon va iX oilalariga tegishli. Shunga o'xshash algoritmlar Itanium va Atom kabi protsessorlarda amalga oshirilgan.

Hyper-Threading, uni qo'llab-quvvatlovchi protsessorlar haqida asosiy ma'lumotlarni o'rganib chiqib, texnologiyaning rivojlanish tarixiga oid ba'zi bir faktlarni ko'rib chiqamiz.

**Rivojlanish tarixi.** Yuqorida ta'kidlaganimizdek, Intel 2001 yilda ushbu konsepsiyani e'lon qildi. Ammo texnologiyani yaratish bo'yicha dastlabki urinishlar 1990-yillarning boshlarida amalga oshirildi. Intel kompaniyasi muhandislari bir qator operatsiyalarni bajarishda kompyuter protsessorlari resurslaridan to'liq foydalanilmayotganligini payqab qolishgan.

Intel mutaxassislari tomonidan hisoblab chiqilganidek, foydalanuvchi shaxsiy kompyuterda ishlayotganda, mikrosxema ma'lum bir vaqt oralig'ida (taxminan 30%.) juda faol ishlatilmaydi.

Biroq, aksariyat IT-mutaxassislar, protsessorning 70 foizli quvvati ishlaymay qolsa ham, ammo bu juda katta miqdordagi ish hajmidir, degan fikrga kelishdi.

### 12.3. Ishlab chiquvchilarning asosiy vazifasi

Intel ushbu holatni asosiy kompyuter chiplari samaradorligini ta'minlashga yangi yondashuv bilan bartaraf etishga qaror qildi. Protessorlarning imkoniyatlaridan faolroq foydalanishga yordam beradigan texnologiyani yaratish taklif qilindi. 1996 yilda Intel mutaxassislari bu borada o'zining amaliy ishlarini boshladilar.

Intel korporatsiyasining konsepsiyasiga ko'ra, bitta dastur ma'lumotlarini qayta ishlovchi protsessor bo'sh turgan resurslarni boshqa dastur bilan ishlashga yo'naltirishi mumkin (yoki amaldagi tarkibiy qismi, lekin boshqa tuzilishga ega va qo'shimcha resurslardan foydalanishni talab qiladi). Tegishli algoritm, shuningdek, kompyuterning boshqa apparat qismlari - RAM, chipset va dasturlar bilan samarali o'zaro aloqani o'z zimmasiga oldi. Intel bu vazifani uddalay oldi. Dastlab texnologiya Willamette deb nomlangan. 1999 yilda u bir nechta protsessorlarning arxitekturasiga kiritildi va sinovlar boshlandi. Tez orada texnologiya hozirgi nomini oldi - Hyper-Threading.

### 12.4. Dasturiy ta'minot va apparat omillari

Windows OS -ning eski versiyalari - Windows 98, NT va nisbatan eskirgan XP-ga kelsak, Hyper-Threading bilan mos kelish uchun zarur shart ACPI-ni qo'llab-quvvatlashdir. Agar u OS-da amalga oshirilmasa, unda tegishli modullar tomonidan yaratilgan barcha hisoblash oqimlari kompyuter tomonidan tan olinmaydi. E'tibor bering, Windows XP odatda ushbu texnologiyadan foydalanadi. Shuningdek, multithreading algoritmlari kompyuter egasi tomonidan qo'llaniladigan dasturlarda amalga oshirilishi keng tarqalgan.

Ba'zan, dastlab o'rnatilgan va texnologiyaga mos kelmaydigan protsessorlarning o'miga Hyper-Threadingni qo'llab-quvvatlaydigan protsessorlarni o'rnatgan bo'lsangiz, sizga mos kompyuter kerak bo'lishi mumkin. Ammo, operatsion tizimlarda bo'lgani kabi, foydalanuvchi o'z ixtiyorida zamonaviy shaxsiy kompyuterga yoki hech bo'lmaganda birinchi Hyper Threading protsessorlarining mos keladigan tarkibiy qismlariga ega bo'lsa, biz yuqorida ta'kidlaganimizdek, Core liniyasida amalga oshirilgan va unga moslashtirilgan bo'lsa, hech qanday muammolar bo'lmaydi.

### 12.5. Hyper-Threadingning amaliy yordami

Ushbu texnologiya Intel protsessorlari uchun sezilarli darajada farq qiladimi? Bu borada turli xil fikrlar mavjud. Ammo ko'pchilik Hyper-Threading texnologiyasi shunchalik ommalashganki, bu echim server tizimlarining ko'plab ishlab chiqaruvchilari uchun ajralmas bo'lib qoldi va bu oddiy kompyuter foydalanuvchilari tomonidan ham ijobiy qabul qilindi.

### 12.6. Uskuna ma'lumotlarini qayta ishlash

Texnologiyaning asosiy afzalligi shundaki, u apparat formatida amalga oshiriladi. Ya'ni, hisob-kitoblarning asosiy qismi protsessor ichida asosiy chip yadrosi darajasiga uzatiladigan dasturiy ta'minot algoritmlari shaklida emas, balki maxsus modullarda amalga oshiriladi - bu kompyuterning umumiy ishlash ko'rsatkichlarining pasayishini anglatadi. Umuman olganda, IT mutaxassislarining fikriga ko'ra, Intel muhandislari texnologiyani ishlab chiqishda aniqlagan muammo - protsessorning ishlashini yanada samaraliroq qilishni hal qilishga muvaffaq bo'lishdi -. Darhaqiqat, testlar ko'rsatganidek, foydalanuvchi uchun juda ko'p muhim vazifalarni hal qilishda Hyper-Threading-dan foydalanish ishni sezilarli darajada tezlashtirishga imkon berdi.

Shuni ta'kidlash kerakki, 4 ta texnologiya uchun qo'llab-quvvatlovchi modullar bilan jihozlangan mikrosxemalar birinchi modifikatsiyalarga qaraganda ancha samarali ishladi. Bu, asosan, kompyuterning bir nechta turli xil turlari mavjud bo'lganda, haqiqiy ko'p vazifali rejimda ishlash qobiliyatida namoyon bo'ldi.

### 12.7. Multitasking va ko'p oqimli (potokli) tizimlar. VLIW, SMT

Multitasking (ing - ko'pmasalalilik) operatsion tizimning (yoki masalani bajaruvchi muhitning) bir nechta masalalarni parallel (yoki psevdoparallel) bajara olish xususiyatidir. Operatsion tizimning ko'pmasalalilik xususiyati asosan tarmoqlangan hisoblash tizimlarida ko'zga tashlanadi. Multitasking ning 2 xil turi mavjud:

1) jarayonli multitasking - bir vaqtda bajariladigan jarayonlarga asoslangan bo'lib, bunda bajariladigan dastur eng kichik element sifatida qaraladi. Ko'p tarqalgan hisoblanadi (masalan audio tinglab o'tirgan xolda matn yozish va shu kabilarda namoyon bo'ladi)

2) potokli multitasking - potokka asoslangan bo'lib, bunda potok eng kichik element sifatida qaraladi. Bitta dastur 2 ta va undan ortiq jarayonni bajarishi mumkin

Ko'p potoklilik multitaskingning maxsus formasidir.

### 12.8. Multitasking va uning xususiyatlari

Oddiy ko'pmasalali muhitlar resurslarni bo'lishib olish yo'nalishida ishlaydi, har bir masala (yoki jarayon) uchun alohida xotira qismi ajratiladi va masala aniq taqsimlangan vaqt intervallarida ishga tushiriladi. Boshqa multitasking tizimlarida resurslar taqsimotini dinamik amalga oshiradi, bunda masala xotirada ishlanishi boshlanadi va tizim strategiyasiga va ustunlik darajasiga qarab xotirani tark etishi ham mumkin. Bunday multitasking tizim quyidagi xususiyatlari bilan ko'zga tashlanadi:

- 1) Har bir masala o'zining ustunlik darajasiga ega, shunga mos ravishda protsessor vaqti va xotirani egallaydi
- 2) Tizim masalalar tartibini shunday tashkil qiladiki, barcha masalalar o'zlarining ustunlik darajasiga va tizimning strategiyasiga ko'ra resurslardan foydalana oladi
- 3) Tizim masalalarni faollashtirish, passivlashtirish, o'chirishni ta'minlovchi uzilishlar ishini tashkil qiladi.
- 4) Ajratilgan kvant vaqtida tizim yadrosi boshqa masalalarga resurslarni taqsimlash maqsadida masalani bajarilish holatidan tayyorlik holatiga o'tkazadi. Xotira kamlik qilgan holat sodir bo'lsa, bajarilmayotgan masalalar sahifalari diskka chiqarilib turishi mumkin (svopping), keyinchalik, tizim belgilagan vaqtdan keyin xotiraga qayta tiklanadi
- 5) Tizim masalaning adres muhitini boshqa masalalar egallab olishidan himoya qiladi
- 6) Tizim o'zining yadrosini boshqa masalalar egallab olishidan himoya qiladi
- 7) Tizim har bir alohida masalani (jarayonni) ishdan chiqishini va "osilib" qolganligini biladi va ularni to'xtatadi
- 8) Tizim bir nechta masala (jarayon) bir vaqtda bir joyga murojaat qilib, ularning umumiy "osilib" qolishini oldini olish uchun resurslar va qurilmalarga kirishni tartiblaydi
- 9) Tizimda har bir masala qachondir albatta faollashuvi ta'minlanadi
- 10) Tizim real vaqt so'rovlarini qayta ishlaydi
- 11) Tizim jarayonlar orasida aloqani ta'minlaydi

### 12.9. Multitasking tashkil qilishning murakkabliklari

Asosiy murakkablik ununig ishonchligini ta'minlashda paydo bo'ladi. Asosan, xotirani himoya qilish, uzilishlarni qayta ishlash, to'xtalib qolishlarni oldini olish va b.

Ishonchlikdan tashqari multitasking unumdor bo'lishi ham talab etiladi. Uni qo'llab-quvvatlashga talab qilinadigan resurslar jarajonlarni bajarilishiga ta'sir qilishi, ularning ishini sekinlashtirishi, xotirani cheklashi kabi amallarni bajarmasligi lozim.

### 12.10. Ko'pmasalali (multitasking) operatsion tizimlar tarixi

Multitasking OT (operatsion tizim) larni dastlabki ishlab chiqilishida birmuncha qiyinchiliklarga duch kelingan, xususan bunday OT larni o'rnatish va ishlatish jarayonida. Shu sababli uzoq vaqtgacha foydalanuvchilar bir masalali tizimdan voz kechishmagan. Bir qancha urinishlardan keyin multitasking tizimlari rivojlanib bordi, va hozirda barcha mashinalarda qo'llanilmoqda.

Dastlabki multitasking OT Multics deb nomlangan (1964-yil). Bundan tashqari multitasking IBM firmasining OS/360 OT da qo'llanilgan (1966-yil). 1969-yilda Multics OT asosida UNIX OT ishlab chiqildi. Bunda multitasking tizimi tartiblangan xolga keltirildi. PDP-11 kompyuterlarida multitaskingning RSX-11 varianti, vaqt taqsimlash tizimining TSX-PLUS varianti ishlatilgan.

Dunyodagi birinchi multimediali Amiga 1000 (1984-yil) shaxsiy kompyuteri multitaskingni to'liq qo'llab-quvvatlovchi mashina bo'lib, AmigaOS oper. Tizimida ishlagan. Amiga1000 ni ishlab chiqarish jarayonida texnik qurilmalar va dasturiy ta'minot (AmigaOS) parallel ishlab chiqarilishi natijasida multitaskingni rejalashtirish vaqti (1/50 sek) uzoq yillar davomida boshqa ishlab chiqaruvchilar uchun yetib bo'lmas ko'rsatkichga aylandi.

Multitaskingni shuningdek Microsoft firmasi Windows OT da qo'llagan. AmigaOS ning tajribasidan foydalanish OT larga yanada yuqori samaradorlik va ishonchlik berdi. Multitasking ishlatilish ko'rsatkichi bo'yicha hozir eng yirik OT lar bu – Windows va UNIX (Linux) hisoblanadi.

Multitasking nafaqat texnik jihatdan yoki OT larda, balki dasturlash tillarida ham ishlatiladi. Masalan, Modula-2 va Ada dasturlash tillarining xususiyati OT ga bog'liqligi yo'q va multitaskingni qo'llab-quvvatlaydi. Modula-2 dasturlash tili yordamida MS-DOS OT dasturi



uchun bir dasturning ko'p potoklari uchun multitasking ning turli tiplari ishlab chiqilgan. Buni amalga oshirish uchun dastur moduliga taymer uzilishlarini qayta ishlovchi masalalar rejalashtiruvchisi qo'shilgan. Bunday xususiyatli dasturlash tillarini, ba'zi hollarda, real vaqt dasturlash tillari deb ham aytiladi.

### 12.11. Pseudoparallel multitasking

Pseudoparallel multitaskingning 3 xil turi mavjud bo'lib, bular:

1. Oddiy ulash usuli. OT xotiraga bir vaqtda 2 ta yoki undan ko'p masalani yuklaydi, markaziy protsessor esa asosiy deb belgilangan masalani hal qiladi, qolgan masalalar esa, navbat bilan bir masalani bajarilishini to'xtatib turib, ikkinchisiga ulanish, ikkinchisini bajarilishini to'xtatib turib uchinchisiga ulanish va h.k tartibda bajariladi. Yorqin misol sifatida 1985-yilda MS DOS uchun ishlab chiqilgan **DESQview** dasturini ko'rsatish mumkin. **Ustunliklari:** multitaskingni qo'llamaydigan masala (jarayon) larni ham ishlatish mumkin. **Kamchiliklari:** nointeraktiv tizimlarda, inson faktorisiz ishlamaydi. Dasturlararo aloqa o'ta cheklangan.

2. Kooperativ multitasking (hamkorlikdagi multitasking). Joriy qayta ishlanayotgan masala (jarayon) boshqaruvni keyingi masala (jarayon) ga uzamaguncha keyingisi bajarilmaydi. Birinchi masalaning tegishli registrida masala bajarilganligi ishirasi paydo bo'lishi kerak. Shundan keyingina ikkinchi (uchinchi, to'rtinchi, va h.k) masalaga boshqaruv uzatiladi. Linux da *myuteks* obyektini band qilgan masala joriy masala deb hisoblanadi va bajariladi. Windowsda foydalanuvchi interfeysi tizimidan xabar kelishi bilan boshqaruv keyingi masalaga uzatiladi. Kooperativ multitaskingni ko'pmasalalikning keyingi bosqichi deyish mumkin. Oddiy ulash usulida boshqaruvni olgan masala (yoki jarayon) markaziy protsessor ni to'liq band qiladi, boshqa barcha masalarni bajarilishi to'xtab turadi. Kooperativ multitaskingda esa masalarning bajarilishi uchun markaziy protsessor ning qancha resursi kerak bo'lsa, shuncha ajratiladi. Barcha masalalae markaziy protsessor vaqtini teng taqsimlaydi, boshqaruvni davriy ravishda uzatilishi bilan jarayon davom etadi. **Yutuqlari:** barcha ma'lumotlarni kritik seksiyalar va myuteks lar deb nomlanuvchi obyektlar bilan himoyalash shart emasligi. Bu esa dasturlash, ayniqsa bir masalali muhitdan ko'p masalaliga o'tganda kod o'zgartirish talab etmaydi. **Kamchiliklari:** bitta masalada xatolik yuz bersa, boshqa masalalar ham bajarilishdan to'xtaydi, chunki, xatolikka uchragan masala protsessorni bo'shata

olmaydi. Bundan tashqari protsessor bir masalani bajarayotganda ikkinchi masala ma'lumot o'qish/yozishni bajarishini ta'minlashda murakkabliklar mavjudligi.

3. Ustunlikli multitasking (real vaqt rejimi). Bunda OT masalalarni bajarilish tartibini o'zi nazorat qiladi. Buning uchun keyingi masala kiritish/chiqarishni yakunlashi, texnik qurilmalarda qandaydir o'zgarishlar sodir bo'lishi, bir dasturdan ikkinchisiga axborot uzatilishi kabi o'zgarishlar sabab bo'ladi. Protsessor vaqtini jarayonlarni rejalashtirish ko'rsatkichi taqsimlaydi. Shuningdek, har bir masalaga foydalanuvchi yoki OT tomonidan ustunlik berilishi mumkin, bunday yondashuv protsessor vaqtdan yanada unumliroq foydalanishga olib keladi (masalan ko'p resurs talab qiladigan va qisqa vaqtda bajarilishi shart bo'lmagan masalaning ustunligini kamaytirib, uning bajarilishini sekinlashtirib, fon dasturlari yoki boshqa masalalarni ustunligini oshirish mumkin). Bunday multitasking foydalanuvchi chaqiruviga tezroq javob beradi. **Yutuqlari.** 1) OT yadrosida ko'pmasalali kiritish/chiqarish tashkil qilingan, bunda bir dastur axborotlarni kiritish/chiqarish bilan shug'ullanib turganda, boshqasi protsessorda bajarilishda davom etaveradi. 2) tizimning himoyalanganligi yuqori darajada ekanligi, bunda bir masalaning ishlamay qolishi boshqa jarayonlarga ta'sir o'tkazmaydi. 3) ko'protsessorli va ko'pyadroli tizimlarning imkoniyatlaridan to'laqonli foydalana olish. **Kamchiliklari.** Dastur kodi yozishda qat'iy tartibning bo'lishi shart ekanligi. Pseudoparallel multitasking quyidagi operatsion tizimlarda tashkil qilingan:

- VMS
- MenuetOS
- Linux
- Barcha UNIX-mos OT larda, jumladan Mac OS X, iOS; Symbian OS
- Windows ning barcha versiyalarida
- AmigaOS ning barcha versiyalarida

### Mavzu yuzasidan testlar

#### Qaysi biri multitasking xususiyati emas?

Tizim masalaning adres muhitini boshqa masalalar egallab olishidan himoya qila olmaydi.

Tizim o'zining yadrosini boshqa masalalar egallab olishidan himoya qiladi.

Tizim har bir alohida masalani (jarayonni) ishdan chiqishini va "osilib" qolganligini biladi va ularni to'xtatadi.

Tizim bir nechta masala (jarayon) bir vaqtda bir joyga murojaat qilib, ularning umumiy "osilib" qolishini oldini olish uchun resurslar va qurilmalarga kirishni tartiblaydi.

#### **Kooperativ multitasking ning alohida xususiyati qaysi?**

O'rnatish va ishlashda qulayliklar tug'diradi.

Qattiq xotirada kam joy egallaydi, jarayonlarni boshqarishda inson omili muhim.

Joriy qayta ishlanayotgan masala (jarayon) boshqaruvni keyingi masala (jarayon) ga uzamaguncha keyingisi bajarilmaydi.

Masalalarni yechish jarayonida fon ilovalari bajarilmasdan, markaziy protsessor ni reaksiyasini kutib turadi.

#### **Tashkil qilinish uslubiga ko'ra multitasking qaysi shakllarga bo'linadi?**

Ichki va tashqi

Xususiy va ommaviy

Jarayonli va potokli

Registrlil va keshli

#### **OT xotiraga bir vaqtda 2 ta yoki undan ko'p masalani hal qilayotganda, keyinga masalaga o'tishda qaysi texnologiya maqbul hisoblanadi?**

Oddiy ulash usuli

Ustunlikli multitasking

Foydalanuvchi aralashuvi

Kooperativ multitasking

#### **..... protsessor vaqtini jarayonlarni rejalashtirish ko'rsatkichi taqsimlaydi. Qaysi texnologiya haqida gap bormoqda?**

Oddiy ulash usuli

Foydalanuvchi aralashuvi

Ustunlikli multitasking

Kooperativ multitasking

#### **Asosan grafik muharrirlar ishlatiladigan kompyuterlarda Hyper-Threading texnologiyali markaziy protsessor tanlash nima uchun maqsadga muvofiq hisoblanadi?**

Chunki Hyper-Threading texnologiyasi quyi darajadagi ustunlikda ishlaydi.

Chunki Hyper-Threading texnologiyasi fon dasturlarini ishlatmaydi.

Chunki fon dasturlariga ikkinchi darajali ustunlik beriladi.

Chunki fon dasturlari bajarilishi to'xtab turishi mashina ishiga ta'sir ko'rsatmaydi.

#### **Hyper-Threading texnologiyasi markaziy protsessor unumdorligini qanday holatda pasaytirishi mumkin?**

Masaladagi ma'lumotlar bir-biriga bog'liq bo'lganda

Mashinaning xotira hajmi katta bo'lganda

Kesh xotira mavjud bo'lmaganda

Ma'lumotlar konveyerli qayta ishlanganda

#### **Hyper-Threading texnologiyasini qo'llaydigan va farqlamaydigan markaziy protsessor lar hajm jihatidan farqlanadimi?**

Ha, chunki qo'shimcha registrlar va uzilishlar kontrolleriga joy zarur

Ha, chunki kam sonli registrlar va xotira ishlatiladi

Yo'q, chunki har ikkala markaziy protsessor bir narsa

Yo'q, chunki Hyper-Threading texnologiyasi fizik qurilma

#### **Hyper-Threading texnologiyali protsessor**

Har bir mantiqiy protsessor uchun bitta registrlar to'plami va bitta uzilish kontrolleri to'plamiga ega bo'lishi shart emas.

Hajm jihatidan boshqa markaziy protsessor lardan keskin farq qiladi.

Bir vaqtda bir necha potoklarda bajarilayotgan jarayonlar haqidagi axborotni saqlaydi.

Unumdorlikni hamisha 2 barobar va undan ortiq ko'rsatkichga yaxshilaydi.

#### **Protsessorda bir necha masala bajarilayotganda masalaning bajarilish vaqti tushunchasi qanday o'lchanadi?**

Hamma masalalar hal qilingan tugallangan vaqt.

Eng murakkab masala bajarilishi uchun sarflanadigan vaqt.

Masalaning alohida qismlari bajariladigan vaqtlarning yig'indisi bilan o'lchanadi.

Barcha protsessorlarning unumdorliklari yig'indisi bilan o'lchanadi.

### 13. Kompyuterda ma'lumotlarning uzatilishi. Shina: tushunchasi, turlari, iyerarxiyasi. Shina hakami (apbtrp), shina protokollari

#### 13.1. Kiritish/chiqarishni shinali tashkil qilish

Shina (bus) deganda axborot almashinishga mo'ljallangan ko'p razryadli va qandaydir standartga asoslangan interfeysni tushunish lozim.

Shinaning xususiyatlari:

- Mantiqiy tashkil qilinganlik (razryadlar soni va mo'lajjanishi)
- Fizik tashkil qilinganlik (razryadlar soni, interfeysga ulanish usuli)
- Elektrik tashkil qilinganlik (ishlatadigan signal darajalari va b.)
- Protokolli tashkil qilinganlik (interfeys signallarining o'zaro bog'langanligi va b.)

Ko'prik (bridge) – ikkita har xil interfeysni moslashtiruvchi interfeys.

Shinali arxitektura hisoblash mashinalari ilk yaratilganidan buyon keng qo'llanilmoqda. Odatda shina interfeysiga ma'lumotlar liniyasi, adres liniyasi, boshqarish loiniyasi kiradi, shuningdek elektr tokiga ulanish, "yerga ulanish" kabi ulanishlar ham kiradi.

Sistema shinasini deganda mashinaning turli qurilmalarini o'zaro aloqaga kirishini ta'minlovchi shina tushuniladi. Bu yerda yana lokal sistema shinasini tushunchasi ham mavjud bo'lib, u shinaning mo'ljallanishi va ishlashi sistema doirasida lokal, lekin sistemaning ichida joylashgan, masalan yadroda. Sistema shinasiga qarama-qarshi ravishda tashqi shinalar mashinani tashqi qurilmalar bilan bog'lashda xizmat qiladi. Odatda, bu shinalar ham standartlashgan bo'ladi, chunki turli ishlab chiqaruvchilarning turli qurilmalari mashinaga ulanishida muammo sodir bo'lmasligi kerak.

Tarixan hisoblash mashinalaridagi birinchi tizim shinasini deb ISA (Industrial Standart Architecture – arxitekturaning sanoat (industrial) standart) tipidagi shinalar aytiladi. U 16 bitli ma'lumotlar shinasini bo'lib, 8,33 Hz chastotada ishlagan. Bu shinaning turli modifikatsiyalari EISA (Extended ISA), Microchannel, VLB (VESA Local Bus – VESA lokal shinasini, bunda VESA-Video Equipment Standarts Association – videoqurilmalar bo'yicha standart assotsiatsiyasi) va boshqalar hisoblanadi. ISA shinasini u yoki bu ko'rinishda boshqa tipdagi shinalarni ishlab chiqilishiga asos bo'ldi, masalan IDE (Integrated Device Electronics – qurilmaga integrallashgan elektronika) – diskli xotira

qurilmalarini ulash shinasini. IDE standarti mukammallashib EIDE ga aylandi, keyin esa ATA (AT attachment – AT ga qo'shilish), keyin ATAPI (ATA ulanish protokoli) bo'lib, turli diskli xotiralarni (masalan CR ROM, DVD ROM va b) ulashda ishlatildi. Zamonaviy diskli xotira qurilmalari ATA 33/66/100 standartida ishlashga mo'ljallangan. ATA33 standart (yoki Ultra DMA 33) axborot almashinish tezligini 33 megabit/s ga oshiradi, bunda takt impulsining ham oldingi, ham oxirgi frontlari ishlatiladi. ATA66 standartida (yoki UltraDMA66) shina chastotasi 16 MHz gacha oshiriladi, shinangn razryadliligi 32 gacha boradi.

Zamonaviy hisoblash mashinalari PCI (Peripheral Component Interconnect) shinasini bazasida qurilmoqda. PCI standartli shina nafaqat kompyuterlarda, balki sanoat mashinalari, katta EHM la va b.larda ham ishlatiladi.

PCI shinasining alohida xususiyatlari:

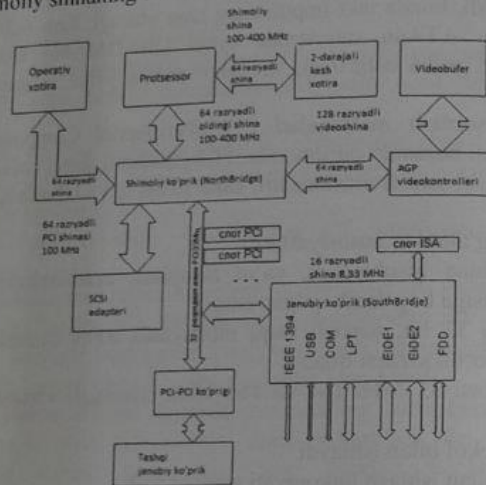
- Universal shina hisoblanadi, ya'ni hisoblash mashinalarining hech bir arxitekturasiga bog'lanib qolgan emas
- Razryadliligi 32 bit, uzatish vaqti mobaynida adres shinasini va ma'lumot shinasini bo'lib xizmat qiladi
- Takt chastotasi 33 MHz (64 va 166 MHz chastotali variantlari ham mavjud)
- Tezkor protokol bilan ishlaydi
- Ko'priklar bilan ishlash imkoniyati mavjud va b.

Ko'priklar bilan ishlay olishi PCI bazasida iyerarxik shinali arxitekturalarni qurish imkonini beradi. Zamonaviy bosqichda shina-ko'priqli arxitektura "Shimol-Janub" uslubida, shimoliy va janubiy ko'priklarni orasida tashkil qilinadi.

Shimoliy ko'priknin vazifasi markaziy protsessor, O3Y, videotizim va mashinaning yadro interfeysini PCI shinasini bilan bog'laydi. Shuningdek, shimoliy ko'prikl SCST (o'qilishi – skazi) bilan bog'lanishni ham bajaradi. Janubiy ko'prikl esa ISA interfeysi bo'yicha tashqi qurilmalarga ulanish kontrollerlari, EIDE, USB, FireWire, parallel va ketma-ket portlar kontrollerlari, klaviatura, PS/2 va boshqa portlarni bog'laydi. Shimoliy va janubiy ko'priklar o'zaro PCI shinasini bilan bog'langan va ularning o'rtasida tashqi PCI platalarni ulashga mo'ljallangan fizik interfeyslar joylashgan. Shimoliy ko'prikl PCI shinasini bilan ulanadigan plataidagi janubiy ko'prikl tashqari tashqi ko'priklar vam ulanishi mumkin. O'z navbatida ular ham o'zlarining

ko'priklariga ega bo'lishi va ulanishlar davom etaverishi mumkin. Bu esa PCI shinali arxitektura ochiq arxitektura ekanligini bildiradi.

Platani "Shimol" va "Janub" ga bo'lishdan olinadigan natija maxsuslashtirilgan va yuqori tezlikla yadroni standart sistema shinasidan va tashqi interfeyslardan ajratib olishdir. Pentium IV larda shimoliy shinaning chastotasi 400 MHz ga teng (13.1-rasm).



13.1-rasm. Platada shinalarning joylashuvi

**Kompyuter shinasini** — kompyuter quyi tizimi, kompyuter funksional birliklari o'rtasida ma'lumotlarni uzatish uchun xizmat qiladi. Shinalarni odatda o'tkazgichlarda bir necha qurilmalar ulanishi nuqtasi-to-nazaridan, mexanik (jismoniy) elektr va mantiqiy (nazorat)ga ajratish mumkin. Har bir shina, bir aloqa, karta va kabel uchun ulagichga ulanadi, uning majmuini belgilaydi. Kompyuter shinalari ilk kompyuterlarda (kabellarining to'plamlar — signal va kuch-quvvat, Ixcham va birga bog'langan texnik qulaylik uchun) bir necha ulanishlar bilan parallel elektr tokini o'tkazish qurilmasi edi. Zamonaviy kompyuter tizimlarida, bir muddat parallel shinalar kompyuter bir xil mantiq funksiyalarini ta'minlash va har qanday mexanizmlari uchun ishlatiladi. Zamonaviy kompyuterda shina parallel yoki ketma-ket ulanish sifatida ishlatiladi va parallel (Eng. Multidrop) va zanjir (Eng. Daisy zanjir) topologiyasiga bo'lish mumkin. USB va boshqa shinalarda

konsentratorlar (xablardan) foydalanish mumkin. Uzatish signallari uchun tezyurar shinalari ayrim turlarida (Fibrin Channel, InfiniBand, tezyurar Ethernet, SDH) elektr ulanishlardan foydalanilmaydi, optik ulanishlar ishlatiladi. Multiplexers, demultiplexers, nazorat shina signali orqali (Multipleksorler, demultiplexers, tamponlar, registrlar, registrlar, haydovchisi) amalga oshiriladi va operatsion tizimining yadrosi tomonidan maxsus drayver kerak bo'ladi. Shinalar tipologiyasi

### 13.2. Parallel shinalar

Proprietary ASUS Media shina, Socket 7 bilan ba'zi ASUS diskdan ishlatiladigan va ulagichi shina PCI line joylashtirilgan, muayyan Iso shina ulagichi vakili. • Tizimlar uchun CAMAC • Iso yoki EISA • Standard Arxitektura yoki Iso • Kam Pin Count yoki LPC • mikrokanal yoki MCA • MBusSanoat tizimlar uchun • Multibus • NuBus yoki IEEE 1196Intel 80486 • Optik mahalliy shina, • Periferik Component Interconnect yoki PCI. • S-100 yoki IEEE 696 shina, Altair va shunga o'xshash mikrokompyuterda ishlatiladi • SBus yoki 1496 IEEE • 80486, protsessor uchun asosan diskda ishlatiladi va terminallari bilan to'g'ridan-to'g'ri bog'liq.

### 13.3. VLB

VMEbus, VERSAmodule Eurocard shina8- va 16-bit mikrosxema tizimlari uchun • STD shina • Unibus • Q-shinaSerial • 1-Wire• HyperTransport • I<sup>2</sup>C• PCI Express yoki PCIe • Serial Periferik Interface shina yoki SPI shina • USB, Universal Serial Bus • FireWire, i.Link, IEEE 1394 ko'proq tashqi shina sifatida ishlatiladi • Direct Media interfeysi (DMI) • Intel QuickPath Interconnect yoki faqat QuickPath (QPI) • SATA / SAS tashqi kompyuter shinalar misollar • Advanced Technology Attachment yoki (shuningdek PATA, edi, Eyd ATAPI sifatida tanilgan) ATA — disk va lenta ulash uchun. • SATA, Serial ATA • USB, Universal Serial Bus, tashqi qurilmalar uchun ishlatiladi, • IEEE-488, GPIB (General-maqsadi Asbobsozlik shina), HPIB, (Hewlett-Packard • Futurebus • InfiniBand • QuickRing

### 13.4. PCI Express

PCI Express, yoki PCI, yoki (shuningdek, 3-Generation I / O uchun 3GIO sifatida tanilgan; PCI-X va PXI bilan adashtirmaslik kerak) PCI-Edan foydalanuvchi ketma-ket signal uzatishga asoslangan yuqori chastotali shina. PCI Express standartini rivojlantirish InfiniBand tomonidan radqilingandan so'ng Intel tomonidan boshlangan edi.

Rasman, birinchi tayanch PCI Express spetsifikatsiyasi iyul 2002 yilda paydo bo'ldi. PCI Maxsus Guruhi PCI standart imkoniyatini rivojlantirish uchun tashkil qilinadi.

### 13.5. Razyomlari

ExpressCard — omili PCMCIA. Shina olingan x1 PCIe va USB 2.0 ExpressCard uyasi, ExpressCardni ulab qo'llab-quvvatlaydi. • AdvancedTCA — telekommunikatsiya uskunalari uchun. • mobil PCI Express moduli (MXM) -, NVIDIA tomonidan Noutbuklar uchun mo'ljallangan. Bu grafik tezlatkichlardan ulash uchun ishlatiladi. • Simli PCI Express xususiyatlari imkon kompyuterlar bir muhim masofada joylashgan, periferik qurilmalar yaratish qiladi metr o'nlab bitta ulanish, uzunligini olib imkonini beradi. • StackPC — kompyuter tizimlarini qurish uchun spetsifikatsiyasi. Ushbu belgilari kengaytirish StackPC, FPE va ularning o'zaro tartibni ifodalaydi.

### Mavzu yuzasidan test savollari

#### PCI shinasining xususiyatlariga kirmaydigan xususiyat qaysi?

Universal shina hisoblanadi, ya'ni hisoblash mashinalarining hech bir arxitekturasiga bog'lanib qolgan emas.

Razryadlilik 32 bit, uzatish vaqti mobaynida adres shinasini va ma'lumot shinasini bo'lib xizmat qiladi.

Axborot almashinishga mo'lallanmagan, faqat komandalari o'tkazishni tashkil qiladi.

Takt chastotasi 33 MHz (64 va 166 MHz chastotali bo'lishi mumkin).

#### Platani "Shimol" va "Janub" ga bo'lishdan nima ko'zda tutiladi?

Maxsuslashtirilgan va yuqori tezlikla yadroni standart sistema shinasidan va tashqi interfeyslardan ajratib olish.

Qurilmalararo interfeysi yaxshilash.

Platada shinalarning joylashish tartibini o'rnatish.

Tashqi qurilmalarni ulanishi va ular bilan muloqot interfeysini yaratish.

#### Shimoliy ko'prik asosan qanday turdagi qurilmalar bilan ishlashga mo'ljallangan?

Tashqi qurilmalar bilan.

USB porti bilan.

Tezkor qurilmalar bilan.

SCSI porti bilan.

Shimoliy va janubiy ko'priklar o'zaro qanday bog'lanadi?

PCI shinasini bilan bog'langan.

ATA shinasini bilan bog'langan.

Plata orqali bog'langan.

Bog'lanmagan.

Mashinaning qurilmalarini o'zaro bog'lash uchun dastlab qaysi tipdagi shina ishlatilgan?

ISA

ATAPI

VLB

SCSI

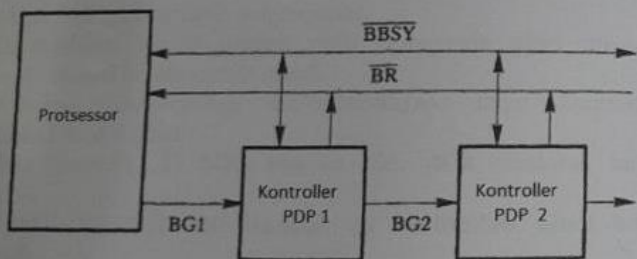
#### 14. Shina hakami (apбyтp), shina protokollari

Qaysi qurilmaga shinadan axborot uzatish huquqi berilgan bo'lsa, bu qurilma *shinaning egasi* deyiladi (bus master). Bu qurilma axborotlarni uzatishni yakunlaganidan keyin, shinaning egaligidan tushadi va boshqa qurilma axborot uzatish uchun shinaning egasiga aylanadi. Shinaning keyingi egasini aniqlash jarayoni maxsus protsedura yordamida aniqlanadi va bu protsedura *shina hakami* deyiladi. Tanlashda turli qurilmalarning talablari o'rganiladi va bunda ustunliklar darajalari qo'llaniladi.

Shinadagi sodir bo'ladigan nizolarni hal qilishning 2 usuli bor: markazlashgan hakamlik va taqsimlangan hakamlik. Nizolarni markazlashgan hakamlik bilan hal qilish yondashuvida shina hakami asosiy hal qiluvchi bo'lsa, taqsimlangan hakamlikda barcha qurilmalar ishtirokida hal qilinadi.

##### 14.1. Markazlashgan hakamlik.

Shinaga ulangan har qanday qurilma shina hakami bo'la oladi (14.1-rasm).

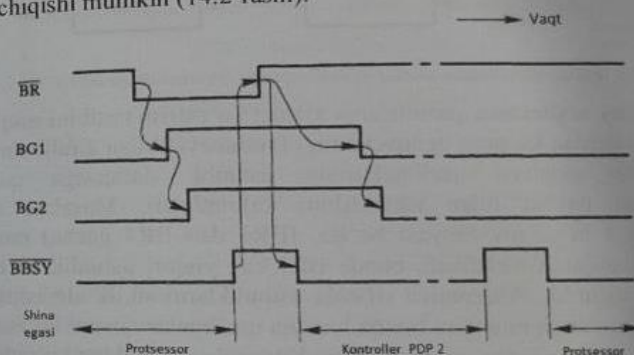


14.1-rasm. Hakamlik sxemasi markaziy protsessor tarkibida bo'lgan kompyuter arxitekturasi

14.1-rasmda hakamlik sxemasi markaziy protsessor tarkibida bo'lgan kompyuter arxitekturasi keltirilgan. Bunday arxitekturada markaziy protsessor odatda shina egasi bo'lib xizmat qiladi, toki egalikni xotiraga to'g'ridan-to'g'ri murojaat (XTM-П/III) qiladigan boshqa qurilmaga uzatmaguncha. XTM qurilmaning kontrolleriga shina kerak bo'lsa, u shina so'rovi liniyasini (BR-Bus Request) faollashtiradi. Bu liniya uzilish so'rovi liniyasiga o'xshash.

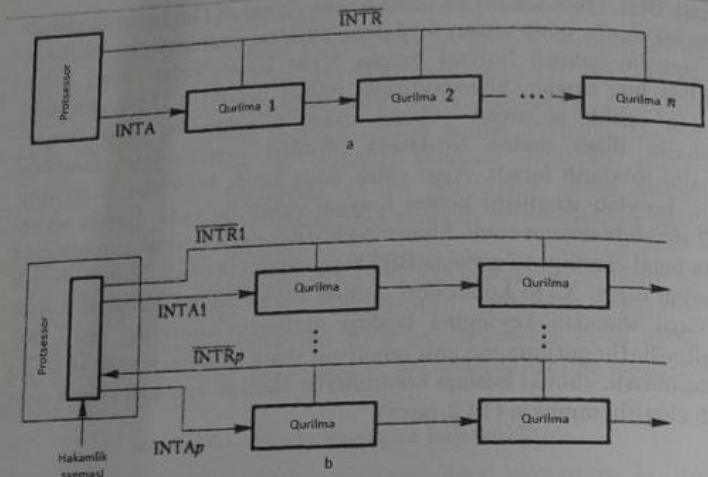
Shinaning so'rov liniyasidagi signal ushbu liniyaga ulangan barcha qurilmalarning signallarining yig'indisidan iborat bo'ladi (YOKI amali). Shina so'rovi liniyasi faollashganda markaziy protsessor shinani berish

signali BG1 (Bus Grant) ni uzatadi. Bu signal XTM kontrollerlariga shina bo'shshasi bilan undan foydalana olishlari mumkinligini bildiradi. Bu signalni uzatish liniyasi barcha XTM kontrollerlari bilan zanjir shaklida bog'langan. Shunday qilib, (misol uchun) XTM-1 qurilmasi shinaga egalik so'rovini yuborar ekan, u zanjir bo'yicha shinani bo'shshasi bilan undan foydalana olishlari mumkinligini haqidagi signalni to'xtatib turadi. Agar shina unga kerak bo'lmasa, u signalni zanjir bo'ylab uzatilishi uchun liniyani ochib qo'yadi. Bunda signal BG2 shaklida davom etadi. Shinaning hozirgi egasi boshqa qurilmalarga shina band ekanligi to'g'risida BBSY signalini yuboradi. Shinani berish signalini olgan XTM kontrolleri qachon BBSY signalini kelishini kutadi va faqat shundan keyingina boshqa qurilmani shinaning egasi deb belgilaydi. Bu qurilma esa shu zahotiy oq shina bandligi signali (BBSY) faollashtiradi, chunki boshqa kontrollerlar shinaga egalikka chiqsa nizo kelib chiqishi mumkin (14.2-rasm).



14.2-rasm. Shinani boshqarishni bir qurilmadan boshqasiga o'takizshda signallar ketma-ketligi

14.2-rasmda yuqorida yozilgan jarayonning grafik shakli keltirilgan. Aniqrog'i, PDP-2 kontrolleri shinaga egalikni so'raganidan boshlab, shinani bo'shatgan vaqtigacha jarayon ko'rsatilgan. Kontroller shina egasi bo'lib turgan vaqtda u blokli rejimda ishlayaptimi yoki shinani egallash rejimidami, farqi yo'q, ma'lumot uzatish bo'yicha bir yoki bir nechta amallarni bajara oladi. Kontroller shinani bo'shatishi bilan protsessor yana shina egaligini oladi. Slaydda hakamlik jarayonida qatnashayotgan signallar orasidagi sabab-oqibat aloqasi keltirilgan. 4.20-rasmda bir liniya so'rov va bir liniya shinani berish ko'rsatilgan. Umuman, kompyuterlarda bunday juftliklar bir nechta bo'lishi mumkin, xuddi uzilish liniya so'rovlari kabi bo'ladi (14.3-rasm).

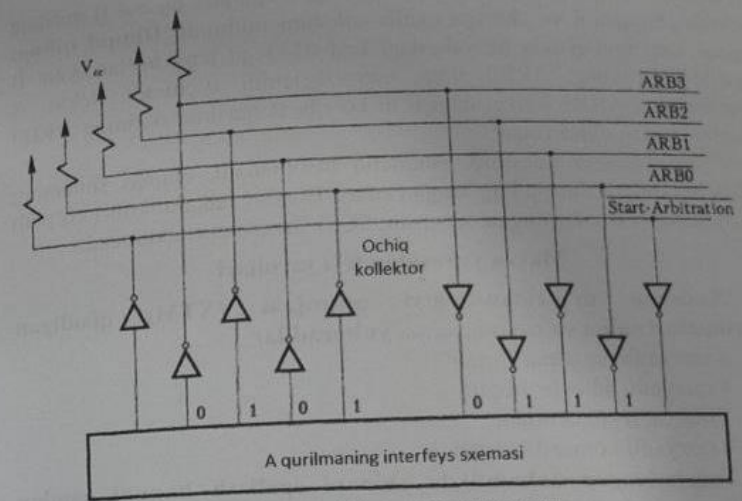


14.3-rasm. Ustunliklarning ustunliklar sxemasi: (a) giriyanda zanjiri; (b) ustunliklar guruhi

Bunday arxitektura qurilmalarga xizmat ko'rsatish tartibini aniqlash nuqtai nazaridan ko'proq egiluvchanligi (гибкость) bilan ajralib turadi. Hakamlik sxemasi qurilmalarning ustunlik darajasiga qarab, so'rovlarni navbat bilan ishlanishini kafolatlaydi. Masalan, agar shinaning 4 ta so'rov liniyasi bo'lsa, (BR1 dan BR4 gacha) ramziy ustunlik darajalari ishlatiladi, bunda BR1 eng yuqori ustunlik va BR4 eng quyi ustunlik. Alternativa sifatida ustunliklarni siklik almashtirish sxemasini qo'llash mumkin, bunda hamma qurilmalar xizmat ko'rsatish uchun teng imkoniyatga ega bo'ladilar. Yuqoridagi misoldan ko'radigan bo'lsak, BR1 liniyasiga xizmat ko'rsatilgandan keyin xizmat ko'rsatish tartibi 2-3-4-1 bo'ladi.

### 14.2. Taqsimlangan hakamlik

Taqsimlangan hakamlikda shinani ishlatish uchun navbatda turgan barcha qurilmalar hakamlikda ishtirok etadilar, chunki bu yerda boshqaruvchi markaziy hakam bo'lmaydi. Taqsimlangan hakamlik quyidagi rasmda keltirilgan (14.4-rasm):



14.4-rasm. Taqsimlangan hakamlik sxemasi

Shina bilan bog'langan har bir qurilmaga 4 razryadli identifikatsiya raqami beriladi. Bir yoki bir necha qurilmalar shina so'raganda, ular hakamlikni boshlaganini bildiruvchi *Start-Arbitration* signalini faollashtiradilar va o'zlarining 4 razryadli identifikatsion raqamlarini liniyaning *ARB-0* dan *ARB-3*gacha ochiq kollektorlarga joylaydi. Shinani egallash huquqi hamma qurilmalarning so'rovlarini qayta ishlash natijasida aniqlanadi yoki, boshqacha aytganda, eng katta identifikatsion raqamli so'rovga beriladi. Bu sxemada *ochiq kollektorli takrorlagichlar* ishlatiladi. Bunda, agar bir takrorlagich chiqishida 1, ikkinchisida 0 bo'lsa va ular bir shinaga ulangan bo'lsa, umumiy chiqishda 0 hosil bo'ladi (mantiqiy VA amali).

Tasavvur qilaylik, A va B qurilmalari bir vaqtda shinani so'radi. (rasm). A qurilma 0101 qiymat, B qurilma 0110 qiymat yubordi. Har ikkalasi 0111 kodini oladi. Shundan keyin har bir qurilma bu kodni o'zining identifikatsion nomeri bilan taqqoslaydi. Taqqoslash kichik razryaddan boshlanadi, agar biror razryadda farq mavjud bo'lsa, shu razryaddagi va shu razryaddan quyi barcha razryadlarning takrorlagichlarini ajratadi. Buning uchun shu takrorlagichlarning chiqishlariga 0 ni yozib qo'yadi. Bizning misolda A qurilmasi ARB1 liniyasida farqni ko'radi va ARB1, ARB0 liniyalarini takrorlagichlarini

o'chiradi. Natijada liniyadagi kod 0110 ga o'zgaradi, bu esa B qurilma g'olib chiqqanini va shinaga egalik qilishini bildiradi. **Diqqat qiling:** qisqa vaqt mobaynida liniyalardagi kod 0111 ga teng bo'lar ekan, B qurilma o'zining ARB0 dagi takrorlagichini o'chiradi, lekin A qurilmaning ARB1 liniyasidagi 0 ni ko'rib, B qurilma o'zining ARB0 dagi takrorlagichini yoqadi.

Taqsimlangan hakamlik ishonchli hisoblanadi, chunki shinaning ishlashi bitta qurilmaga bog'langan emas. Bunday hakamlik turi ko'plab sxemalarda tashkil qilingan, xususan, SCSI shinasi arxitekturasida.

#### Mavzu yuzasidan test savollari

Xotiraga to'g'ridan-to'g'ri murojaat (XTM) qiladigan qurilmalari shina so'rovini ..... yuboradilar

- 4 razryadli registrlar orqali
- 4 razryadli adreslar orqali
- 4 razryadli shina orqali
- 4 razryadli komanda orqali

Taqsimlangan hakamlikda shinani egallash huquqi qanday qurilmaga birinchi beriladi?

- Eng ko'p marta so'rov yuborgan qurilmaga
- Eng katta identifikatsion raqamli so'rovli qurilmaga
- Mantiqiy zanjirda birinchi turgan qurilmaga
- markaziy protsessor ga yaqin joylashgan qurilmaga

**Shina hakami nima?**

- Shinani qurilmalar orasida teng taqsimlovchi qurilma
- Mantiqiy zanjir elementi

Shinaning keyingi egasini aniqlash protsedura

Shinadagi nizolarni bartaraf etish vositasi

.....qurilmalarning ustunlik darajasiga qarab, so'rovlarni navbat bilan ishlanishini kafolatlaydi

- Identifikatsion raqam
- Mantiqiy zanjir
- Boshqaruv shinasi
- Hakamlilik sxemasi

**BG1 (Bus Grant) signali qaysi amalni bajaradi?**

- Shinani berish signali
- Shinani bandlik signali
- Shinani so'rov liniyasi
- Shinani ulanish porti

**PCI shinasining xususiyatlariga kirmaydigan xususiyat qaysi?**  
Universal shina hisoblanadi, ya'ni hisoblash mashinalarining hech bir arxitekturasiga bog'lanib qolgan emas

Razryadliliigi 32 bit, uzatish vaqti mobaynida adres shinasi va ma'lumot shinasi bo'lib xizmat qiladi

Axborot almashinishga mo'lallanmagan, faqat komandalari o'tkazishni tashkil qiladi

Takt chastotasi 33 MHz (64 va 166 MHz chastotali bo'lishi mumkin)

**Platani "Shimol" va "Janub" ga bo'lishdan qanday maqsad ko'zda tutiladi?**

Maxsuslashtirilgan va yuqori tezlikla yadroni standart sistema shinasi va tashqi interfeyslardan ajratib olish

Qurilmalararo interfeysi yaxshilash

Platada shinalarning joylashish tartibini o'rnatish

Tashqi qurilmalarni ulanishi va ular bilan muloqot interfeysini yaratish

**Shimoliy ko'prik asosan qanday turdagi qurilmalar bilan ishlashga mo'ljallangan?**

- Tashqi qurilmalar bilan
- USB porti bilan
- Tezkor qurilmalar bilan
- SCSI porti bilan

**Shimoliy va janubiy ko'priklar o'zaro qanday bog'lanadi?**

- PCI shinasi bilan bog'langan
- ATA shinasi bilan bog'langan
- Plata orqali bog'langan
- Bog'lanmagan

**Mashinaning qurilmalarini o'zaro bog'lash uchun dastlab qaysi tipdagi shina ishlatilgan?**

- ISA
- ATAPI
- VLB
- SCSI



**15. Kiritish-chiqarish tizimlarining manzil maydonlari. Kiritish-chiqarish modullari. Kiritish-chiqarishni boshqarish metodlari. Kiritish-chiqarish kanallari va portlari, ularning xususiyatlari**

Kompyuterga kiritish/chiqarish qurilmalarini ulashning eng oddiy usuli shina orqali ulanish bo'lib, u rasmda ko'rsatilgan. Shinaga ulangan barcha qurilmalar o'zaro axborot almasha oladilar. Odatda shinalar uchta to'plamli liniyadan iborat bo'ladi: adres, ma'lumot va komanda uzatish to'plamlari. Har bir kiritish/chiqarish qurilmasiga alohida adres to'plami ajratiladi. markaziy protsessor adres liniyasiga aniq adresni joylaganda, shu adresli qurilma boshqaruv liniyasidan kelayotgan komandaga javob beradi (uni bajaradi). Kiritish/chiqarish qurilmalari va xotira bitta adres muhitidan foydalangan xolda kiritish/chiqarishni tashkil qilinishi *kiritish/chiqarishni xotirada akslanishi* deyiladi.

Kiritish/chiqarishni xotirada akslanishini ishlatganda xotiraga murojaat qiluvchi har qanday mashina komandasi kiritish/chiqarish qurilmalari bilan ma'lumot almashishda ishlatilishi mumkin. Masalan, DATAIN – kirish buferi adresi bo'lsa,

*Move DATAIN,R0*

komandasi DATAIN dan ma'lumotlarni o'qiydi va R0 (R-nol) registrga joylashtiradi.

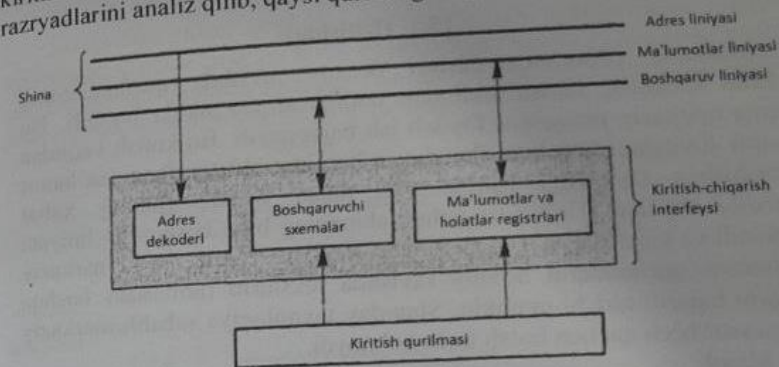
Xuddi shuningdek,

*Move R0,DATAOUT*

komandasi R0 reghistrdagi ma'lumotlarni DATAOUT adresiga (bu displey, printer va/yoki boshqa qurilma bo'lishi mumkin) uzatadi.

Kiritish/chiqarishni xotirada akslanish texnologiyasi hamma kompyuter tizimlarida ishlatiladi. Ba'zi markaziy protsessor lar kiritish/chiqarish ni bajarish uchun maxsus *In* va *Out* komandalrini ishlatadilar. Masalan *Intel* protsessorlari maxsus kiritish/chiqarish komandalari va kiritish/chiqarish qurilmalari uchun alohida 16 razryadli adres maydoniga ega. Bu protsessor asosida kompyuter tizimini ishlab chiqishda qurilmalarni: 1) umumiy kiritish/chiqarish adres maydonidan foydalanadigan yoki 2) har birini adres maydonining bir qismi sifatida tizimga ulashi mumkin. Bunday, 2)-yondashuv dasturiy ta'minotni soddalashtirganligi sababli urfda keng qo'llaniladi. Bu yondashuvning ustunliklaridan biri qurilmalar kam sonli adres liniyalarini band qiladilar. Kiritish/chiqarishning alohida adres muhitiga egaligi, kiritish/chiqarish adres liniyalari xotiraning adres liniyalaridan alohidaligini bildirmaydi. markaziy protsessor dan uzatilgan signal

kiritish/chiqarish tizimiga tegishli ekanligini shinadan uzatiluvchi maxsus signal aniqlaydi. Bu signalni olgan xotira uni inkor qiladi. kiritish/chiqarish qurilmalari esa shinadan kelgan adresning kichik razryadlarini analiz qilib, qaysi qurilmaga tegishliligini aniqlaydi.



15.1-rasm. Kiritish qurilmalari uchun kiritish-chiqarish interfeysi

Qurilmani shinaga ulash uchun talab qilinadigan elementlar slaydda kelritilgan. Adres liniyasida paydo bo'lgan qurilma adresini qurilma dekoder (deshifrotor) yordamida aniqlaydi. Qurilma va markaziy protsessor almashadigan ma'lumotlar ma'lumotlar registrida joylashadi. Holat registri esa kiritish/chiqarish qurilmasining ishlash holati haqidagi axborotni saqlaydi. Holat reishtri va ma'lumotlar registri ma'lumotlar shinasini bilan bog'lanadi va ularga adres beriladi. Adres deshifrotori, holat va ma'lumot registrlari, boshqaruvchi sxemalar – hammasi *qurilmaning ulanish sxemasi* yoki *interfeysini* tashkil qiladi.

Kiritish/chiqarish qurilmalarining ish tezligi markaziy protsessor ish tezligidan keskin farq qiladi. Misol uchun, foydalanuvchi klaviaturadan kiritadigan ketma-ket ikkita simvolning orasida o'tgan vaqt mobaynida markaziy protsessor millionlab amallarni bajaradi. Klaviaturadan simvolni o'qiydigan komanda faqatgina, komanda klaviatura interfeysining kirish buferida bo'lgandagina bajariladi. Klaviaturadan ma'lumot kiritganda registrning bir razryadida *holat bayrog'i* – *SIN* qo'shiladi. Klaviaturadan signal kiritilganda uning qiymati 1 teng, kiritilgan simvol markaziy protsessor tomonidan o'qilganda 0 ga teng. Shu tartibda, ya'ni *SIN* bayrog'ining holatini tekshirib, markaziy protsessor ma'lumotlarni o'qish amalini to'g'riligini ta'minlaydi. Buning uchun dasturiy sikl tashkil qilinadi. Sikl holat

registrini SIN bayrog'i holatini tekshirib turadi. Bayroq 1 ga teng bo'lganini sezgan dastur kiruvchi ma'lumotlar registridan uni o'qiydi. Xuddi shu tartibda chiqarish amali ham bajariladi, faqat bunda SOUT holat bayroqchasi ishlatiladi.

### 15.1. Uzilishlar

Markaziy protsessor dasturi doimiy ravishda qurilmalarning holatini tekshirib, kutish sikli deb nomlanadigan siklda bo'ladi. Bu vaqtda markaziy protsessor foydali ish bajarmaydi. Bu kutish vaqtdan unumli foydalanish uchun shunday qilish kerakki, qurilma ma'lumot almashishga tayyorligi haqida o'zi xabar bersin. Bunday xabar uzilishlar deyiladi. Uzilishlar uchun shinaning bitta boshqaruv liniyasi ajratiladi va u uzilish so'rovlari liniyasi deyiladi. Bunda, ya'ni markaziy protsessor qurilmalarni doimiy ravishda tekshirib turmasdan boshqa ishlarni bajarib turishi mumkin. Shunday texnologiya sababli markaziy protsessor hech qachon bo'sh turib qolmaydi.

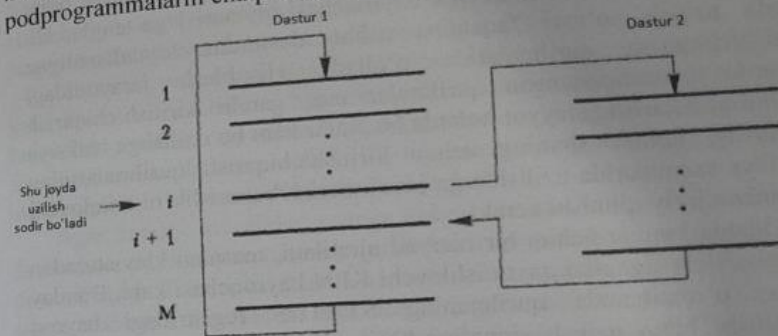
Misol.

Aytaylik, kompyuter qandaydir hisoblashlarni bajaradi va natijalarni printeriga chiqaradi, keyin boshqa hisoblashlarni bajaradi, ularni printeriga chiqaradi va h.k. bu ishni bajaruvchi dastur 2 ta COmarkaziy protsessorUTE va PRINT podprogrammalaridan iborat deb tasavvur qilamiz. COmarkaziy protsessorUTE  $n$  ta satrni PRINT programmasi printerda chop etishi uchun uzatadi.

Masalani hal qilishda COmarkaziy protsessorUTE dasturi bir necha marta bajariladi, keyin PRINT bir marta bajariladi. Printer esa bir urinishda bir qator matnni qabul qiladi. Shu sababdan PRINT podprogrammasi bir satrni printeriga uzatadi, kutadi, keyingi satrni uzatadi, kutadi va h.k. bunda markaziy protsessor ham printerning tayyorligini har gal tekshirish uchun ko'p resurslarini sarflaydi. Umumiy unumdorlikni oshirish uchun COmarkaziy protsessorUTE va PRINT podprogrammalarini navbatli ishlatish mumkin, ya'ni dastlab COmarkaziy protsessorUTE ni bajarish,  $n$  ta satrni olish, keyin PRINT ni bajarish. Printer bir satrni chop qilguncha, PRINT podprogrammasi faqat kutish bilan band bo'ladi. Ana shu kutish vaqtini COmarkaziy protsessorUTE podprogrammasi bajarilishiga uzatish mumkin. Printer bir satrni chiqarib, ikkinchisini chiqarishga tayyor bo'lganda, bu haqda markaziy protsessor ni ogohlantiradi, markaziy protsessor COmarkaziy protsessorUTE ning bajarilishini to'xtatadi, boshqaruvni PRINT ga beradi. PRINT yana bir satrni printeriga uzatadi va yana ishini to'xtatadi.

Bajarilishi to'xtatilgan COmarkaziy protsessorUTE dasturi kelgan joyidan bajarilishda davom etadi. Va h.k. Pechatdan ko'ra hisoblashga ko'p vaqt ketishini hisobga oladigan bo'lsak, markaziy protsessor hamma vaqt hisoblashlar bilan band bo'lishini ko'ramiz.

Yuqoridagi misol uzilishlar qanday tashkil qilinganligini yaqqol ko'rsatadi. Uzilish talabiga ko'ra bajariladigan dastur uzilishlarni qayta ishlash dasturi deyiladi. Bizning misolda bu PRINT dasturi. Uzilishlar podprogrammalarini chaqirishga o'xshaydi.



15.2-rasm. Uzilish yordamida boshqaruvni uzatish

Faraz qilamiz, uzilish talabi  $i$  komandasini bajarish vaqtida kelgan bo'lsin (15.2-rasm). Markaziy protsessor komandani bajarilishini to'xtatadi, komanda hisoblagichida uzilishlarni qayta ishlash dasturining birinchi komandasi adresini yuklaydi. Aytaylik, bu adres markaziy protsessor da yozilgan bo'lsin. Uzilishlarni qayta ishlash komandasini bajarib bo'lgandan keyin markaziy protsessor ishini  $i+1$  komandadan boshlashi kerak. Buning uchun uzilishlarni qayta ishlash dasturini chaqirishdan oldin registrlarning qiymatlari xotirada saqlangan bo'lishi kerak. Uzilishlarni qayta ishlash dasturining oxiridagi uzilishdan qaytish komandasiga shu saqlangan registr yozilgan bo'ladi va bajarilishi  $i+1$  dan boshlanadi. Ba'zi kompyuterlarda qaytish adresi protsessor stekida joylashadi. Uni boshqa joyda, masalan o'zi uchun alohida registr qilib shunda saqlash ham mumkin.

Uzilishni qayta ishlayotgan markaziy protsessor qurilmaga uning so'rovi o'qilganligi haqida xabar beradi, qurilma esa shinadan maxsus signal uzatish orqali uzilishga so'rov berishni to'xtatadi. Boshqa variantda buni tashkil qilish uchun markaziy protsessor va kiritish/chiqarish interfeysi orasida ma'lumot uzatish bajariladi.

Uzilishlarni qayta ishlash dasturida holat registrining yoki ma'lumotlar registrining qiymati o'zgartiriladi va qurilma uzilish so'rovi qabul qilinganligini biladi.

### 15.2. Kiritish-chiqarish kanallari va portlari, ularning xususiyatlari

**Qurilmalar so'rovlarini boshqarish.** Ma'lumki, kiritish/chiqarish qurilmasi interfeysi qurilma ma'lumot uzatishga tayyor bo'lishi bilan uzilishga so'rov yuboradi va SIN bayroqchasi qiymati 1 ga tenglanadi. Bunda uzilish so'rovi faqatgina ushbu dasturdan foydalanadigan kiritish/chiqarish qurilmalarida o'qiladi. Hisoblash jarayonidagi dasturda qatnashmayotgan qurilmalar esa, garchi kiritish/chiqarish amallarini bajarishga tayyor holatda bo'lsalar ham bu uzilishga reaksiya bermasligi kerak. Shuning uchun kiritish/chiqarish qurilmalarining interfeys sxemalarida uzilishlarga javob berish/bermaslik ni aniqlovchi mexanizm joriy qilinishi kerak.

Odatda buning uchun bir razryad ajratiladi, masalan klaviaturadan kelgan uzilish signalini qayta ishlovchi KEN bayroqchasi kabi. Bunday bayroq o'rnatilganda qurilmaning STATUS registridagi bayroq o'rnatilishi bilan uzilish signalini hosil qiladi. Shu vaqtda interfeys sxemasi KIRQ yoki DIRQ razryadini o'rnatib, ushbu qurilma uzilish so'rovini yuborganligini belgilab qo'yadi.

Agar uzilishga ruxsat bayrog'i 0 ga teng bo'lsa, interfeys sxemasi holat bayrog'ining qanday bo'lishidan qat'iy nazar uzilishga so'rov yubormaydi.

Uzilishlarni boshqarishning ikki mexanizmi mavjud. Qurilma tomonidan unga uzilish signalini hosil qilish kerakmi-yo'qligi uzilish razryadiga ruxsat berish xolatiga bog'liq bo'ladi. markaziy protsessor tomonidan uzilish signalini qabul qilish kerakmi-yo'qligi PS registridagi uzilishga ruxsat berish razryadi yoki ustunliklar tizimiga bog'liq bo'ladi.

### 15.3. Istisnolar

Uzilish – bir dasturning ishini to'xtatib turish va ikkinchisini ishini boshlashdir. Yuqorida ko'rilgan uzilishlar barchasi qurilmalarning so'rovi bo'yicha qilingan uzilishlar edi. Uzilishlar mexanizmi esa, ko'plab boshqa xolatlarda ham namoyon bo'ladi. Odatda uzilishga olib keluvchi har qanday holatni *istisno* deyiladi. Demak,

kiritish/chiqarishdagi uzilishlarni ham istisno deyish mumkin. Istisnoning turli ko'rinishlari.

### 15.4. Xatolardan keyingi qayta tiklanish

Kompyuterning texnik qurilmalarining to'g'ri hamkorlikda ishlashi turli xil texnologiyalar yordamida tashkil qilinadi. Masalan, ko'plab kompyuterlarning asosiy xotirasida xatolarni nazorat qilish kodi bor bo'lib, u saqlangan ma'lumotlardagi xatoliklarni aniqlaydi. Agar xato sodir bo'lsa, boshqaruvchi sxema uni aniqlaydi va uzilish yordamida bu haqda markaziy protsessor ni ogohlantiradi.

Markaziy protsessor qandaydir nostandart holat yuzaga kelganda yoki dastur komandalarini bajarishda xatoliklar kuzatilganda dasturning bajarilishi uzilishi mumkin. Masalan, komandadagi kodning amali mavjud komandalarning hech birida bajarila olmasa, yoki, nolga bo'lish amali sodir bo'lsa, va shu kabi boshqa holatlar.

Agar bunday xatoliklar yuzaga kelganda xatoliklarni qayta ishlash faollasha, markaziy protsessor uzilishni bajargan holatni bajaradi. U dasturni bajarilishini to'xtatadi va istisnolarni qayta ishlash dasturini ishga tushiradi. Bu dastur xatolikdan keyin dasturni qayta tiklashga urinib ko'radi, agar imkoni bo'lmasa foydalanuvchini bu haqda ogohlantiradi. Kiritish/chiqarish dagi uzilishlarda markaziy protsessor dasturning ishini to'xtatib, keyin uzilishni qayta ishlab boshlaydi. Agar uzilish xatolik sababli bo'lsa, bu komandani bajarish mumkin emas, shu sababdan markaziy protsessor istisnolarni qayta ishlash dasturini zudlik bilan ishga tushiradi.

### 15.5. Interfeys sxemalar

Kiritish/chiqarish qurilmasi interfeysi qurilmani kompyuter shinasi bilan bog'lovchi sxemadir. Bu sxemaning bir tarafida ma'lumot, adres, komanda shina signallari, ikkinchi tarafida (*port* deyiladigan tarafi) ma'lumot va boshqaruv signallarini uzatish liniyalari mavjud. Port parallel yoki ketma-ket bo'lishi mumkin. *Parallel port* u yoki bu yo'nalishda bir vaqtda 8 yoki 16 bit uzata oladi. *Ketma-ket port* esa bir vaqtda bir bit uzatadi. Har ikki xolatda ham shina bilan hamkorlik bir xil uslubda tashkil qilinadi, parallel yoki ketma-ket kelgan axborotni qayta ishlash esa interfeys sxemaning ichki jarayonida bajariladi.

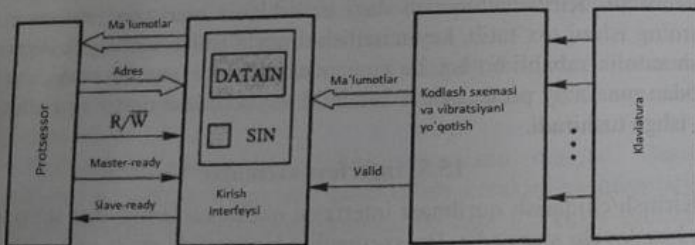
Parallel port orqali ulanish hosil qilinganda ko'pkontaktli razyom va shunga mos kabel ishlatiladi. Kabel uchlaridagi sxemalar sodda, chunki parallel va/yoki ketma-ket signallar formatlarini qayta ishlashga

ehtiyoj yo'q. Bunday ulanish kompyuterga juda yaqin masofada joylashgan qurilmalarga mo'ljallanadi. Agar masofa uzoqlashsa, ko'chirish signali halal berishi natijasida ma'lumot uzatish tezligi keskin kamayib ketadi. Shu sababdan kompyuter va qurilma orasidagi masofa uzoq bo'lganda ketma-ket portni ishlatgan ma'qul.

Umuman olganda, kiritish/chiqarish interfeysi quyidagi amallarni bajaradi:

- Ma'lumotlarni saqlash buferini taqdim etadi
- markaziy protsessor ga holat bayrog'ini namoyish qiladi, markaziy protsessor bufering to'lganligi (ma'lumot kiritishda) yoki bo'shligi (ma'lumot chiqarishda) ni aniqlaydi va shunga mos komandalarni bajaradi
- Qurilma qachon markaziy protsessor tomonidan adreslanishini aniqlovchi adresni dekodlash sxemasiga ega
- Shinani boshqarish sxemalarini ishlashi uchun takt signallari hosil qiladi
- Ma'lumot almashishda, zarur xollarda parallel formatni ketma-ketga o'zgartirishni bajaradi

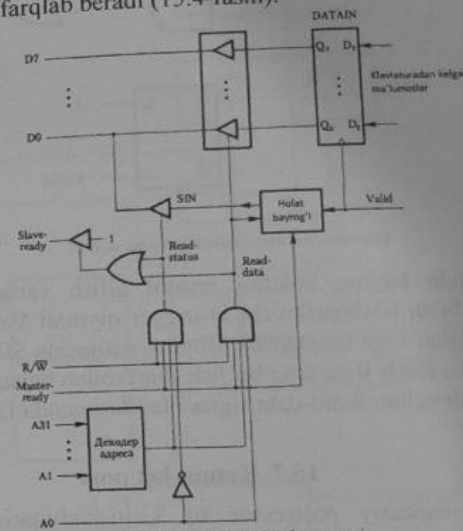
### 15.6. Parallel port



15.3-rasm. Klaviatura va protsessorning ulanish sxemasi

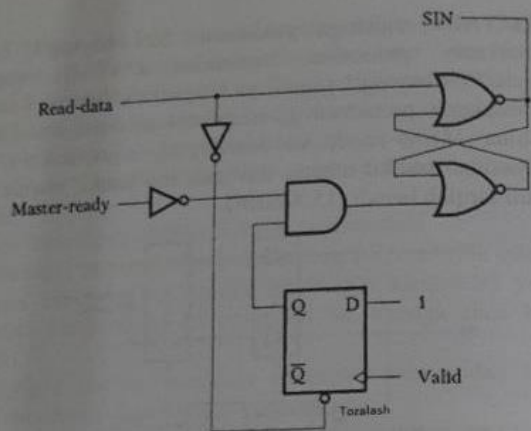
15.3-rasmda klaviatura signalini qayta ishlanishi keltirilgan. Klaviatura oddiy mexanik kalitlardan iborat va ular normal holatda ochiq turadi. Klavish bosilganda kalit yopiladi va elektr toki o'tadi. Bu signal kodlash sxemasida aniqlanadi, u esa mos ASCII kodini faollashtiradi. Kodlash sxemasining chiqish signali simvolning bitli shakli va Valid deb nomlanuvchi, klavish bosilganligini bildiruvchi boshqarish signalidan iborat. Bu ma'lumotlar o'zida ma'lumotlar registri (DATAIN) va holat bayrog'I (SIN) ni saqlovchi interfeys sxemaga uzatiladi. Klavish bosilganda Valid signali 1 ga tenglanadi.

ASCII kod DATAIN registriga yuklanadi, SIN bayrog'I 1 ga tenglanadi. markaziy protsessor tomonidan DATAIN registri o'qilgandan keyin SIN bayrog'I yana 0 ga tenglanadi. Interfeys sxema ma'lumotlarni markaziy protsessor ga uzatuvchi asinxron shina bilan bog'langan. Shina Master-ready va Slave-ready signallari va RW boshqarish liniyasidan tashkil topadi. RW esa, ma'lumki, o'qish yoki yozish amallarini farqlab beradi (15.4-rasm).



15.4-rasm. Kiritishning interfeys sxemasi

DATAIN ning chiqish liniyalari ma'lumotlar shinasini liniyalari bilan uch xolatli takrorlagichlar orqali ulanadi. Ular esa markaziy protsessor shu registr adresidan ma'lumot o'qiganida faollashadi. SIN signali holat bayrog'ga qarab faollashadi. Bu signal ham uch xolatli takrorlagich orqali shinaga uzatiladi. U D0 biti bilan bog'lanadiva holat registrining nolinch razryadi deb qaraladi. Registrning boshqa razryadlari foydali ma'lumotga ega emas. Adres dekoderi kiritish qurilmasi interfeysini tanlashga mo'ljallangan. Bu adres shinadagi interfeysga belgilangan adreslarning yuqori 31 razryadi mos kelganda ishlatiladi. Master-ready signali faollashganda qaysi registrni o'qish kerakligini A0 adres razryadi aniqlaydi. Shu tartibda Read-data va Read-status signallari follahtiriladi (15.5-rasm).

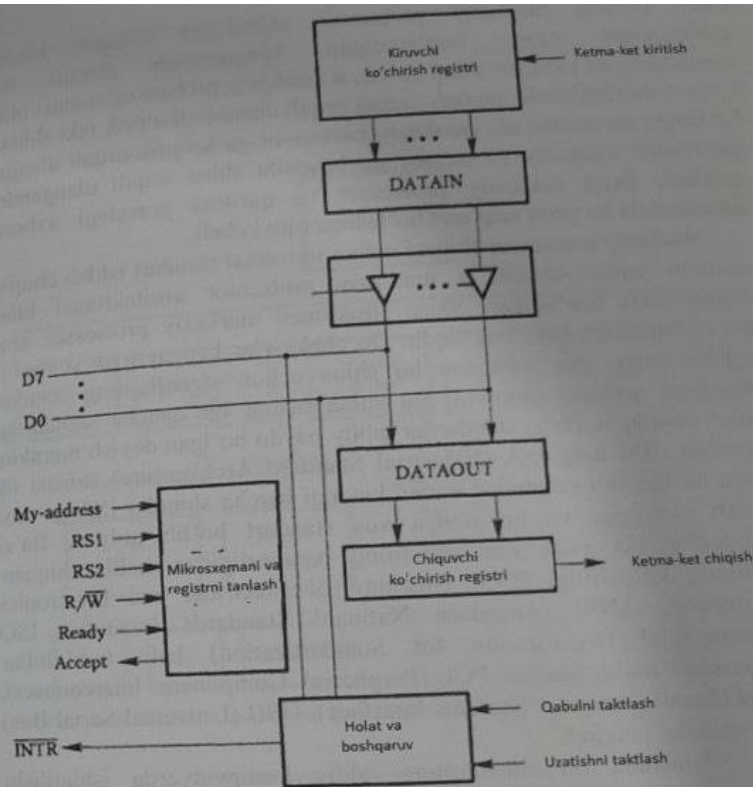


15.5-rasm. Holat bayrog'ini shakllanish sxemasi

15.5-rasmda bayroq holatini tashkil qilish varianti keltirilgan. Signal fronti bilan boshqariluvchi D-trigger qiymati Valid liniyasidagi signal fronti bilan 1 ga tenglanadi. Buning natijasida  $SIN=1$  bo'ladi va bu faqat Master-ready 0 ga teng bo'lish sharti bilan bajariladi. DATAIN registrini o'qish uchun Read-data signali faollashganda D-trigger yana 0 ga tenglanadi.

### 15.7. Ketma-ket port

Bu port markaziy protsessor ba kiritish/chiqarish qurilmalari orasida bitli almashinishga xizmat qiladi. Ketma-ket port interfeys sxemasining alohida xususiyati shundaki, u ma'lumotlarni ketma-ket rejimda qurilma tomonidan, parallel rejimda esa shina tomonidan uzata oladi. Bunday format almashtirishlar parallel murojaat funksiyasiga ega bo'lgan ko'chirish registrlari yordamida bajariladi.



15.6-rasm. Oddiy ketma-ket interfeysning sxemasi

Bu rasmda (15.6) oddiy ketma-ket interfeysning sxemasi keltirilgan. Kiruvchi ko'chirish registri kiritish/chiqarish qurilmasida ketma-ket bitlarni oladi. 8 bit yuklangandan keyin registr ma'lumotlari parallel rejimda DATAIN registriga yuboriladi. Shu shaklda chiquvchi ma'lumotlar DATAOUT registridan chiquvchi registrga joylashtiriladi, undan ketma-ket shaklda tashqi qurilmaga.

### 15.8. Kiritish/chiqarish interfeysi standartlari

Markaziy protsessor shinasi protsessor sxemalari kabi boshqariluvchi shinadir. Unga markaziy protsessor bilan ishlaganda yuqori tezlik talab etiladigan qurilmalar ulanadi, masalan asosiy xotira

### 9.5. Almashish algoritmlari

To'g'ri akslanuvchi keshda har bir blokning o'rnini ma'lum, bloklarni almashirish strategiyasi shart emas. Assosiativ va to'plam-assosiativ keshda esa bloklarni turli usullarda almashirish mumkin. Keshga yangi blokni joylashda u to'lgan bo'lsa, kesh kontrolleri bloklardan birini o'chirishni tanlashi zarur. Buni qanday hal qilishiga esa tizimning unumdorligi bog'liq bo'ladi. Asosiy g'oya shundayki, xotirada yaqin orada kerak bo'ladigan bloklar uchun joy qolishi kerak. Buni qanday aniqlash mumkin? Bu yerda murojaatlarni lokallashtirish prinsipi (tamoyili) ga suyanish mumkin. Ma'lum sohadagi takrorlanuvchi komandalar, ma'lum vaqt oralig'ida ishlatilib turilar ekan, yaqinda murojaat qilingan bloklarga yana murojaat qilish ehtimoli yuqoriroq bo'ladi. Shu sababdan eng uzoq vaqt davomida murojaat qilinmagan blokk endi ehtiyoj sezilmasli ehtimoli yuqori bo'ladi. Bunday algoritmi eng uzoq vaqt ishlatilmagan bloklarni o'chirish algoritmi deyiladi (*Last Recentli Used - LRU*).

LRU ni ishlatish uchun kesh-kontroller barcha bloklarga murojaatni kuzatib borishi kerak. Har bir blokka 2 razryadli hisoblagich ishlatiladi. Dastlab barcha bloklarning hisoblagichlari 0 ga teng bo'ladi. Har murojaatda hisoblagich 1 ga oshadi. Keshda keraklo blok mavjud bo'lmasa, yangi keluvchi blok uchun joy mavjud bo'lsa, yangi blok uchun 0 qiymat o'rnatiladi, keshda mavjud barcha hisoblagichlar qiymati 1 ga oshiriladi. Agar to'plam to'lgan bo'lsa, hisoblagich ko'rsatkichi 3 ga teng blok o'chiriladi, uning o'rniga yangi blok yoziladi. Barcha hisoblagichlarning qiymati 1 ga oshiriladi. Ko'rinib turibdiki, barcha bloklarning ko'rsatkichlari har xil bo'ladi.

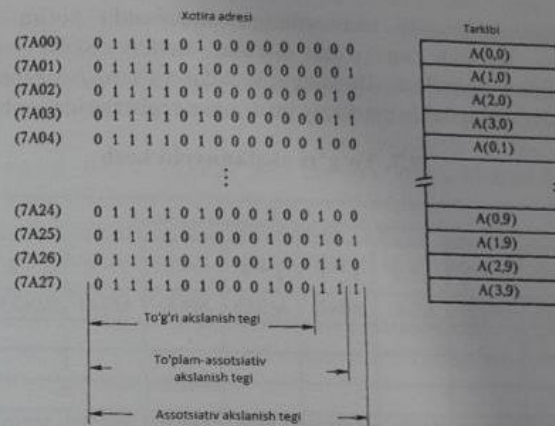
LRU algoritmi keng qo'llaniladi. Ba'zi hollarda uni qo'llash unumdorlikni pasayishiga olib keladi, masalan massiv elementlariga ketma-ket murojaat qilishda, agar massiv ko'p sonli bo'lib, keshga sig'masa va shu kabi holatlarda.

### 9.6. Aralash va bo'lingan kesh xotira. Bir bosqichli va ko'p bosqichli kesh xotira

**Akslanish texnologiyasiga misollar.** Xotirani keshda akslanishiga misol. Faraz qilaylik, Mpda komandalar uchun alohida va ma'lumotlar uchun alohida kesh mavjud. Ma'lumot keshida faqat 8 ta blok bo'lsin. Blok bitta 16 razryadli so'zdan iborat, xotira esa 16 razryadli adreslar

bilan so'zma-so'z adreslanadi. Bloklarni almashirishda LRU algoritmidan foydalanilsin.

Endi quyidagi jarayonni hal qilishda ma'lumotlar keshida sodir bo'ladigan o'zgarishlarni tahlil qilamiz. 4 x 14 razmerli A sonli massivi mavjud, har bir son bitta so'zga teng, asosiy xotirada 7A00 dan 7A27 gacha adreslarda joylashgan bo'lsin. Bu massiv elementlari ustunlar ketma-ketligida saqlanadi (9.5-rasm).



9.5-rasm. Asosiy xotirada saqlanadigan massiv

9.5-rasmda ko'rinib turganidek turli akslanish texnologiyalarida xotira adresi teglari ajratiladi. Bu misolda blokda so'zni aniqlovchi maxsus bitlar yo'q, har blok bitta so'zni saqlaydi deb qabul qilganmiz. Dastur massivning birinchi satri elementlari qiymatini satr elementlarining o'rtacha qiymatiga nisbatan normallashtiradi. Masala sharti har satrning o'rta qiymatini topish va har bir elementni shu qiymatga bo'lishdan iborat. Yechimning matematik ko'rinishi:

$$A(0,i) \leftarrow \frac{A(0,i)}{\left(\sum_{j=0}^9 A(0,j)\right)/10}; i = 0, 1, \dots, 9 \quad (1)$$

Ushbu yechimni bajaruvchi dastur quyidagicha:

```

SUM := 0
for j:=0 to 9 do
  SUM := SUM + A(0,j)
end
AVE := SUM / 10
for i := 9 downto 0 do
  A(0,i) := A(0,i) / AVE
end
9.1-listing. Har bir
elementni o'rtta
arifmetikka bo'lish dasturi

```

9.1-dasturda massiv elementlariga murojaatda xotira adreslari ishlatilgan. Yig'indi va o'rtta qiymat uchun SUM va AVE o'zgaruvchilari ishlatilgan. Bu o'zgaruvchilar va  $i$  va  $j$  o'zgaruvchilari ham hisoblash jarayonida markaziy protsessor registrlarida joylashadi.

### 9.7. To'g'ri akslanuvchi kesh

Sikl bajarilgandan keyin ma'lumotlar keshining tarkibi

Siklning holati	$j-1$	$j-3$	$j-5$	$j-7$	$j-9$	$i-6$	$i-4$	$i-2$	$i-0$
0	A(0,0)	A(0,2)	A(0,4)	A(0,6)	A(0,8)	A(0,6)	A(0,4)	A(0,2)	A(0,0)
1									
2									
3									
4	A(0,1)	A(0,3)	A(0,5)	A(0,7)	A(0,9)	A(0,7)	A(0,5)	A(0,3)	A(0,1)
5									
6									
7									

9.6-rasm. To'g'ri akslanuvchi keshning tarkibi

9.6-rasmda to'g'ri akslanishda keshning tarkibi qanday o'zgarishi ko'rsatilgan. Jadval ustunlarida dasturning ikki siklidan keyingi keshning holati keltirilgan. Masalan, birinchi sikldan ikkinchi marta o'tilganda ( $j=1$ ) keshda A(0,0) va A(0,1) elementlari mavjud bo'ladi. Ular adreslarining 3 ta kichik razryadlari uchun joy qoldirib, 0 va 4 bloklarda saqlanadi. Keyingi siklda A(0,0) elementi xuddi shu blok adresiga ega bo'lgan A(0,2) bilan almashadi. Diqqat qilinadigan joyi – massiv elementlari keshning faqat ikki blokiga mos keladi, boshqa bloklar o'zgarishsiz qoladi, normallashtirish jarayonida esa oxirgi qiymat boshlang'ich qiymatga teng bo'ladi. Birinchi sikldan 10 marta o'tilganda ( $j=9$ ) keshda A(0,8) va A(0,9) elementlari saqlanadi. Ikkinchi siklni bajarishda elementlar teskari tartibda qayta ishlanar ekan, bu

siklning dastlabki ikki o'tishida ishlatiladigan elementlar keshda joylashgan bo'ladi. Uchunchi o'tishda ( $j=7$ ) A(0,8) elementi A(0,7) bilan almashtiriladi, keyin A(0,6) bilan va h.k. shunday qilib, ikkinchi siklni bajarishda keshda 8 ta element almashtiriladi.

**Assosiativ kesh.** Dastur bajarilishi oldidan kesh bo'sh bo'lsa, dastlabki 8 ta o'tishda massiv elementlari ketma-ket o'rinlarga nusxalanadi. Ma'lumotlarni unumli keshlash uchun ikkinchi siklda massiv elementlarini teskari tartibda qayta ishlash zarur. Agar ikkinchi sikl ham birinchi kabi tartibda elementlarni qayta ishlasa nima ro'y beradi? Agar LRU algoritmi ishlatilsa ikkinchi siklda hamma elementlar qayta ishlanmasidan oldin qayta yozilgan bo'ladi.

**To'plam-assosiativ kesh.** Faraz qilaylik, ma'lumotlarning to'plam-assosiativ keshi 4 ta blokli 2 ta to'plamli bo'lsin. Bu esa adresning kichik biti to'plamga mos bo'lgan xotira blokini ko'rsatadi deganidir. Yuqori 15 razryad esa teg ni tashkil qiladi.

Barcha ma'lumotlar bloki juft raqamli adresli ekanligidan, ularning barchasi 0 to'plamda bo'ladi. Shu sababli 6 ta element ikkinchi siklni bajarilishida qayta yuklanadi.

Yuqorida ko'rilgan misollardan eng unumli texnologiya to'liq assosiativ, keyin to'plam-assosiativ akslanishdir. Eng unumdorligi past texnologiya to'g'ri akslanishdir.

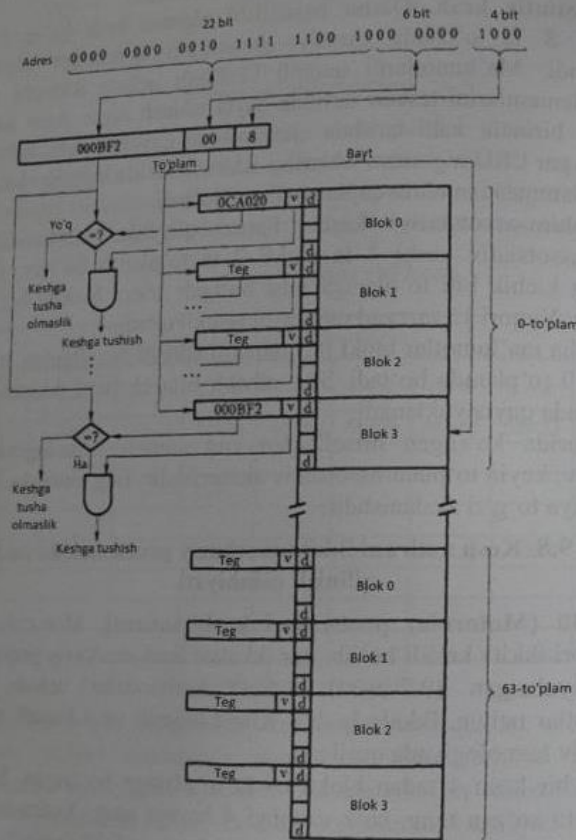
### 9.8. Kesh xotirani ikkita markaziy protsessor da tashkil qilinish uslubiyati

**68040 (Motorola) protsessori kesh xotirasi.** Motorola 68040 protsessori ikkita keshli bo'lib, har ikkalasi ham markaziy protsessorga integratsiyalangan (9.7-rasm). 1-kesh komandalar uchun, 2-kesh ma'lumotlar uchun. Ikkala kesh 4 Kbayt hajmli va 4 kanalli to'plam-assosiativ texnologiyada qurilgan.

Har bir kesh 4 tadan blokli 64 ta to'plamga bo'lingan. Blokning hajmi 4 ta so'zga teng, so'z uzunligi 4 baytga teng. Xotira bloklarini kesh bloklarida akslantirish uchun adreslar 5.22 rasmdagi kabi interpretatsiya qilinadi. 4 ta kichik razryad baytning blokda o'rini aniqlaydi. Keyingi 6 razryad 64 tadan bitta to'plamni tanlaydi, yuqori 22 razryad tegni tashkil qiladi. Tasvirda sonlar 16 lik formatda berilgan.

Keshni boshqarish mexanizmi har bir blok uchun bitta ishonchlik biti, har bir so'z uchun bitta o'zgartirish (modifikatsiya) biti bo'lishini nazarda tutadi. Blok 1-marta keshga yuklanganda uning ishonchlik biti 1 ga tenglanadi. Bir so'zda o'zgarish bo'lib, u qayta yozilsa,

modifikatsiya biti 1 ga tenglanadi va shu xolatda qoladi. Qachonki bu blok keshdan asosiy xotiraga nusxalansa ishonchlilik biti yana 0 ga tenglanadi.



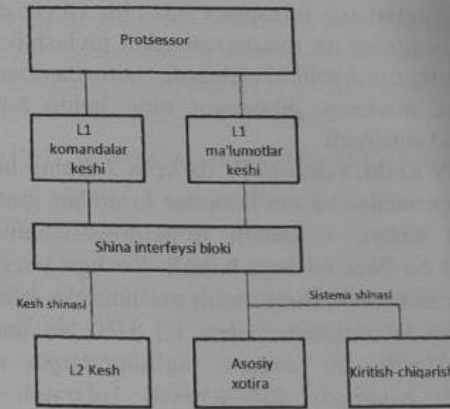
9.7-rasm. Motorola 68040 protsessoring ma'lumotlar keshi

Keshga murojaat qilganda adresning teg razryadlari mos to'plamdagi 4 ta teglar bilan solishtiriladi. Agar ularning biri mos kelsa va shu blokning ishonchlilik biti 1 ga teng bo'lsa, keshga tushish sodir bo'ldi (попадание в кэш). 5.23 rasmdagi misolda adreslanayotgan ma'lumotlar 0-to'plamning 4-blokidagi 3-so'zda topilganligi ko'rsatilgan.

Ma'lumotlar keshi uchun teskari yoki to'g'ri yozuvni tanlashni operatsion tizim aniqlaydi. Komanda keshining tarkibi o'qish xatoligi tufayli yangi komandalarga keshga kelganda o'zgaradi. Yangi blok to'lgan to'plamga joylashtirilishi kerak bo'lganda almashtirish algoritmi o'chirilishi kerak bo'lgan tasodifiy aniqlaydi. Agar tanlangan blokning ishonchlilik biti 1 ga teng bo'lsa, uni o'chirishdan oldin teskari yozuv qo'llaniladi.

**Pentium III, IV keshi.** Pentium III yuqori unumdorlikli markaziy protsessors deb loyihalashtirilgan edi. Unumdorlik ma'lumotlarga va komandalarga murojaat vaqti bilan o'lchanar ekan, bu markaziy protsessors ham 2 darajali kesh bilan ta'minlangan. 1-darajali kesh xotira 16 Kbaytli komanda keshi va ma'lumotlar keshidan iborat. Ma'lumotlar keshi 4 kanalli to'plam-assotsiativ texnologiyali va unga nisbatan ha teskari, ham to'g'ri yozuv qo'llanishi mumkin. Komandalarga keshi 2 kanalli to'plam-assotsiativ texnologiyali. Dastgurning bajarilishi davomida komandalarga modifikatsiyaga uchramasligidan, komanda keshi yozuvi uchun alohida yondashuv shart emas.

2-darajali kesh nisbatan kattaroq hajmga ega. Unda komandalarga ham, ma'lumotlar ham joylashadi. Bu kesh tizimning boshqa qismlari bilan quyidagicha bog'langan:



9.8-rasm. Pentium III protsessoring kesh xotirasi va tashqi ulanishi

Shinali interfeys bloki 3 ta kesh, asosiy xotira va axborot kiritish/chiqarish qurilmalarini bog'laydi. markaziy protsessors va boshqa qurilmalarni aloqasi uchun ikkita alohida shina mavjud: tezkor kesh shinasi, u markaziy protsessors va 2-darajali keshni bog'laydi, va



tizim shinasini, u markaziy protsessorni asosiy xotira va kiritish/chiqarish qurilmalari bilan bog'laydi.

2-darajali kesh markaziy protsessor mikrosxemasidan tashqirida tashkil qilinishi ham mumkin. Pentium III Katmai versiyasida shunday qilingan, bu kesh 512 Kbayt hajmli va SRAM asosida tashkil qilingan. U 4 kanalli to'plam-assotsiativ texnologiyali, to'g'ri va teskari yozuvlarni ishlatadi. Keshning shinasini 64 bit.

Integral sxemalarni qurish texnologiyasini rivojlanishi 2-darajali keshni markaziy protsessorning o'ziga integratsiya qilish imkonini berdi. Bunday struktura Pentium III Coppermine da qo'llanilgan. Uning 2-darajali keshi (L2) 256 Kbaytli, 8 kanalli to'plam-assotsiativ texnologiyali, markaziy protsessor bilan bitta mikrosxemada joylashganligi sababli 256 razryadli shina bilan bog'langan.

Savol paydo bo'lishi mumkin: kesh integratsiyalanganini maqbulmi yoki alohida joylashganini? Tashqi keshni hajmini katta qilish mumkin, lekin markaziy protsessor bilan keng tarmoqli shina aloqasini qurish murakkablashadi, chunki chiqishlar soni ko'payadi, shina uchun platada ko'p joy ajratish lozim, ko'p energiya iste'moli va h.k. Bundan tashqari tashqi kesh kam takt chastotasiga ega. Katmai ning 2-darajali keshi markaziy protsessorga nisbatan 2 barobar past chastotada ishlaydi, Coppermine niki markaziy protsessor bilan bir xil chastotada. 2-darajali keshni markaziy protsessor mikrosxemasiga joylashtirish kutish vaqtini kamaytiradi, o'tkazuvchanlikni oshiradi, natijada unumdorlik oshadi. Kamchiligi esa, markaziy protsessorning hajmi kattalashib, ishlab chiqarish murakkablashadi.

**Pentium IV keshi.** Pentium IV da kesh 3 darajali bo'ladi. 1-darajali kesh ikkita: komandalar va ma'lumotlar keshidan iborat. Ma'lumotlar keshi hajmi 8 Kbayt, 4 kanalli to'plam-assotsiativ texnologiyaga asoslangan. Har bir blok 64 bayt hajmli. Keshga yozish uchun to'g'ri yozuv protokoli ishlatiladi. Butun sonli ma'lumotlar keshdan ikki takt da o'qiladi. Pentium IV mikrosxemalari 1,3 GHz va undan yuqori takt chastotasida ishlaydi, bu degani ma'lumotlarga murojaat 2 ns (nanosekund) da bajariladi. 256 Kbaytli 2-darajali kesh 8 kanalli to'plam-assotsiativ texnologiyali. Har blok hajmi 128 bayt. Yozishda teskari yozuv protokoli ishlatiladi. Murojaat vaqti 7 takt.

1- va 2-darajali keshlar markaziy protsessorga integratsiya qilingan. Pentium IV arxitekturasi markaziy protsessorga 3-darajali keshni ham integratsiyalash imkonini beradi. Bu mikrosxemalar asosan serverli tizimlarda ishlatiladi.

## 10. Virtual xotira tushunchasi, tashqi xotira va ularning turlari. RAID massivlar

**Virtual xotira.** Dastlabki kompyuterlarda xotira juda kichik va narxi baland bo'lgan. 1950-yillarning oxiridagi eng yaxshi kompyuter IBM-650 ning xotirasi 2000 ta so'zga ega bo'lgan. ALGOL ning 60 ta kompilyatori 1024 so'z xotirali kompyuterlar uchun ishlab chiqilgan. U davr dasturchilari o'z dasturini xotiraga joylashtirishni eng asosiy muammo deb bilganlar va bunga juda ko'p vaqtlarini sarflaganlar. Ko'pincha tezkor algoritmlarning o'rniga sekinroq algoritmlarni ishlatishgan, chunki tezkor algoritmlar xotirada ko'p joy egallagan.

Bu muammoni hal qilishda asosan tashqi xotiralarga suyanilgan. Dasturchi dasturni *overlay* deb nomlanuvchi bir necha qismlarga bo'lib, xotiraga joylashtirgan. Dasturni to'liq ishlatish uchun dastlab 1-overley ishlatilgan, u ishini yakunlagach, 2-overlay va h.k. dasturchi dasturni overlaylarga bo'lish masalasini va qaysi overlay yordamchi xotiraning qaysi qismida joylashishi masalasini hal qilgan, yana asosiy va tashqi xotira o'rtasida overlaylar almashinishini nazorat qilgan, umuman, bu jarayonni kompyuter ishtirokisiz o'zi bajargan.

Bunday texnologiya juda murakkab bo'lishiga qaramasdan ancha vaqt davomida qo'llab kelingan. 1961-yilda bir guruh olimlar (Manchester, Angliya) bu jarayonni avtomatik bajarilish uslubini taqdim etgan. Hozirgi kunda **virtual xotira** deb nomlanuvchi bu metod 1960-yillarda ishlab chiqarilgan kompyuterlarda qo'llanilgan. 1970-yillardan keyin esa barcha kompyuterlarda joriy qilingan.

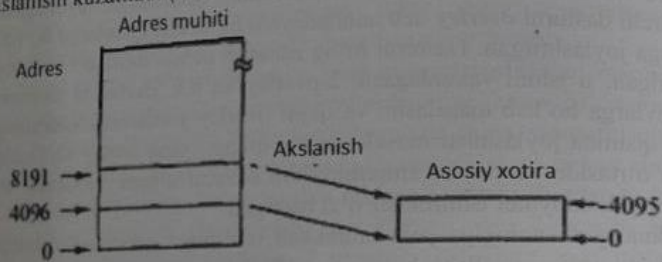
### 10.1. Xotiraning sahifali tashkil qilinishi

Manchesterlik olimlar adres maydoni va xotira adresi tushunchalarini alohidalashni taklif qilishgan. Masalan, 16 bitli adres maydonli 4096 so'zli xotiraga ega mashinani qaraymiz. Bu kompyuterda ishlovchi dastur xotiraning 65 536 ta so'ziga murojaat qilishi mumkin ( $2^{16}=65\,536$ ). **E'tibor bering:** adreslanadigan so'zlar soni adres bitlari soniga bog'liq bo'lib, xotiradagi real so'zlar soniga esa bog'liq emas. Bu xotiraning adres maydoni 0, 1, 2, 3, ..., 65535 dan iborat. Xotira esa faqat 4096 so'zni saqlay oladi xolos.

Bu yerda 4095 gacha va 4096 dan keyingi adreslar orasida keskin farqlanish yuzaga kelgan. 4095 gacha foydali adres makoni, 4096 dan keyingilari foydasiz adres makoni deyilgan. Chunki 4096 dan keyingi adreslar xotiraning real adreslariga mos kelmagan. Adres maydoni va

xotira adresi orasida farqlanish mavjud bo'lmagan, chunki ular orasida o'zaro birxillik munosabati mavjud deb tushunilgan.

Adres maydoni va xotira adresini alohidalash g'oyasi quyidagicha tushuntiriladi. Har bir joriy vaqtda xotiraning 4096 ta so'ziga bevosita murojaat qilish mumkin. Bu esa ushbu so'zlar 0 dan 4095 gacha adresda joylashishi shart degani emas. Masalan, 4096-adresga murojaat qilinganda xotiraning 0-adresidagi so'z ishlatilishi, 4097-adresga murojaat qilinganda xotiraning 1-adresidagi so'z ishlatilishi, ..., 8191-adresga murojaat qilinganda xotiradagi 4095-so'z ishlatilishi lozim va h.k. Boshqacha aytganda adresli maydonni haqiqiy xotira maydonida akslanishi kuzatiladi (10.1-rasm).



10.1-rasm. Xotiraning 4096 dan 8191 gacha virtual adreslari asosiy xotiraning 0 dan 4095 gacha adreslarida akslanadi

Agar dastur 8192 adresdan 12287 adresga o'tishna bajarsa nima bo'ladi? Virtual xotirasiz mashinalarda xatolik sodir bo'ladi, ekranda "Mavjud bo'lmagan xotira adresi" yozuvi chiqadi. Virtual xotirali mashinada esa quyidagilar bajariladi:

- 1) 4096 dan 8191 gacha so'zlar diskda joylashadi
- 2) 8192 dan 12287 gacha so'zlar asosiy xotiraga yuklanadi.
- 3) Adreslarni akslanishi o'zgaradi. Endi 8192 dan 12287 gacha adreslar asosiy xotiradagi 0 dan 4095 gacha yacheykalarga moslanadi.
- 4) Dasturning bajarilishi davom etadi.

Adreslarni bunday avtomatik yoyish texnologiyasi *xotirani sahifali tashkil qilinishi* deyiladi, diskdan o'qiladigan dastur qismlari esa *sahifalar* deyiladi.

Adreslarni adres maydonidan xotiraning real adreslariga akslantirishining yana boshqa murakkabroq usuli ham mavjud. Dastur murojaat qilishi mumkin bo'lgan adreslarni *virtual adres muhiti* deyiladi, xotiradagi real adreslarni esa *fizik adres muhiti* deymiz.

Xotirani taqsimlash sxemasi va sahifalar jadvali virtual adres va fizik adreslarni o'zaro moslashtiradi. Diskda to'liq virtual adres muhitini saqlash uchun joy yetrali deb hisoblanadi (juda bo'lmaganda joriy vaqtda bajarilib turgan qismini).

Dasturlar virtual adres muhitidan so'zlarni o'qishi va/yoki virtual adres muhitiga yozishi mumkin, aslida fizik xotira muhitida bunga joy bo'lmasa ham. Virtual xotira kompyuterning xotirasi katta ekanligi haqida tasavvur uyg'otadi xolos. Aslida kompyuterning fizik xotirasi ancha kichik bo'ladi. Katta hajmli xotira tasavvuri xotiraning sahifli tashkil qilinishidan kelib chiqqanligini dasturiy vositalar bilan aniqlab bo'lmadi. Xohlagan adresga xohlagan vaqtda murojaat qilinsa, ma'lumotlar yoki komandalar chiqib kelaveradi. Bu mexanizm shaffof mexanizm deyiladi.

## 10.2. Xotirani sahifali tashkil qilish

Virtual adres maydon bir necha bir xil hajmli sahiflarga bo'linadi. Odatda ular 512 baytdan 64 Kbaytgacha, kamdam-kam xollarda 4 Mbaytgacha teng bo'ladi. Sahifaning hajmi ikkining darajasiga teng bo'ladi. Fizik adres maydoni ham teng miqdordagi qismlarga shunday bo'linadiki, har bir qism bitta sahifani sig'dira oladi. Asosiy xotiraning bu qismlari sahifali kadrlar deyiladi. Bunda asosiy xotira faqat bitta sahifali kadrlarni saqlaydi.

10.2-rasm (a) da virtual adres muhitini dastlabki 64 K hajmini 4 K hajmli sahifalarga ajratilgani ko'rsatilgan. Adres bayt ham, so'z ham bo'lishi mumkin, bunda ketma-ket so'zlar ketma-ket adreslarni bildiradi. Rasmdagi virtual xotirani sahifalar jadvali yordamida tashkil qilish mumkin. Jadval elementlari soni va virtual xotira sahifalari soni teng bo'ladi. 10.2-rasm (b) da 4 K hajmli 8 ta sahifalik fizik xotira keltirilgan.

32 bitli virtual adresni asosiy xotiraning fizik adreslarida qanday ifodalash mumkin? Ma'lumki, fizik xotira faqat real adreslar bilan ishlaydi, virtual adreslarni qabul qilmaydi. Buning uchun har bir kompyuterda virtual adreslarni fizik adresga aylantirib beruvchi qurilma mavjud bo'ladi. Bu qurilma *xotirani boshqarish kontrolleri (MMU – Memory Management Unit)* deyiladi. U bevosita markaziy protsessor mikrosxemasida yoki alohida joylashishi mumkin. Bizning misolda MMU 32 bitli virtual adresni 15 bitli fizik adresga aylantiradi, shu sababli unga 32 bitli kuruvchi registr va 15 bitli chiquvchi registr kerak bo'ladi (10.3-rasm).

Sahifa	Virtual adreslar
15	61440 P 65535
14	57144 P 61439
13	53248 P 57343
12	49152 P 53247
11	45056 P 49151
10	40960 P 45055
9	36864 P 40959
8	32768 P 36863
7	28672 P 32767
6	24576 P 28671
5	20480 P 24575
4	16384 P 20479
3	12288 P 16383
2	8192 P 12287
1	4096 P 8191
0	0 P 4095

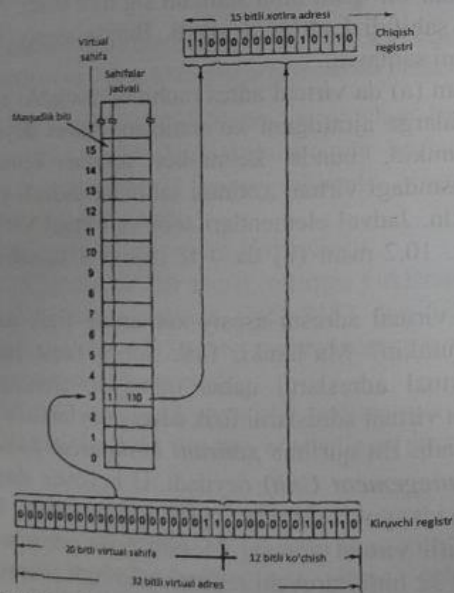
Asosiy xotiraning quyi  
32 K adreslari

Sahifali kadr	Fizik adreslar
7	28672 P 32767
6	24576 P 28671
5	20480 P 24575
4	16384 P 20479
3	12288 P 16383
2	8192 P 12287
1	4096 P 8191
0	0 P 4095

a

b

10.2-rasm. (a) Virtual adres maydonining dastlabki 64 K hajmi har biri 4K dan iborat 16 ta sahifaga bo'linadi. (b) Asosiy xotiraning 32K hajmi biri 4K dan iborat 8 ta kadrlarga bo'linadi



10.3-rasm. Asosiy xotira adresining virtual xotira adreslaridan shakllanishi

10.3-rasmda xotirani boshqarish kontrollerining ishlashi keltirilgan. xotirani boshqarish kontrolleriga 32 bitli virtual adres kelganidan keyin adresni 20 bitli virtual sahifa raqami va 12 bitli shu sahifada ko'chishga ajratadi. Virtual sahifa raqami sahifalar jadvalida ushbu sahifani topish uchun indeks bo'lib ishlatiladi. 10.3-rasmda virtual sahifa raqami 3 ga teng, shu sababli jadvaldan 3-element tanlanadi.

Dastlab xotirani boshqarish kontrolleri kerakli sahifa hozirgi vaqtda xotirada mavjudligini aniqlaydi. Hozirgi vaqtda  $2^{20}$  ta virtual sahifalar va 8 ta sahifali kadr bor ekan, hamma virtual sahifalar bir vaqtda xotirada mavjud bo'la olmaydi. MMU sahifalar jadvali elementining mavjudlik bitini tekshiradi. Bizning holatda bu 1 ga teng va sahifa xotirada mavjudligini bildiradi. Keyin tanlangan elementdan sahifa kadri qiymatini olish lozim (bizning holatda - 6) va uning chiquvchi 15 bitli registrining yuqori 3 bitini nusxalash kerak. Aynan 3 ta bit shartligi, fizik xotirada 8 ta sahifa kadri borligidan kelib chiqadi. Bu operatsiya bilan parallel ravishda virtual adresning quyi 12 biti (sahifadagi ko'chirish maydoni) chiquvchi registrning quyi 12 bitiga nusxalanadi. Shu tartibda hosil qilingan 15 bitli adres keshga yoki asosiy xotiraga yuboriladi.

### 10.3. Qattiq disk

Qattiq disk har bir kompyuterning asosiy tashkil etuvchisidir. Ular asosida qurilgan RAID massivlar tashkil qilinish uslubiyati quyidagicha. (RAID - Redundant Array of Independent Disks)

Ma'lumki, qattiq disklar mustahkamlik zahirasi deb ataluvchi ish vaqtiga va unumdorligiga ta'sir qiluvchi xarakteristikalariga ega.



10.4-rasm. RAIDning ramziy tasviri

Bu diskarning va undagi ma'lumotlarning ishonchli saqlanishini ta'minlash uchun rey-d-massivlar tushunchasi mavjud. Bunday massivlar

tartib raqamlariga ega bo'ladi (0, 1, 2, 3,...) va turli funksiyalarni bajaradi.

RAID – bu disk massivi (yoki kompleksi) bo'lib, bir necha qattiq disklardan tashkil qilinadi. Ba'zan bunday tashkil qilinishdan ma'lumotlarni ishonchli saqlash ko'zlanadi, ba'zan o'qish/yozish tezligini oshirish ko'zlanadi, ba'zanhar ikkalasi ham. Shu kabi vazifalardan birini bajaruvchi RAID massivni foydalanuvchi ehtiyojiga moslashtiradi. Boshqacha aytganda RAIDning konfiguratsiyasini tanlaydi. Bu konfiguratsiyaning turli tiplari turlicha raqamlanadi, 1, 2, 3, 4,... va turli vazifalarni bajaradi.

RAIDLarni ishlatish 1 ta qattiq diskni ishlatishdan ko'ra qulay va unumdorligi keskin seziladi. Boshqa qurilmalarni kuzatadigan bo'lsak, masalan markaziy protsessor da yadro, chastota, kesh, arxitektura; videokartalarda pikseli konveyer soni, xotira hajmi va razryadliligi, sheyder bloklari; O.X.Q esa chastota, tayming kabi tashkil qiluvchilardan iborat va ularning qaysidir bir elementining unumdorligi oshirilsa, shu qurilma unumdorligi ham oshadi. Qattiq disklar esa faqat hajm va golovka aylanish soniga bog'liq bo'sa, bu ko'rsatkich 7200 ayl/min ga chiqqandan keyin o'zgargani yoq. Kompyuter tizimini asosiy tashkil etuvchisi bo'lib hisoblansada, unumdorlik bo'yicha xotira diski o'mida qotib qolgan deyish mumkin.

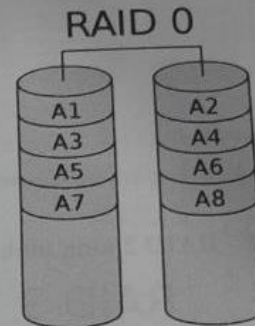
Texnik jihatdan RAID massiv bu – 2 dan ortiq qattiq diskni ona platada joylashtirish va dastur yordamida massiv hosil qilishdir. Bu dastur RAID tipini tanlash, RAIDga qo'shilgan disklar qanday tartibda ishlashini aniqlaydi va ta'minlaydi.

Dasturiy jihatdan qaraganda RAID hosil qilingan tizimda hech qanday alohida xususiyatlar namoyon bo'lmaydi. Faqat BIOS ning sozlanmalarida o'zgarish bo'ladi, shuningdek drayverni ishlatishda ham. Boshqa o'zgarish sezilmaydi, o'sha-o'sha Мой компьютер, papkalar, fayllar, C, D disklar va h.k.

Massivni o'rnatish uchun ona plata RAID texnologiyada ishlashi talab etiladi. Bu plataga 2 ta (yoki undan ortiq) to'liq bir xil bo'lgan (hajmi, keshi, interfeysi va h.k.) xarakteristikali va ishlab chiqaruvchisi va modeli ham bir xil bo'lgan disklar ona plataga o'rnatiladi. Keyin kompyuter ishga tushiriladi, BIOSga kirib, **Configuration: RAID** deb o'rnatiladi.

Kompyuterni qayta yuklab, yuklanish jarayonida disklar haqidagi ma'lumot ekranga chiqqanda **Ctrl+I** bosiladi, va RAID sozlanmalari o'rnatiladi.

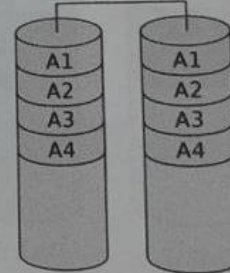
RAIDni hosil qilayotganda yoki o'chirayotganda disklardagi hamma ma'lumotlar **o'chiriladi**, shu sababli RAID bilan eksperiment o'tkazilganda kerakmas disklar bilan ishlagan ma'qul. RAID massivlarning turli xillari mavjudligidan quyidagilarni alohida ta'kidlash mumkin:  
RAID 0 – o'qish/yozish tezligini oshirish uchun mo'ljallangan tizim



10.5-rasm. RAID 0 ning ramziy tasviri

### RAID 1 – akslanuvchi disklar massivi

#### RAID 1



Диск 0      Диск 1

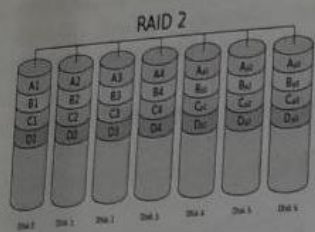
a



b

10.6-rasm: (a) RAID 1 ning ramziy tasvir; (b) Akslanuvchi massivni hosil qilish uchun minimum 2 ta disk zarur

### RAID 2 – Xemming kodi ishlatiladigan massiv

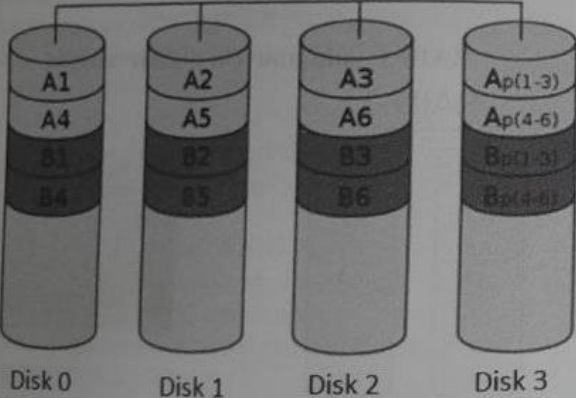


Uzunliklar paritet 1 pariteti	Uzunliklar paritet 2 pariteti	Paritetliklar pariteti 3	Uzunlik pariteti
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20
21	22	23	24
25	26	27	28
29	30	31	32
33	34	35	36
37	38	39	40
41	42	43	44
45	46	47	48
49	50	51	52
53	54	55	56
57	58	59	60
61	62	63	64
65	66	67	68
69	70	71	72
73	74	75	76
77	78	79	80
81	82	83	84
85	86	87	88
89	90	91	92
93	94	95	96
97	98	99	100

10.7-rasm. RAID 2 ning ramziy tasviri va RAID 2 ni tashkil qilish uchun foydalanilgan disklar soni

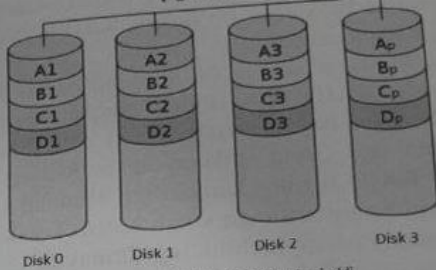
### RAID 3 – RAID 2 ning mukammal shakli

#### RAID 3



10.8-rasm. RAID 3 ning ramziy tasviri

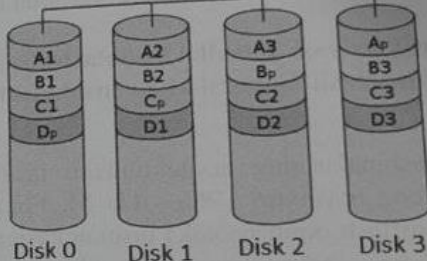
### RAID 4 – WAFL fayl sistemasida guruhli yozishda RAID 4



10.9-rasm. RAID 4 ning ramziy shakli

### RAID 5 – barcha disklarga parallel yozish imkoniyati

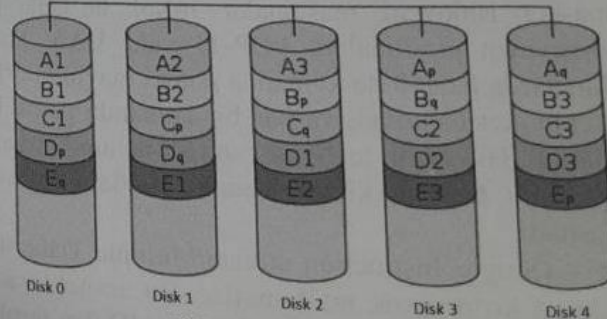
#### RAID 5



10.10-rasm. RAID 5 ning ramziy shakli

### RAID 6 – Rid-Solomon kodida ishlovchi tizim. Ikkita disk ishdan chiqqanda ham ishchanlik qobiliyati saqlab qolinadi

#### RAID 6



10.11-rasm. RAID 6 ning ramziy shakli

**RAID 0** da ikkidan 4 gacha qattiq disklar birgalikda axborotlarni qayta ishlaydi. Unumdorlik yuqori bo'ladi. Dasturiy tashkil qilinishi jihatidan axborotlar bloklarga bo'linadi, har bir diskda alohida qayta ishlanadi va diskarga navbat bilan yoziladi.

**RAID 0** da fayllar bloklarga bo'linib, bir necha disklarda saqlanishi jihatidan, qaysidir RAID disk ishdan chiqsa, faylning shu diskdagi qismi yo'qoladi va uni qayta tiklash imkoni yo'q. Shunday hodisa sodir bo'lganda nechta fayl qayta ishlanayotgan bo'lsa, ularning hammasi yo'qotiladi. Bu RAID 0 ning kamchiligi. Bundan qochish uchun har zamonda fayllarni tashqi xotiraga yozib borish kerak. RAIDning tezligi yuqoriligidan bunday ish qiyinchilik tug'dirmaydi.

**RAID 1** da 2ta diskning bittasidagi ma'lumotlar ikkinchisiga ko'chiriladi, natijada 2-diskni ishlatish imkoni yo'qoladi. Lekin bu xavfsizlikni oshiradi. Bir diskni ishdan chiqishi bilan ma'lumotlar ikkinchisida saqlanib turaveradi. Odatda serverlarda ishlatiladi.

### 11. Flinn klassifikatsiyasi. Parallel hisoblash tizimlari. Konveyerli hisoblash. Superskalyar hisoblashlar

Hisoblash mashinalarining arxitekturasini tasniflashning birinchi taklif etilgani va eng mashxuri 1966 yilda M. Flinn tomonidan taklif etilganidir. Bu tasniflash oqim (potok) tushunchasiga asoslanadi. *Oqim (Potok) – markaziy protsessor tomonidan qayta ishlanadigan ma'lumotlar, komandalar yoki elementlarning ketma-ketligidir.* Komandalar potoki soni va ma'lumotlar potoki soniga ko'ra Flinn arxitekturani 4 ta sinfga ajratishni taklif etadi.

1) **SISD** (Single Instruction stream/Single Data stream) – bitta komandalar potoki va bitta ma'lumotlar potoki. Bu sinfga klassik klassik ketma-ket ishlovchi mashinalar misol bo'ladi, boshqacha aytganda fon-neyman mashinalari. PDP-11 yoki VAX 11/780 kabi. Bunday mashinalarda faqat bitta komanda potoki mavjud bo'lib, barcha komandalar ketma-ket bajariladi va har bir komanda potok bilan bitta operatsiya bajaradi. Hisoblash tezligini oshirish uchun konveyerli tizim ishlatilishi ham shu tizimga kiradi, chunki bunda ham komandalar ketma-ket bajariladi.

2) **SIMD** – (Single Instruction stream/Multiple Data stream) bir potok komanda va ko'p potok ma'lumotlar. Bu arxitekturada yagona potokli komanda tizimi saqlanib qolgan, faqat SIMD dan farqli ravishda

vektorli komandalar qo'shilgan. Vektorli komandalar esa bitta arifmetik amalni vektorning elementlari ustida (boshqacha aytganda ko'plab ma'lumotlar bilan) bajaradi. vektorli amallarni qanday bajarilishini tashkil qilinishi ahamiyatsiz – u markaziy protsessor matritsasi shaklida (ILLIAC-IV kabi) yoki konvaeyer shaklida bajarilishi mumkin (CRAY-1 kabi).

3) **MISD** – (Multiple Instruction stream/Single Data stream) ko'p potok komanda va bir potok ma'lumotlar. Bunda bitta masalani hal qilish bilan shug'ullanadigan ko'p protsessorli arxitekturalar tushuniladi. Hozirgacha aynan bir hisoblash mashinasini bu texnologiyaga tegishli deb qat'iy xulosa qilinmagan. Ba'zi kuzatuvchilar konveyerli mashinalarni shu turga kiritish mumkin deb hisoblashadi, lekin bu ilmiy tarafdin asoslanmaganligi sababli bunday tipdagi mashinalar amaliyotda mavjud emas deb hisoblanadi.

4) **MIMD** – (Multiple Instruction stream/Multiple Data stream) ko'p potokli komandalar va ko'p potokli ma'lumotlar. Bu sinfdagi mashinalarda ma'lumotlarni qayta ishlovchi bir necha qurilmalar mavjud va har bir qurilma o'zining komandalari bilan o'ziga tegishli ma'lumotlarni parallel ravishda bajaradi.

SISD tipli mashinalarga bir protsessorli ketma-ket mashinalar kireadi dedik, masalan VAX 11/780. Ba'zi izlanuvchilar bu tipli mashinalarga CRAY-1 kabi vektorli mashinalarni ham taalluqli deb hisoblaydilar, chunki vektorli-konveyerli mashinalarda vektorni bitta komanda uchun bo'linmas birlik deb qarash mumkin. Bu aniq ilmiy tasdiqlangan emas, shu sababli SIMDning aniq vakillari deb ILLIAC IV va boshqalarni bemalol aytish mumkin. Bunday mashinalarda bitta boshqarish qurilmasi ko'plab markaziy protsessor larni ishini boshqaradi. Har bir markaziy protsessor har bir joriy vaqtda boshqarish qurilmasidan bitta komandano oladi va o'zidagi ma'lumotlarga nisbatan ishlatadi. Klassik (*klassika-namuna*) protsessor matritsalarini uchun qo'shimcha talablar qo'yilmaydi, lekin, ishlash prinsipiga ko'ra bu sinfga CRAY-1 kabi vektorli-konveyerli mashinalarni ham kiritish mumkin. Bunday xolda vektorning har bir elementini ma'lumotlar potokining alohida elementi sifatida qarash kerak bo'ladi.

MIMD sinfi esa ancha keng tarqalgan hisoblanadi, chunki keng tarqalgan Cm\*, C.mmp, CRAY Y-markaziy protsessor, Denelcor HEP, Intel Paragon, CRAYT3D kabi multiprotsessorli tizimlar asosiga qurilgan bo'ladi. Agar konveyerli qayta ishlashni komandalar potokini qayta ishlash deb hisoblanadigan bo'lsa, (konveyer darajalari amallari)

yuqoridagi barcha vektor-konveyerli mashinalarni ushbu sinfga tegishli deb hisoblash mumkin bo'ladi.

Ushbu taklif qilingan klassifikatsiya sxemasi keng qo'llaniladi. Bunda kompyuterning dastlabki xarakteristikalariga qaraladi. Agar kompyuter SIMD yoki MIMD sinfli deyilsa, uning ishlash prinsipi shu jumla bilan tushunarli bo'ladi. Bu sxemaning kamchiliklar ham mavjud. Masalan, ko'p amaliyotda ishlatiladigan alohida arxitekturalar, misol uchun dataflow yoki vektorli-konveyerli mashinalar bunday klassifikatsiyaning qaysi turiga mansubligi aniq ma'lum emas. Yana bir kamchiligi MIMD sinfining haddan tashqari ko'pligi. Bu sinfni yanada aniqroq tasniflovchi boshqa sxemaga ehtiyoj paydo bo'lmoqda. Masalan, protsessorlar soni, tabiati, ular orasidagi munosabat topologiyasi, xotirani tashkil qilinishi, dasturlashtirish texnologiyasi kabi ko'rsatkichlar bo'yicha alohida sinflar tashkil qilish mumkin.

MISD kabi (hozircha) bo'sh sinfning mavjudligi esa sxemaning kamchiligi deb qaralmasligi lozim. Ba'zi izlanuvchilarning fikriga ko'ra bunday sinflar hisoblash mashinalarini qurishda yangi konsepsiyalar ishlab chiqishda foydali bo'ladi.

### 11.1. Parallel hisoblash tizimlari

Parallel hisoblash tizimlarini qurishda turli maqsadlar ko'zda tutiladi, masalan

- Unumdorlikni oshirish
- Tannarxni kamaytirish (yoki unumdorlik/narx ko'rsatkichini minimallashtirish)
- Ishlash ishonchligini oshirish va h.k.

Parallel hisoblash tizimlari quyidagicha tasniflanishi mumkin

1. Komandalar potoki va ma'lumotlar potokining o'zaro munosabatiga ko'ra (Flinn klassifikatsiyasi, yuqorida batafsil ko'rildi)
2. Boshqarish potokiga ko'ra:
  - IF – instruction flow (komanda potoki boshqaradi),
  - DF – data flow (ma'lumotlar potoki boshqaradi)
3. Xotirani ishlatilishiga ko'ra:
  - umumiy xotirali (ajratilgan xotira), **Shared Memory**
  - lokal xotiral – har bir markaziy protsessor ga alohida xotira (taqsimlangan xotira) **Distributed Memory**
4. Protsessorlararo almashinish uslubiga ko'ra:
  - umumiy ajratilgan (yoki taqsimlangan) xotira orqali
  - xabar almashinish orqali

5. Qo'llanilgan parallellik uslubiyatiga ko'ra

- tabiiy yoki vektorli parallellizm (vektorli va matritsali hisoblash tizimlarida)
- mustaqil tarmoqlar parallelligi ("yirik" parallellik, simmetrik ko'pprotsessorli tizimlarda – SMD)
- "mayda" parallellik (MIMD da, umumiy parallel tizimlarda)
- Aralash amallar parallelligi (*Instruction Level Parallelism – ILP*)

uzun so'zli komandali mashinalarda VLIW ha b.

6. Ma'lumotlarni yuklash shakliga ko'ra:

- Ketma-ket yuklash (ketma-ket kod bilan, bitli tartibda)
- Parallel yuklash (so'zli tartibda)
- Aralash tartibda

7. Aloqa tizimi bo'yicha

- To'liq aloqa tipli (har bir markaziy protsessor barcha markaziy protsessor bilan bog'langan)
- Ajratilgan kommutator tipli
- Umumiy shina tipli
- Chiziqli yoki matritsali massiv shaklida (o'zaro yondosh markaziy protsessorlar bog'lanishi)

lekin massiv ko'p o'lchamli – 3,4, va h.k)

- Ulanish konfiratsiyasini o'zgartirish mumkin bo'lgan parallel mashinalar

▪ Dasturlanadigan kommutatorlar

8. Ulanish tizimi murakkabligiga ko'ra:

- Ulanish zanjiri (tarmog'i) ga ko'ra – bir darajali ulanish
- Klasterli ulanish – bir guruhga ulangan mashinalarning ulanish sxemasi ushbu guruhlar o'zaro ulangan sxemadan farqli bo'lganda

9. Hisoblash tarmoqlarining taqsimlanganligi bo'yicha:

- Lokal hisoblash tarmoqlari
- Hisoblash komplekslari – kuchli bog'langan (umumiy xotira orqali almashinish) va kuchsiz bog'langan (xabar almashinish orqali)
- Klasterli arxitektura (hisoblash kompleksi varianti sifatida)
- Taqsimlangan hisoblash komplekslari, jumladan hisoblash mashinalari tamorlari va taqsimlangan klaster tizimlar

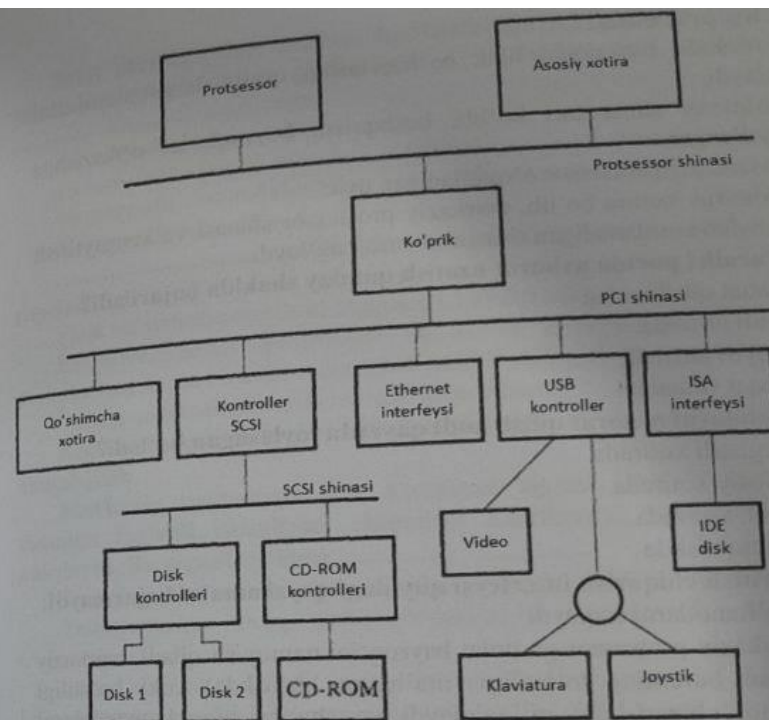
### 11.2. Ommaviy parallel tizimlar

Massiv-parallel protsessorlar – Massive Parallel Processing – bular MIMD tipli "mayda" parallellikka asoslangan tizimlardir. Bunday

kabi. Platada markaziy protsessor shinasidan tashqari boshqa qurilmalarni o'zaro bog'laydigan, kengaytirish shinasini deb nomlanadigan yana bir shina bo'lib, u markaziy protsessor shinasini bilan ko'priq deb ataluvchi maxsus sxema orqali ulanadi. Ko'priq ikki shinani bir-biriga moslashtiradi. markaziy protsessor ga ko'priq orqali ulangan qurilmalar markaziy protsessor ga bevosita shina orqali ulangandek qaraladi, faqat markaziy protsessor va qurilma orasidagi axborot almashishda ko'proq vaqt sarf bo'lishiga olib keladi.

Markaziy protsessor shinasini uchun universal standart ishlab chiqish mumkin emas, chunki u markaziy protsessor arxitekturasi bilan chambarchas bog'langan. Shina strukturasi markaziy protsessor ning takt chastotasiga ham bog'liqdir. Bu cheklovlar kengaytirish shinasiga tegishli emas, shu sababdan bu shina uchun signallarning standart sxemasini qo'llash mumkin. Bu shina uchun bir qancha standartlar ishlab chiqilgan. Ba'zi standartlar tabiiy paydo bo'lgan deyish mumkin. Masalan IBM ning ISA (Industrial Standard Architecture) shinasini ilk shina bo'lganligi sababidan undan keyingi barcha shinalar ISA ga mos iahlab chiqilgan va bu o'ziga xos standart bo'lib qolgan. Ba'zi standartlar esa yirik korxonalar hamkorligidan kelib chiqqan. Ularning ko'pchiligi IEEE (Institute of Electrical and Electronics Engineers), ANSI (American National Standards Institute), ISO (International Organization for Standardization) kabi tashkilotlar tomonidan ma'qullangan PCI (Peripheral Component Interconnect), SCSI (Small Computer Systems Interface), USB (Universal Serial Bus) tipli shinalar mavjud.

15.7-rasmda bu shinalarning oddiy kompyuterda ishlatilishi ko'rsatilgan. PCI shinasini plataning kengaytirish shinasini bo'lsa, SCSI va USB qo'shimcha qurilmalarni nafaqat tashqi, balki ichki tomondan ham qo'shish imkonini beradi.



15.7-rasm. Bir necha interfeyslar qo'llanilgan kompyuter tizimiga misol

### Mavzu yuzasidan test savollari

- Ketma-ket portning asosiy vazifasi nimada ko'rinadi?**  
 markaziy protsessor va kiritish/chiqarish qurilmalari orasida bitli almashinishga xizmat qiladi  
 Qo'shimcha qurilmalarni ulashda foydalaniladi  
 Ko'priq vazifasini bajaradi  
 Tashqi qurilmalarni o'zaro munosabatini tashkil qiluvchi port



### Ko'prik nima?

Alohida funksional blok bo'lib, tashqi qurilmalarga murojaatni cheklaydi.

Maxsus shina turi bo'lib, boshqarish buyruqlarini o'tkazishga mo'ljallangan.

Markaziy protsessor o'rnatiladigan uya.

Maxsus sxema bo'lib, markaziy protsessor shinasi va kengaytirish shinasi deb nomlanadigan shinani o'zaro bog'laydi.

### Parallel portda axborot uzatish qanday shaklda bajariladi?

Faqat qurilmadan

Bitli tartibda

Baytli tartibda

Faqat shinadan

### Xatolarni nazorat qilish kodi qayerda joylashgan bo'ladi?

Registrlı xotirada

Asosiy xotirada

Kesh xotirada

Qattiq diskda

### Kiritish/chiqarish interfeysi quyidagi qaysi amalni bajarmaydi:

Ma'lumotlarni saqlaydi

markaziy protsessor ga holat bayrog'ini namoyish qiladi, markaziy protsessor buferning to'lanligi (ma'lumot kiritishda) yoki bo'shligi (ma'lumot chiqarishda) ni aniqlaydi va shunga mos komandalarni bajaradi

Qurilma qachon markaziy protsessor tomonidan adreslanishini aniqlovchi adresni dekodlash sxemasiga ega

Shinani boshqarish sxemalarini ishlashi uchun takt signallari hosil qiladi

### Uzilishning mohiyati:

Bir dasturning ishini vaqtinchalik to'xtatib, ikkinchi dasturning ishini to'xtagan joyidan boshlab ketish

Maxsus dasturlar bilan markaziy protsessor bajarayotgan hamma jarayonlarni to'xtatish

markaziy protsessor va kiritish/chiqarish qurilmalari orasidagi aloqa interfeysi

Barcha holat to'g'ri

### Intel protsessorlari maxsus kiritish/chiqarish komandalari va kiritish/chiqarish qurilmalari uchun alohida ega

16 razryadli portga

16 razryadli xotirani adreslash uslubiyatiga

16 razryadli komandalar bilan ishlovchi maxsus vositaga

16 razryadli adres maydoniga

### .....kiritish/chiqarishni xotirada akatlanishi deyiladi

Fizik va virtual adreslarni taqqoslab, kerakli adresni topish

Kiritish/chiqarish qurilmalari va xotira bitta adres muhitidan foydalangan holda kiritish/chiqarishni tashkil qilinishi

Tashqi qurilmalarni uzilishlarini qayta ishlash

Xotiraning turli darajalari tashqi qurilmalar haqida orasida axborot almashinish

### Markaziy protsessor dan uzatilgan signal kiritish/chiqarish tizimiga tegishli ekanligini shinadan uzatiluvchi maxsus signal aniqlaydi. Bu signalni olgan .....

Tashqi qurilma, o'zining so'rovi bo'yicha shinani band qiladi

Tashqi qurilma boshqa tashqi qurilmalar ishini bloklaydi, ustunlikni o'ziga oladi

Xotira uni inkor qiladi, kiritish/chiqarish qurilmalari esa shinadan kelgan adresning kichik razryadlarini analiz qilib, qaysi qurilmaga tegishligini aniqlaydi

Xotira uni qayta ishlaydi, tegishligi bo'yicha qurilmaga uzatadi

### Uzilish so'rovlari liniyasi nima?

Uzilishlar uchun shinaning bitta ajratilgan boshqaruv liniyasi

Uzilish so'rovini yuborgan qurilmaning egallagan shinasi

Uzilish so'rovini qayta ishlovchi dasturning uzatgan ma'lumotlari

Uzilishlar uchun so'rov yuborganda sarflangan vaqt

**16. Protsessorlarning tuzilishlari va xususiyatlari. Markaziy protsessor qurilmasining tuzilishi. Registrlar. Protsessorga kiruvchi va chiquvchi boshqarish signallari. Mashina takti. Komanda sikli va mashina takti sikli**

CPU (Central Processing Unit) – markaziy protsessor (markaziy protsessor). “Markaziy” deyilishi dastlabki kompyuterlarda faqat bitta protsessor bo‘lib, hamma ishni o‘zi bajargan va o‘zi nazorat qilgan. Texnologiyalarning rivojlanishi va foydalauvchilarning talablarini inobatga olib, markaziy protsessor ning ichki tuzilishi doimiy ravishda rivojlanib, mukallashib bormoqda. Yuqori unumdorlikli markaziy protsessorlar i/chiqishning umumiy yo‘nalishi – ko‘p sonli turli qurilmalarning parallel ishlashini ta‘minlashga qaratilgan. Xususan, bunda konveyerli yondashuv qo‘llaniladi, chunki konveyerda birinchi komanda ishini tugatmasdan oldin ikkinchi komanda bajarilishni boshlaydi. Yana bir yondashuv – superskalyar yondashuvda xotiradan bir vaqtda bir necha komanda o‘qiladi va bajariladi.

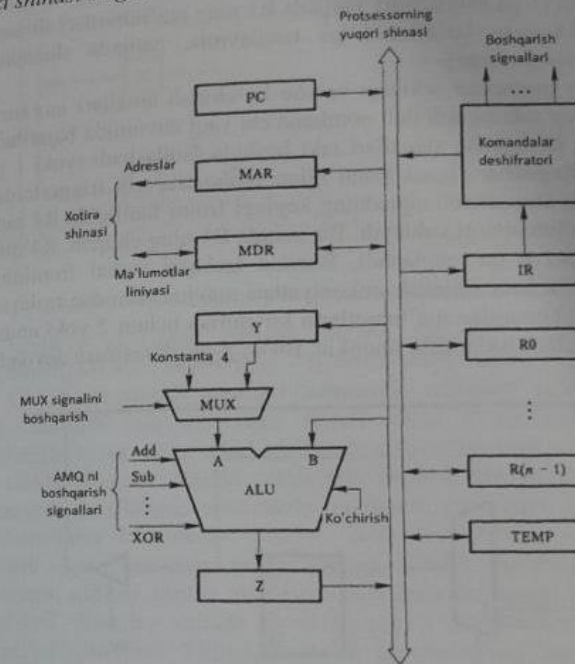
16.1-rasmda yagona shinali markaziy protsessorning arxitekturasi keltirilgan. Hamma registrlar va arifmetik-mantiqiy qurilma shina bilan bog‘langan. Bu shina markaziy protsessor ning ichki shinasi bo‘lib, markaziy protsessor ni boshqa qurilmalar bilan ulash shinasi bilan adashtirmaslik lozim.

Tashqi shinaning ma‘lumotlar liniyasi va adres liniyasi markaziy protsessor ning ichki shinasi bilan xotira registri (MDR) va adres registri (MAR) orqali ulangan. MDR registrida ikkita kirish va ikkita chiqish bor. Unga ma‘lumotlar tashqarida yoki markaziy protsessor dan ichki shinadan kelishi mumkin. Shuningdek, MDR dagi ma‘lumotlar shu ikki shinaga uzatilishi ham mumkin.

markaziy protsessor registrlari soni ( $R_0$  dan  $R_{n-1}$  gacha) turli markaziy protsessorlarda turli sonda bo‘ladi. Bu umumiy mo‘ljallangan registrlar bo‘lib, dasturlarda ishlatiladi. Ularning ba‘zilari maxsus registr deb belgilanishi ham mumkin, masalan indeksli registr yoki stek ko‘rsatkichi kabi.

Dastur komandalari bajarilishi mobaynida ma‘lumotlar bir registrdan boshqasiga uzatiladi, qayta ishlaganda esa arifmetik-mantiqiy qurilma ga uzatiladi va qayta ishlanadi. *Komandalar deshifratori va mantiqiy boshqaruvchi blok* IR registriga yuklangan komandani bajaradi. Tanlangan komandani bajarish va ma‘lumotlarni uzatish uchun registrlarga kerak bo‘ladigan komandalarni deshifrator faollashtiradi.

*Registrlar, arifmetik-mantiqiy qurilma va markaziy protsessor ning ichki shinasi birgalikda ma‘lumotlar trakti ni tashkil etadi.*



16.1-rasm. Yagona shinali protsessorida ma‘lumotlar traktini tashkil qilinishi

Komandani bajarilish jarayoni quyidagi bosqichlardan iborat bo‘ladi:

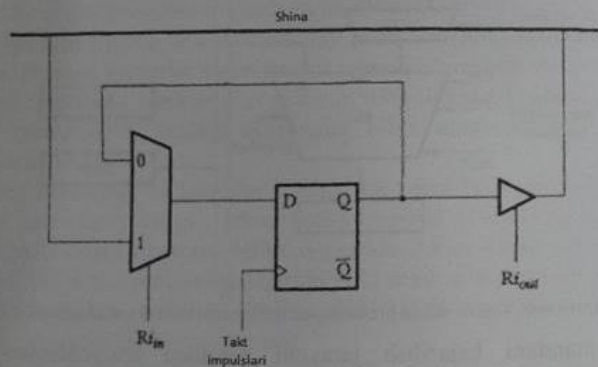
- Ma‘lumotlarni bir registrdan boshqasiga yoki arifmetik-mantiqiy qurilma ga uzatish;
- Arifmetik yoki mantiqiy amalni bajarish va natijani markaziy protsessor registrida saqlash
- Berilgan xotira adresining tarkibini protsessor registriga yuklash

**16.1. Ma‘lumotlarni bir registrdan boshqasiga yuborish**

Komanda bajarilayotgan vaqtda ma‘lumotlar doimiy ravishda u registrdan bu registrga o‘tib turadi. Registrdagi ma‘lumotni shinaga yuklash va shinadan registrga yuklash amalini maxsus sxema bajaradi.

Misol uchun R1 registrdagi ma'lumotlarni R4 registrga ko'chirish talab etilgan bo'lsin. Bu quyidagicha bajariladi: 1) R1 ning chiqishi faollashtiriladi (1 ga tenglanadi), natijada R1 ning ma'lumotlari shinaga o'tadi; 2) R4 ning kirishini 1 ga tenglaymiz, natijada shinadagi ma'lumotlar R4 ga o'tadi.

Markaziy protsessor ichidagi barcha ko'chirish amallari *markaziy protsessor ning takt signali* deb nomlanuvchi vaqt davomida bajariladi. Ko'chirishni boshqarish signallari takt boshida faollashadi (yoki 1 ga tenglanadi). Registrlar signal fronti bilan boshqariluvchi triggerlardan tashkil topadi. Shu sababli signalning keyingi fronti faolligida R4 ning triggerlari ma'lumotlarni yuklaydi. Bir vaqtda R1 ning chiqish, R4 ning kirish signallari 0 ga tenglanadi. Bundan tashqari signal frontining oldini va oxirini ham ishlatish imkoniyatlari mavjud. Bundan tashqari, triggerlarni ishlatmasdan ma'lumotlarni ko'chirish uchun 2 yoki undan ko'p takt signallari ishlatilishi mumkin. Bu *ko'p fazali taktlash* deyiladi.

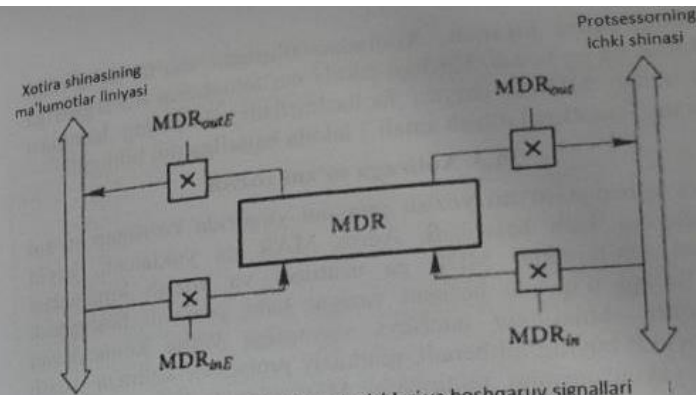


16.2-rasm. Registrning bitta razryadining kirish va chiqish ventillari

### 16.2. Xotiradan so'zni o'qish

Xotiradan so'zni o'qish uchun markaziy protsessor so'z adresini va o'qish komandasini yuborishi kerak. Bunda xotiradan komanda o'qiladimi yoki ma'lumotlarni, ahamiyati yo'q. So'zning adresi chiqishlari xotira shinasini bilan bog'langan MAR registriga joylastiriladi. Bir vaqtda xotira shinasining boshqaruvchi liniyalari orqali o'qish komandasini yuboradi. Xotiradan o'qilgan ma'lumotlar MDR registriga joylanadi, undan esa boshqa registrlarga jo'natilishi mumkin bo'ladi.

MDR registrining aloqalari shinalari 16.3-rasmda keltirilgan.



16.3-rasm. MDR registrining ulanishlari va boshqaruv signallari

Unda 4 ta signal mavjud:  $MDR_{in}$ ,  $MDR_{out}$  – ichki shinaga bog'lanish,  $MDR_{inE}$  va  $MDR_{outE}$  – tashqi shinaga ulanish.

O'qish va yozish amallari jarayonida markaziy protsessor ning ichki taktlash jarayoni shina orqali aloqa o'rnatilgan qurilmaga moslanishi kerak. Bitta ichki ma'lumot yuborishni markaziy protsessor bir taktida bajaradi. Bunda qabul qiluvchining imkoniyatlari hisobga olinmaydi. Ma'lumki, zamonaviy markaziy protsessorlar xotiradan ma'lumotlarni bir taktida o'qishni ta'minlovchi kesh xotiraga ega bo'ladi. Agar markaziy protsessor ga kerakli ma'lumotlar keshda bo'lmasa, qidiruv asosiy xotiraga, unda ham bo'lmasa, qattiq diskka qaratiladi. Shunday xolatda bir necha taktida ma'lumot o'qiladi. Agar o'qish/yozish amali kiritish/chiqarish qurilmasining asosiy xotirada akslangan registri adresiga yuborilsa, bunday registrlar keshlanmagan bo'lganligi sababidan bunday murojaatga bir necha takt talab qilinadi.

Biror qurilma bilan aloqaga kirishishdan oldin markaziy protsessor avval jo'natilgan o'qish komandasining yakunlanishini kutadi. Bu ishni MFC (Memory Function Complete) funksiyasi aniqlaydi. Agar ko'rsatilgan adresdagi ma'lumot o'qilgan va shinaga joylangan bo'lsa, adreslanayotgan qurilma MFC ning qiymatini 1 ga tenglaydi.

Soddaroq bo'lishi uchun, MAR registrining chiqishi hamisha ochiq deb faraz qilamiz. Demak, unung tarkibi shinasining adres liniyasi uchun hamisha ochiq. MAR ga yangi adres yuklanganda keyingi taktning boshlanishida xotira shinasida paydo bo'ladi.

Boshqaruvchi o'qish signali registrning yuklanishi bilan bir vaqtda ishga tushadi. Bu signalga javob qilib shina interfeysi shinaga MR

o'qish komandasini joylaydi. Xotiradan o'qilgan ma'lumotlar takt oxirida MDR ga joylanadi. Keyingi taktida ma'lumotlarni R2 registriga yuborish uchun MDR<sub>out</sub> liniyasi faollashtiriladi. Bularning hammasi xotiradan ma'lumotlarni o'qish amali 3 taktida bajarilganini bildiradi.

### 16.3. Xotiraga so'zni yozish

Xotira adresiga so'zni yozish jarayoni yuqorida ko'rilgan so'zni o'qish jarayoni kabi bajariladi. Adres MAR ga yuklanadi, keyin yoziladigan ma'lumotlar MDR ga uzatiladi va yozish komandasi beriladi. So'zni o'qishda bo'lgan vaziyat kabi yozishni boshqarish signali xotira shinasining interfeys sxemasiga Write komandasini shinaga joylash buyrug'ini beradi. markaziy protsessor xotiraga yozish jarayoni yakunlanganini bildiruvchi MFC signali kelguncha kutib turadi.

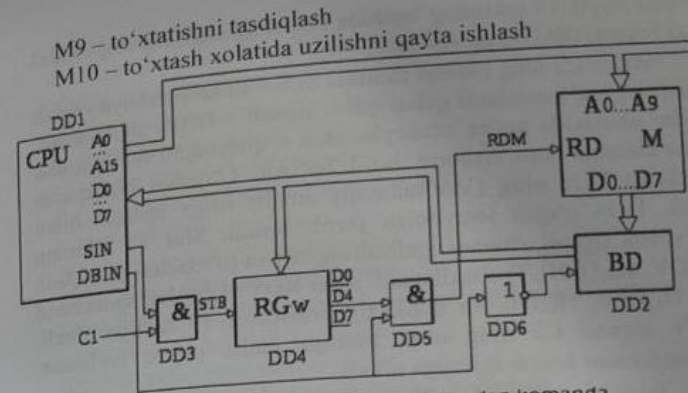
### 16.4. Mashina takti. Komanda sikli va mashina takti sikli

Komanda sikli – markaziy protsessor bitta komandani bajarishi uchun sarflanadigan vaqt. Komanda sikli o'z navbatida mashina sikllaridan tashkil topadi. Mashina sikli – markaziy protsessor ning ma'lum bir guruh harakatlarni bajarish jarayonidir. Komanda sikliga 3 tadan 5 tagacha mashina sikllari kiradi va ularning bittasi albatta M1 bo'lishi shart. Har bir mashina sikli qator mashina taktlaridan iborat. Mashina takti – sinxron impulslarning seriyasi vaqti. Bitta mashina siklida 3 tadan 5 tagacha mashina takti bo'ladi.

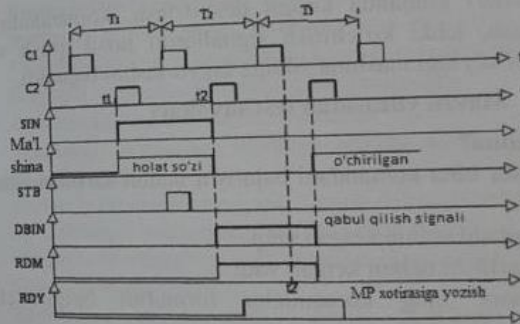
Mashina sikllarining turlari:

M1 – komandaning birinchi baytini olish. D7D6D5D4D3D2D1D0 – so'z (holat so'zi yoki holat bayti). markaziy protsessor mashina sikli yakunlanganligini mos holat baytiga yuboradi. Holat bayti registrga yoki sistema kontrolleriga yoziladi. Holat bayti asosida sistemada kerakli boshqaruv signallari shakllanadi. Bunda holat baytining ba'zi bitlari ishlatiladi. Masalan M1 uchun holat bayti 10100010 kabi bo'lishi mumkin.

- M2 – xotiradan o'qish
- M3 – xotiraga yozish
- M4 – stekdan o'qish
- M5 – stekka yozish
- M6 – tashqi qurilmaga ma'lumot uzatish
- M7 – tashqi qurilmadan ma'lumot qabul qilish
- M8 – uzilish (tasdiqlash yoki hosil qilish)



16.4-ram. Markaziy protsessorning xotiradan komanda (yoki ma'lumot) o'qish sxemasi



16.5-rasm. Markaziy protsessorni ishlashining vaqt diagrammasi

Tushuntirish:

Jarayonda adres shinasiga markaziy protsessor o'qilishi kerak bo'lgan (ma'lumot yoki komandaning) xotira yacheykasining adresini joylaydi.

- T1 taktida adresni uzatishdan tashqari C2 impulsining oldingi fronti bo'yicha sinxronlash signali hosil bo'ladi. Ma'lumotlar shinasiga bajarilayotgan mashina sikli haqidagi so'z (bayt) kelib tushadi.
- Qabul qiluvchi signal (DBIN) bu taktida yo'q (0 ga teng). U shina o'zgartirgich ishini boshqarib turadi

- Holat bayti T2 taktining boshida (C1 ning ikkinchi impulsida) holat so'zi registri (RGW) ga yoziladi.

- T2 taktida C2 ning oldingi frontida SIN – sinxronizatsiya signali yakunlanadi va ma'lumotlarni qabul qilish signali – DBIN shakllanadi. Ma'lumotlar shinasida xotira yacheykasidan o'qilayotgan ma'lumotlar yoki komandalarga mos axborot hosil bo'ladi. O'qishni boshqarish signali (RDM) DD5 ning (VA mantiqiy amali) ishga tushishi bilan shakllanadi. DD4 o'qish jarayoniga javob beradi. Shu T2 taktning o'zida tayyorlik signali, shinani egallash signali va to'xtashni tasdiqlash signali (RDY, HLD, HLT) shakllanadi. Agar RDY=1 bo'lsa, shina band emas, va HLD=0, HLT=0, va markaziy protsessor T3 taktiga o'tadi. Agar RDY signali C2 ning orqa frontiga qadar paydo bo'lmasa, markaziy protsessor kutish rejimiga o'tadi.

- C1 impulsi yakunida T3 taktida ma'lumotlar shinasidan markaziy protsessor buferida yozish bajariladi. C2 ning oldingi frontida DBIN va RDM signallari 0 ga tenglanadi.

- T4 va T5 taktlari komanda kodini deshifrlash, komandadagi baytlar sonini aniqlash, ichki ko'chirish signallarini hosil qilish va markaziy protsessor ni keyingi mashina sikliga tayyorlashni bajaradi.

#### Mavzu yuzasidan test savollari

##### Komanda sikli nima?

markaziy protsessor bitta komandani bajarishi uchun sarflanadigan vaqt

Komandaning o'qilishi uchun ketgan vaqt

Komandaning bajarilishi uchun ketgan vaqt

markaziy protsessor ning komandalar formatini belgilovchi ko'rsatkich

##### Komanda sikliga ..... mashina sikllari kiradi

3 tadan 5 tagacha

8 tagacha

Komanda formatiga teng miqdorida

Adres baytiga teng miqdorda

**Barcha turdagi kompyuterlarga mos keladigan markaziy protsessor turi bu –**

Maxsus markaziy protsessor lar

Umummo'ljallangan mikroprotsessorlar (universal markaziy protsessor lar)

Bir kristalli markaziy protsessor lar

Barcha javob to'g'ri  
**CISC tushunchasiga ta'rif bering**

To'liq komandalar to'plami

Cheklangan komandalar to'plami

Uzun so'zli komandalar to'plami

CISC-arxitektura hisoblanmaydi

**Buyruqlar tizimi arxitekturasi nima?**

Yangi tipdagi arxitektura

Bir qancha arxitekturalarning to'plami

Buyruqlarni shakllantirish formati

Kompilyator uzatayotgan ma'lumotlarning darajasi tushuniladi

**Superskalyar arxitektura nima?**

Hisoblashning juda yuqori tezligi

\*Bir taktida bittadan ko'p komandani ishlanishi

Tartiblanmagan arxitektura

Ketma-ket keluvchi potoklar

**Boshqarishdagi nizolar qanday vaziyatlarda yuzaga keladi?**

markaziy protsessor ga ikkita qurilma bir vaqtda murojaat qilganda

\*Shartli o'tish komandalarini konveyerlashda yuzaga keladi

Registrlil xotiraga murojaat qila olmaganda

Axborot almashishdagi uzilishlarda

**Markaziy protsessor ichidagi barcha ko'chirish amallari qanday vaqtda bajariladi?**

markaziy protsessor ning takt signali deb nomlanuvchi vaqt davomida

Ko'chirishnii boshqarish signallari takti boshida

O'qish komandasining yakunlanishini kutmasdan

Shina bo'shagan vaqtda

**Xotiradan so'zni o'qish uchun markaziy protsessor qanday ma'lumotlarni uzatadi?**

Registr adresini va 0 ni

So'z adresini va o'qish komandasini

Registr adresini va 1 ni

So'zning o'zini

**Ma'lumotlar trakti ni tashkil etuvchilarini ko'rsating**

Faqat registrlar

Registrlar va shina

Registrlar, arifmetik-mantiqiy qurilma va markaziy protsessor ning ichki shinasi birgalikda

arifmetik-mantiqiy qurilma va markaziy protsessor

## 17. Protsessorlar turlari: Matritsali va vektorli protsessorlar. Multiprotsessorlar

### 17.1. Matritsali protsessorlar

Konveyerlardan va superskalyar arxitekturadan foydalanish ish tezligini 5-10 barobar oshiradi. Unumdorlikni 50, 100 va undan ko'p oshirish uchun esa ko'p protsessorli kompyuter ishlab chiqish talab etiladi.

Parallel kompyuter tizimlarida bitta masalaning turli qismlarini hal qilayotgan protsessorlar axborot almashish uchun qandaydir tartibda hamkorlik qilishlari kerak. Aynan qanday? Buning uchun ikkita yondashuv taklif etilgan: multiprotsessorlar va multikompyuterlar. Bu ikki yondashuvni farqlovchi asosiy ko'rsatkich – umumiy xotiraning bor va/yoki yo'qligidir. Bu farq bu tizimlarning konstruksiyasida ham, qurilishida ham, dasturlanishida ham, razmerlarida ham va narxida ham ta'sirini o'tkazadi.

### 17.2. Matritsali protsessorlar

Fizik va texnik masalalarning deyarli barchasi massivlar yoki boshqa tartiblangan strukturalarni ishlatilishini nazarda tutadi. Ko'pincha bir xil bo'lgan hisoblashlar turli xil ma'lumotlar to'plamlari ustida bir vaqtda bajarilishi mumkin. Bunday hisoblashlarni bajarish uchun tartiblanganlik va strukturalanganlik darajasi komandalarni parallel qayta ishlanishi va hisoblashlarni tezlatishda juda qulaydir.

Matritsali protsessor (array processor) bir xil turdagi ko'plab protsessorlardan tashkil topgan bo'lib, bir xil amallar ketma-ketligini turli ma'lumotlar bilan bajaradi. Bunday birinchi protsessor ILLIAC IV. Uni yaratishda dastlab har biri 8x8 (protsessor+xotira) matritsali 4 ta blokli sxema qurish rajalashtirilgan edi. Har bir blok uchun alohida nazorat bloki bo'lishi talab etiladi. U yuborgan komanda barcha protsessorlarda bir vaqtda bajariladi, shu bilan birga har bir protsessor o'zining xotirasidan foydalanadi (ma'lumotlarni yuklash initsializatsiya jarayonida bajariladi). Bunday, fon-Neyman prinsipidan farqli bo'lgan yechimni SIMD arxitekturasi deb nomlangan. Nazariya va amaliyot bir-biriga mos kelmagan. Chunki bunday tizimni qurishning bahosi astronomik summa bo'lib ketgan. Natijada faqat bitta blok qurilgan, u Qo'zg'aluvchi vergulli sonlar bilan 50mln amal/sek hisoblash bajargan. Agar 4 ta blokni ham qurib, ishga tushirganda 1 mlrd amal/sek tezlikka

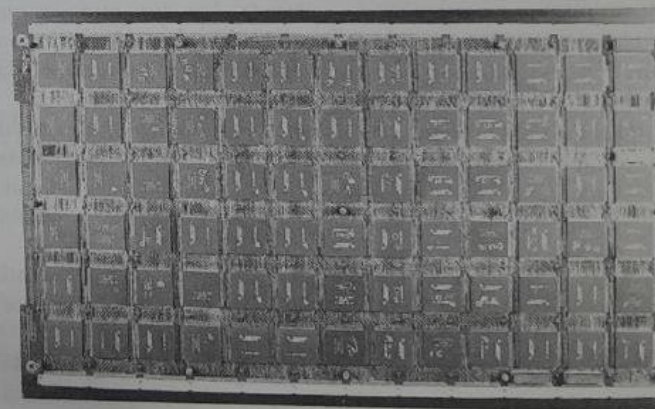
erishib, butun dunyodagi kopmyuterlarning barchasidan unumdorroq bo'lar edi.

### 17.3. Vektorli protsessorlar

Dasturiy nigohda qaraganda vektorli protsessor matritsali protsessorga o'xshab ketadi. Matritsali protsessor kabi u ham hisoblashda yuqori unumdorlikka ega, lekin undan farqli ravishda barcha qo'shish amallari konveyerli strukturaga ega bo'lgan qo'shish blokida bajariladi. Bunday asosda CRAY-1 kompyuteri ishlab chiqilgan.

Har ikkala tipdagi protsessorlar bir xil amalni bajaradi, misol uchun ikkita vektor elementlari juftlarini qo'shadi. Lekin, matritsali protsessor qo'shuvchi elementlar soni massiv elementlari soniga teng bo'lsa, vektorli protsessor shartli registrlar to'plamidan iborat vektorli registrga ega bo'ladi. Bu registrlar xotiraga bitta komanda bilan joylashadi. Qo'shish komandasi bu vektorlar elementlarini konveyer strukturali qo'shish qurilmasiga yuklab qo'shadi. Natijada qo'shish qurilmasidan boshqa vektor chiqadi va vektorli registrga yoziladi yoki boshqa amal bajarish uchun operand sifatida ishlatiladi.

Hozirgi kunlarda matritsali protsessorlar ishlab chiqilmaydi, lekin ular asoslangan tamoyil hali ham ishlatiladi. Bunday g'oya Pentium IV ning MMX va SSE to'plamlarida mumltimediya dasturlarini tezkorligini oshirishda ishlatiladi.



17.1-rasm. CRAY vektorli kompyuterining protsessor platasi

**Multiprotsessorlar.** Multiprotsessor (ing. *multiprocessor, multiprocessing*) – ko'pprotsessorli kompyuter tizimining bir sinfi bo'lib, bir nechta protsessorlar va ular uchun umumiy bo'lgan yagona adres muhitiga ega bo'lgan tizim. Flinn klassifikatsiyasi bo'yicha multiprotsessorlar MIMD sinfiga tegishli deb hisoblanadi. Multiprotsessorda bir jadvali bitta OT (operatsion tizim) ishlatiladi.

#### 17.4. Klassifikatsiya

Multiprotsessorli tizimdagi protsessorlarning *tutgan o'rniga qarab: simmetrik multiprotsessorlar (SMP)* – bunda barcha protsessorlarning darajasi teng va xotiraga va qurilmalarga bir xil murojaat qila oladi, va *asimmetrik multiprotsessorlar, (Amarkaziy protsessor)* bunda protsessorlarning darajalari turli xil, xotiraga va qurilmalarga turli xil murojaat qiladi, *turlarga bo'linadi.* Amarkaziy protsessor texnologiyasi 1960-yillarda SMP texnologiyasi qurilguncha vaqtinchalik ishlatilgan.

Xotirani adreslash usuli bo'yicha bir necha xil multiprotsessorlar farqlanadi, mas: UMA (Uniform Memory Access), NUMA (Non Uniform Memory Access), COMA (Cache Only Memory Access). Bundan tashqari multiprotsessorlar gomogen tipli bo'lishi mumkin (barcha protsessorlar bir xil) yoki geterogen (tizimdagi protsessorlar turli xil) bo'lishi mumkin.

#### 17.5. Dasturlashtirish

Multiprotsessorlarni dasturlashtirishda ikki xil modeldan foydalanish mumkin: *ko'ppotoklilik* – bunda har bir protsessorga bajarish potoki yuklanadi va axborot almashishda umumiy xotiradan foydalanishadi, *message passing*, bunda har bir protsessorida alohida jarayon bajariladi va o'zaro xabar uzatish orqali axborot almashadilar.

Ko'ppotokli dasturlashtirish ochiq, yopiq yoki avtomatik bajarilishi mumkin. Ochiq dasturlashtirish – tizimli API yordamida kompilyatsiyalanadigan tillar, mas, C/C++ bilan, interpretatsiyalanadigan Java va C# kabi tillarning konstruksiyasida bajariladi.

Yopiq dasturlashtirish – deklarativ, kompilyator direktivalari yordamida bajariladi. Avtomatik dasturlashtirish esa kompilyator tomonidan mustaqil bajariladi.

Multiprotsessorli mashinalar (SMP/UMA) 1990-yillarning boshida *minisuperkompyuter* sifatida, keyinchalik ma'lumotlarni qayta ishlash serveri sifatida ishlab chiqilgan. x86 arxitekturali Intel 80386, 80486

protsessorlari bazasidagi multiprotsessorlar (CompaqSystemPro) keng tarqalgan. Multiprotsessorli x86-serverlarga talab ortishi Intel tomonidan MultiProtsessor Specification standartini ishlab chiqilishiga olib keldi va uzoq yillar davomida qo'llanildi.

17.1-jadval.  
SMP/UMA-tizimiga misollar

Система	Процессор	Максимум процессоров
SGI Power Challenge	MIPS R10000	36
DEC AlphaServer 8400	Alpha 21164	14
Sun E6000	UltraSPARC II	30
Sun E10000	UltraSPARC II	64
HP/Convex Exemplar	PA-8000	16
Cray T90	Cray Vector	32

17.2-jadval.

ccNUMA-tipli multiprotsessorlar

Система	Процессор	Процессоров на узел	Максимум процессоров
Data General AV-20000	Pentium	4	32
Sequent NUMA-Q 2000	Pentium	4	32
SGI Origin 2000	MIPS R10000	2	128
HP Exemplar X-Class	PA-8000	16	256
HP 9000 V-Class			

Hisoblash texnikasida elementlar hajmi kichiklashishi CMP (chip-level multiprocessing) – bir korpusda yoki bir kristallda bir necha protsessorlarni joylashtirib, *multi yadroli protsessorlarni* ishlab chiqishga yo'l ochdi. Bu takt chastotasi yoki konveyerlar sonini oshirish kabi standart yo'ldan emas, balki chipdagi yadrolar sonini ko'paytirish orqali unumdorlikni oshirish yo'lini ko'rsatdi. Bunday ko'p yadroli

mikroprotessorlar xotiradan unumliroq foydalanadi, ko'ppotokli ilovalar bilan ishlaydi, kam energiya sarflaydi.

2005-yillardan boshlab barcha mikroprotessorlar ko'p yadroli variantda ishlab chiqilishi yo'lga qo'yilgan, bizning holatimizda kichik multiprotessorlarda yoki, boshqacha aytganda xotiraga UMA kirishli SMP-mashinalarida. Bu yo'nalishining rivojlanishi ko'p yadroli multiprotessorlarni ishlab chiqilishga olib keldi, bunday multiprotessorlar bir necha protessorlardan ular esa o'z navbatida bir necha yadrodan iborat bo'ladi. Ko'p tarqalgan variantlaridan 8 yadroli Intel Xeon, Oracle Sparc va boshqalar.

#### 17.6. Intel Pentium protessorlari. Celeron protessorlari. AMD, UltraSPARC, Cyrix protessorlari. Superkompyuter protessorlari

**Intel firmasi protessorlari.** Intel firmasining quyidagi Pentium protessorlarini ko'rib chiqamiz.

**CPU Pentium** (Socket 370 protessor yadrosi Tualatin), takt chastotasi 1133-1400 MHz, texnologiyasi 0,13 mkm, shina chastotasi 100-133 MHz, KESh L2-512 Kbayt, kengligi MMX, SSE. Bu protessorlar 0,13 mkm – texnologiya bilan ishlab chiqilgan bo'lsa ham va ikki protessorli (dual) shaklni ta'minlay olsa ham eskirib qolgan protessorlar hisoblanadilar yaqin orada ishlab chiqarilishi to'xtatiladi.

- **CPU Celeron** (Socket 370, protessor yadrosi Tualatin), takt chastotasi 1200-1300 MHz, texnologiyasi 0,13 mkm, shina chastotasi 100 MHz, KESh L2-250 k.bayt, kengligi MMX, OSSE. Shina chastotasi –100 MHz (133 MHz emas). Bu protessorlar ham ishlab chiqarilishdan olinadi.

- **CPU Pentium 4** (Socket 478, protessor yadrosi Willamette), takt chastotasi – 1.5-2.0 GHz, shina chastotasi – 400 MHz, KESh L2-256 k.bayt, kengligi MMX, SSE, SSE2. Bu turdagi protessorlar yangi ishlash unumdorligi yuqori protessorlar orasida eng ko'p tarqalgani hisoblanadi. Uning kamchiliklari sifatida eski 0,18 mkm – texnologiyasini KESh L2 ning 256 Kbayt va ko'p protessorlikni ta'minlashi mavjud emasligini ko'rsatish mumkin. Uning ustunligi deb narx/mahsuldorlik=maximum ekanligini ko'rsatish mumkin. 1,5-1,7 GHz etib ishlanganlari professional darajadagi eng ko'p xarid qilinadigan protessorlar hisoblanadi.

- **CPU Pentium 4** (Socket 478, protessor yadrosi Northwood), takt chastotasi – 1.6-3.06 GHz, shina chastotasi – 400 MHz (chastotasi 3.06

GHz bo'lgan protessorning shina chastotasi –533 MHz), KESh L2-512 Kbayt, kengligi MMX, SSE, SSE2. Bu protessor hozircha Intel ning eng yuqori chastotali protessorlaridir (AMD haqida ham shunday deyish mumkin), ular birinchi bo'lib 3 GHz chegarasidan o'tgan. Ikkinchidan CPU Pentium 4 – 3,06 GHz dan boshlab, Intel korporatsiyasi ish joyida foydalanilgan sistemalariga mo'ljallangan protessorlar ham Hyper – Threading texnologiyalarini ishlatishga qaror qiladi. Bu texnologiya ilgari ham ishlashi mumkin bo'lgan, lekin Intel Pentium 4 protessorlari ilgari modellarida bu vazifani sun'iy ravishda o'chirib qo'ygan.

- **CPU Celeron** (Socket 478) protessor takt chastotali 1,5-1,8 GHz. Shina chastotasi 400 MHz, kesh 42-128 kbayt, kengayishi MMX, SSE, SSE2.

- **CPU Celeron** (Socket 478, protessor yadrosi Northwood) takt chastotasi 2,1-2,2 GHz, shina chastotasi 400 MHz, kesh 42-128 kbayt, kengayishi MMX, SSE, SSE2. Bu ish joyi sistemalari uchun "kichik" klassdagi eng yuqori chastotali protessorlardir. Intel uni 2002 yildan ishlab chiqara boshladi.

- **Celeron** (Socket 370, protessor yadrosi Coppermine) chastotasi 500-1000 MHz, shina chastotasi 66(100) MHz, ket 42-128 kbayt, kengayishi MMX, SSE. Albatta ularga talabning kattaligi protessorning arzonligi sabab.

Hozirgi kunda eng mahsuldor protessorlar Pentium 4 3,06 GHz va undan keyin ishlab chiqarilganlari hisoblanadi.

Ammo protessorning narxi hatto birgina protessor yadrosiga qarab katta farq qilishi mumkin. Bu shundan kelib chiqadiki, protessorni ishlab chiqarish texnologiyasini takomillashtirilganda yuqori chastotalarda barqaror ishlay oladigan protessorlar foizi unchalik katta emas, ana shu sababli yuqori chastotali SPU narxlarida katta farq kelib chiqadi. Protessorlarni tanlashda yuqori chastotaligi bo'lmasa ham, lekin zamonaviy protessor yadrosiga ega protessorni tanlash lozim. Unumdorlik farqini kompyuterning boshqa qismlariga qo'shimcha bilan o'rnini to'ldirish mumkin (xotirasi, NDD, MV va shu kabilar) bundan tashqari bugungi kunda kompyuterlarning mahsuldorligi umuman olganda faqatgina SPU ga unchalik kuchli bog'liq emas.



### 17.7. Hyper-Threading texnologiyasi yoki bir CPU doirasida "ikki protsessorlik"

Hyper-Threading texnologiyali yangi Intel protsessorlarini siz qo'llay olishingiz uchun quyidagi shartlarni bajarishingiz kerak:

- Tizimli platalarda shu texnologiyani ta'minlovchi va 533 MHz-850E, 845RE, 845GE, 845GV, 845E va 845G Intel mikrosxemalari to'plami qo'llanilgan bo'lishi kerak. Boshqa ishlab chiqaruvchilarda ham yaqin oradabu texnologiyani ta'minlovchi tizimli mantiqlar to'plami paydo bo'lsa kerak. Plata yangilangan BIOS va kerakli drayverlarga ega bo'lishi kerak.

- Tizim Hyper Threading dan foydalanish uchun moslashtirilgan operatsion tizim bilan boshqarilishi kerak. Bu ish MS Windows XP va Limax 2,4x dasturida to'laligicha amalga oshirilgan.

### 17.8. AMD kompaniyasi protsessorlari

AMD kompaniyasining oddiy foydalanuvchi uchun protsessorlari Intel protsessorlariga o'xshab ikki turga bo'linadi — qimmatroq va unumlirog'i (Athlon) va arzonrog'i — (Duron).

AMD Athlon protsessorlari CPU Intel Celeron narxlariga kategoriyasiga kirib, unumdorligi esa Intel Pentium 4 bilan teng.

Intel Pentium protsessorlari bilan haqiqatda raqobatlasha oladigan birinchi DES firmasining RISC protsessori — Alfa 21064 bazasida ishlab chiqilgan 7-avlod AMD Athlon protsessori 1999 yilda paydo bo'ldi. 2000 yildan boshlab, Athlon protsessorlarini 0,18mkm — texnologiyasi bilan Thunderbird protsessor yadrosi, 650-950 MHz takt chastotali ishlab chiqara boshladilar, misli ulanishni texnologik jarayonni qo'llash bilan esa 1000-1400 MHz taktl chastotasiga erishildi. AMD Athlon protsessorlari Intel Pentium protsessorlariga nisbatan keyinroq esa Pentium 4 ham katta kamchilikka ega bo'lgan — ancha ko'proq issiqlik chiqarishi va kristall qizishini aniq nazorat qilib turish uchun protsessorga o'ratiladigan termiodning yo'qligidadir. Bu holat kuler to'xtaganida protsessor kuyib qolishiga olib kelishi mumkin. Intel protsessorlari qizib ketadigan bo'lsa, tizim faqatgina to'xtab qoladi.

AMD ning Palomino yadrosiga asoslangan protsessorlarining keyingi turlarida bu kamchiliklar ancha yo'qotildi, yadrosi tuzilishini optimallashtirish hisobiga esa issiqlik ajralishi 20% ga kamaytirildi, protsessor uyusiga termiod kiritildi. AMDning Palomino protsessori yadrosiga eng yangi protsessorlari Athlon XR (Extreme Performance) deb, ko'p protsessorlik (ikki) konfiguratsiyalar uchun esa- Athlon MR

deb atala boshlandi. Birinchi ishlab chiqarilgan Athlon XR protsessorlari Athlon MR protsessorlaridan deyarli hech narsa bilan farq qilmaydi va ikki protsessorlikni ta'minlagan, ammo Athlon XR 1600+ turidan boshlab bu imkoniyat sun'iy ravishda o'chirib qo'yilgan. Athlon XR-2000+ bu degani 2 GHz takt chastotaga ega Pentium 4 protsessoriga qaraganda bu protsessor ancha yaxshi ishlashini anglatadi. Aslida esa Athlon XR-2000+ protsessorining takt chastotasi 1666 MHzni tashkil qiladi. Palomino protsessor yadrosiga ega modeli- Athlon XR AMD-K7-1600+ protsessor shinasini 266 MHz (aslida esa 133 MHz, lekin ma'lumotlar uzatish signalning ikki fronti bo'yicha olib boriladi) va kesh xotirasi L1+L2=384 Kbayt. Agarda masalan Intel protsessorlarida kesh xotirasi birinchi bosqichi L1(16+16 Kbayt) faqatgina protsessorni o'ziga xizmat qilsa, AMD protsessorlarida esa kesh xotira L1(64+64 Kbayt) va L2 qo'shiladi, samarali ravishda 384 Kbayt ishlaydi deb hisoblanadi.

**Qaysi protsessorni tanlash kerak?** Intel platformadami yoki AMD. Albatta boshlang'ich darajadagi tizimlar uchun Intel Celeron protsessorlari juda mos hisoblanadi. Deyarli har qanday ofisda va uyda qo'llash uchun CPU Celeron Socket 378 (Tualatin) ham va CPU Celeron Socket 478 (Willamette yoki Northwood) ham to'g'ri keladi. To'la imkoniyatga ega o'yin uchun kompyuterga protsessorning ishlash darajasi unchalik muhim emas, balki videokartasining ishlash darajasi ko'proq muhimdir, shuning uchun Celeron protsessorini tavsiya qilish mumkin. Lekin takt chastotasi 1 GHz dan kam bo'lmasligi kerak.

Professional qo'llanilishi uchun mo'ljallangan sistemalar uchun faqatgina CPU Pentium 4 ni olish kerak, u Northwood yadrosida bo'lgani yaxshi. Birinchidan KESH — xotirasi L2 512 k.bayt (Willamette yadrosida — 256 k.bayt) xususan xuddi ana shu ko'rsatkich bo'yicha R4 Celeron dan ustun turadi. Ikkinchidan, R4 Northwood protsessori 0,13 mkm — texnologiyasi bo'yicha ishlab chiqariladi va aslida kam issiqlik chiqarishi va kam qizishi kerak. AMD Athlon XP protsessorlari haqida gapiradigan bo'lsak, bu protsessorlar ko'plab testlar va kuzatishlar bo'yicha hisoblash kuchi, imkoniyati bo'yicha Pentium 4 dan ustun. Ammo Intel protsessorlari bazasidagi sistemalar har doim ancha barqaror ishlaydi, umumiy sistemada qo'shimcha qismlar moslashuvida kamroq injiqliklar qiladi. Ammo Sizning oldingizda Celeron va Athlon XP o'rtasida tanlash masalasi turgan bo'lsa, Athlon XP ni tanlaganingiz yaxshi.

Agar o'z kompyutingizni ikki protsessorli platformada bo'lishini xohlasangiz Pentium Tualatin ikki protsessorli bazalisini o'rnatish tavsiya etiladi.

#### Mavzu yuzasidan testlar

**CPU Pentium 4 ning 2-darajali kesh xotirasi hajmi qancha?**

256 Kbayt

512 Kbayt

1024 Kbayt

128 Kbayt

**CPU Pentium 4 ning takt chastotasini ko'rsating**

1,0-1,5 GHz

1,5-1,6 Ghz

0,5-1,0 Ghz

1.6-3.06 GHz

**CPU Pentium 4 ning shina chastotasi**

400 MHz

500 MHz

600 MHz

200 MHz

**CPU Celeron ning takt chastotasi**

1,0-1,5 GHz

1,5-1,6 Ghz

0,5-1,0 Ghz

2,1-2,2 GHz

**CPU Celeron ning shina chastotasi**

600 MHz

400 MHz

500 MHz

200 MHz

**Matritsali protsessor (array processor) –**

Bir xil turdagi ko'plab protsessorlardan tashkil topgan

Har xil turdagi ko'plab protsessorlardan tashkil topgan

Bir xil turdagi ko'plab xotira bloklaridan tashkil topgan

Har xil turdagi ko'plab xotira bloklaridan tashkil topgan

**Multiprotsessor bu....**

Tezkor kompyuter

Bir nechta protsessorlar va ular uchun umumiy bo'lgan yagona adres muhitiga ega bo'lgan tizim

Protsessorlarning unumdorlik ko'rsatkichi  
Hajmi kichik protsessor

**Multiprotsessorlarni dasturlashtirish modellari –**  
Kompilyatsiya va interpretatsiya

Kamxatolilik prinsipi  
Ko'ppotoklilik va message passing

Sikllarni optimallashtirish uslubi  
**Ko'ppotokli dasturlashtirish turlari:**

Avtomatlashtirilgan, mexanik, aralash  
Yuqori darajadagi dasturlash tillariga asoslangan

Mashina tilida dasturlashga asoslangan  
Ochiq, yopiq, avtomatik

**ILLIAC IV protsessori qanday yondashuv asosida qurilgan?**

Matritsali protsessor

Multiprotsessor

Vektorli protsessor

Multikompyuter

## 18. Tashqi qurilmalarning turlari va vazifalari. Tashqi xotira qurilmalari. Tarmoq adapteri, strimmer

### 18.1. Klaviatura

Insonning kompyuter bilan bevosita aloqasini ta'minlovchi qurilma. Klaviatura ikki xil bo'ladi. 1-turdagi klaviatura plataga montaj qilingan mexanik kalitlar massividan tashkil topadi. Ustunlar va satrlar shaklida tashkil qilingan bo'lib, platadagi mikrokontrollerlar bilan o'zaro bog'langan. Biror klavish bosilganda kalit yopiladi va kontroller qaysi ustun va satrdagi klavish bosilganini qayd etadi, ASCII kodini hosil qiladi va markaziy protsessor ga yuboradi.

2-turdagi klaviatura tekis 3 qatlamli strukturaga ega. Yuqorigi qatlam plastik materialdan yasalgan bo'lib, bir tomonida klavishlar chizilgan, orqa tomonida o'tkazuvchi birikmalar surilgan. O'rta qatlam rezinadan, klavishlar to'g'risida teshikchalar ochilgan. Quyi qatlam metallardan, klavishlar to'g'risida bo'rtmalar mavjud. Foydalanuchi yuqori qatlamdagi klavishga tegganda, uning orqa tarafi quyi qatlamdagi metall bo'rtmaga tegadi, kalit yopiladi va 1-turdagi klaviatura kabi ish davom etadi. Bunday klaviaturalar arzon va ishonchlidir.

### 18.2. Sichqoncha

1968-yilda o'lab topilgan. Bungacha faqat klaviatura ma'lumot kiritishning asosiy qurilmasi sanalgan. Sichqinchaning paydo bo'lishi kompyuterda grafik lementlar qo'llanilish davrini boshlab berdi.

Klassik sichqoncha foydalanuvchi qo'lga moslashgan, kichik razmerli, tekislikda harakatlanuvchi qurilma. Elektron sxema harakat yo'nalishini (X va Y o'qlari bo'yicha) va harakat masofasini aniqlab, bu haqdagi ma'lumotni shina orqali markaziy protsessor ga yuboradi. Mexanik sichqoncha kichik sharning harakatidan sensorlarni harakatga keltiradi, sensorlar esa ikkita hisoblagichni ishga tushirib, sichqonchaning harakat masofasini X va Y o'qlari bo'yicha aniqlaydi. Sichqonchaning knopkalaridan olingan xabar (o'ng tomon bosildi, chap tomon bosildi, o'ng tomon 2 marta, chap tomon ikki marta) sichqoncha mikrokontrolleriga tushadi, unda 3 baytli paketga kodlanadi va ketma-ket ulanish shinasini orqali markaziy protsessor ga uzatiladi.

Optik sichqonchada LED-svetodiod (*Light-Emitting Diode*) ishlatilib, yorug'likka o'ta sezgir elementlar harakat miqdorini o'lchab boradi.

1999-yilda *Microsoft* firmasi *IntelliMouse* deb nomlangan yangi turdagi sichqonchani taklif qildi. Bu sichqoncha har qanday yuzada bir xil ishlaydi. Unda yorug'lik sensori o'rniga raqamli kamera o'rnatilgan. U sichqoncha ostidagi kichik uchastkani tasvirini raqamli shaklga o'tkazib, masofani aniqlaydi. Kamera 1500 kadr/sekund tezlikda ishlaydi. Agar yuza notekis bo'lsa, kamera ketma-ket olingan tasvirlarni tahlil qilib, sichqonchaning bosib o'tgan yo'lini aniq hisoblaydi. Bir tasvirdan ikkichisigacha masofani o'lchash uchun *korellyatsiya* deb nomlangan, signallarni qayta ishlash texnologiyasidan foydalanadi. Soniyasiga 1500 ta tasvirni qayta ishlash uchun kamera protsessori 18 mli amal/sekund bajaradi.

### 18.3. Trekbol, joystick va sensorli panel

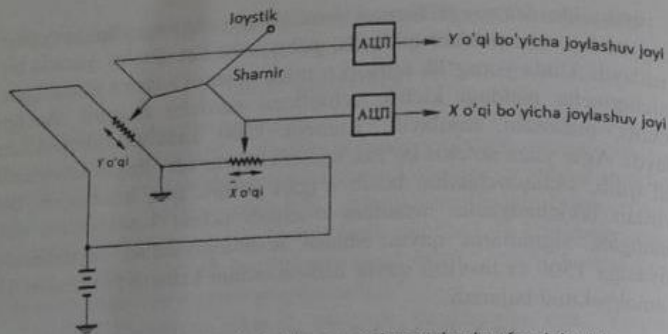
Sichqoncha foydalanuvchiga kursorni ekranda erkin harakatlantirish imkonini beradi. Shunga o'xshash vazifani boshqacha shaklda bajarishga bo'lgan talab boshqa qurilmalarni ishlab chiqilishiga olib keldi.

**Trekbol** mexanik sichqoncha prinsipida qurilgan. Klaviaturada sharik (*trekbol*) qo'yilgan bo'lib, uni aylantirish yordamida ko'rsatkich ekranda suriladi.



18.1-rasm. Trekbol

**Joystick** sharnirli kichik ruchka bo'lib, ixtiyoriy yo'nalishda egish mumkin. Egilishi natijasida X va Y o'qlaridagi siljishlar sodir bo'ladi va tegishli axborot markaziy protsessor ga uzatiladi.



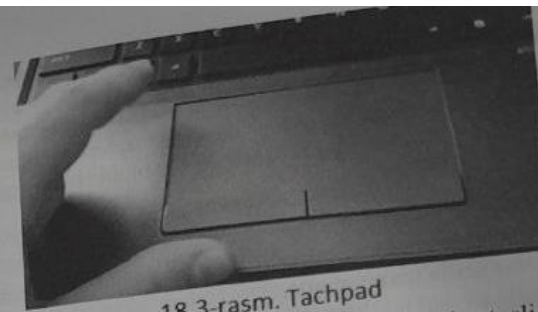
18.2-rasm. Joystik holatini aniqlashda potensiometrlardan foydalanish

Ruchkaning pozitsiyasi to'g'ri holat va burchak holatdagi datchiklar bilan seziladi (10.1 rasm). X va Y potensiometrlarining chiqish kuchlanishlari ikkita analog-raqamli o'zgartirgichga uzatiladi, uning chiqishida esa ko'rsatkichning yo'nalishi va masofasi hosil bo'ladi.

Joystiklar portativ kompyuterlarda va video o'yinlarda ishlatiladi. Portativ kompyuterlarda (noutbuklarda) klavishlar orasiga qo'yiladi va uni bitta barmoq bilan ishlatish mumkin. Uning afzalligi ishonchliligi va kichik joyni egallashidir. O'yinlarda ishlatiladigan joystik esa qo'lda o'rnatilgan darajada katta va unda ma'lum komandalarni bajarish uchun knopkalar mavjud.

#### 18.4. Sensorli panel (tachpad)

Foydalanuvchi sensorga tekkanda tegilgan joydagi elektr ko'rsatkichlar o'zgaradi. Panel bu nuqtaning joylashuvini aniqlaydi va markaziy protsessor ga xabar yuboradi. Panelda barmoqning uzliksiz yuritilishi bilan, bunday signallar ham uzliksiz jo'natilib turadi, shunga mos ravishda markaziy protsessor ekrandagi ko'rsatkichning pozitsiyasini o'zgartiradi. Uning ishonchliligi yuqori hisoblanadi, chunki sichqoncha yoki trekboldagi kabi harakatlanuvchi qismlarga ega emas. Asosan ko'chma kompyuterlarda (noutbuklarda) ishlatiladi.



18.3-rasm. Tachpad

Sensorli panelni (tachpadni) ishlab chiqishda turli materiallar ishlatiladi. Oxirgi loyihalardan biri ko'plab kichik optik tolalarda tashkil topganidir. U foydalanuvchining barmog'i tekkan nuqtaning nafaqat koordinatasini, balki bosilish kuchini ham aniqlaydi. Bu material dastlab kosmik robotlarni ishlab chiqishda qo'llanilgan bo'lsa, keyinchalik ko'plab oddiy foydalanuvchilar talablariga muvofiq ishlab chiqildi. Masalan, klaviatura fortepianosi ishlab chiqishda.

Sensorli panelni suyuq kristalli displey bilan birlashtirish natijasida sensorli ekran hosil bo'ladi (sensorli telefonlar, planshetlar).

#### 18.5. Skanerlar

Qog'ozdagi tasvirni raqamli shaklga o'tkazuvchi qurilma. Dastlabki skanerlarda qog'oz shisha silindrga mahkamlangan, u esa sensorlar orasida aylangan va tasvir hosil bo'lgan. Zamonaviy skanerlarda qog'oz shisha yuzaga qo'yiladi va yorug'lik bilan skanerlanadi. Qaytgan yorug'lik zaryadli aloqa uskunalari (ZAU=прибор с зарядовой связью-ПЗС) ning chiziqli matritsasida fokuslanadi. ZAU ga yorug'lik tushganda u bilan bog'langan kichik kondensator zaryadlanadi, uning zaryadi tushgan yorug'lik kuchiga teng bo'ladi. Analog-raqamli o'zgartirgich yordamida bu zaryad raqamli shaklga o'tkaziladi. Rangli skanerlarda qizil-ko'k-yashil rangli filtrlar bo'lib, ular yordamida ranglar farqlanadi va keyin alohida-alohida qayta ishlanadi. Yorug'likning qog'oz yuzasidan o'tishida sensorli matritsa signali ko'p marta qayta o'qiladi, shu shaklda tasvirning pikselli satrlari hosil qilinadi.

Skanerlash yakunlanib, axborot kompyuterga uzatiladi va xotirada piksellar massivi shaklida saqlanadi. Oq/qora skanerlaganda har bir nuqta bir bitga teng bo'ladi (0=qora, 1=oq). Rangli skanerlaganda esa

har bir nuqta ko'pi bilan 3 baytga teng bo'lib, har baytda rang va tiniqlik darajasi beriladi.

Skanerlangan matnni Word dagi matnga o'zgartirishda qora ranglar matn yozuvlarini bildiradi. Maxsus dasturlar qora ranglar ketma-ketligini kompyuter xotirasidagi harflash shabloni bilan taqqoslab, matnni o'qib olishni tashkil etadi.

### 18.6. Displeylar

Elektron-nurli displeylar eskirib qolgani sababli ular haqidagi ma'lumotlarni bu yerda qaramaymiz.

Undan tashqari suyuq kristalli, plazmali va elektroluminescent displeylar mavjud. Portativ kompyuterlarning (noutbuk) yaratilishida aynan shunday displeylar muhim rol bajardi.

**Suyuq kristalli displey** ikkita shaffof panel orasiga joylashgan yupqa qatlami suyuq kristall (kristallik xususiyatlariga ega bo'lgan suyuqlik = LCD-Liquid Crystall Display) dan iborat. Yuqori panelda shaffof elektrodlar joylashgan, orqa panel esa ko'zgu. Plastinalarga elektr kuchlanishi berib, diffuziyani o'zgartirib yoki yorug'likning qutblanishini o'zgartirib suyuq kristallning turli segmentlari faollashtiriladi. Natijada kristall segmentlari yorug'likni yoki akslantiradi yoki yutadi. Tasvir esa yorug'likning asosidan chiqib, suyuq kristall segmentlaridan o'tib, ko'zguda akslanishidan hosil bo'ladi.

LCD ikki xil bo'ladi. *Statik (yoki passiv-matritsali)* oddiy tuzilishda bo'lib, elektrodlar yuqori panel o'qida va unga perpendikulyar panel o'qida joylashadi. Segmentni yoritish uchun ustun va satrni bildiruvchi ikkita elektrodga kuchlanish beriladi. Hosil bo'lgan elektr maydonning ustun va satr kesishgan nuqtasi yorishadi. Bundan yuqori sifatli displey ishlab chiqishda ustun va satrning har bir kesishmasida elektrodlar joylashtiriladi (elektrodlar matritsasi hosil qilinadi). Tranzistorlar yupqa plyonkada bir panelni joylashadi. Bunday displeylar *TFT-displey (Thin-Film Transistor – yupqa plyonkali tranzistor)* deyiladi.

**Plazmali displey** ikkita shisha plastinali bo'lib, ularning orasi gaz (odatda neon gazi) bilan to'ldiriladi. Har bir plastinaga qator parallel plastinalar joylashtirilgan. Ikki plastina elektrodleri to'g'ri burchak ostida joylashtirilgan. Turli plastinalarda joylashgan ikkita elektrodga kuchlanish berilganda ularning kesishish joyidagi kichik nuqta yorishadi. Plazali displeylar juda tiniq tasvir hosil qiladi, lekin narxi baland.

**Elektroluminescent displeylarda** ikkita o'tkazuvchi panel orasida yupqa fosfor qatlami ishlatiladi. Panellarga berilgan elektr kuchlanish natijasida kerakli nuqatada fosforning yorishishi kuzatiladi va tasvir hosil bo'ladi.

### 18.7. Printerlar

Printerlar kompyuterning matnli yoki grafik shaklda chiquvchi ma'lumotlarni qog'oz shaklini chop etadi. (Lazerli) printerlar quyidagicha ishlaydi. Yuzasi musbat zaryadli fotoo'tkazuvchi material bilan qoplangan baraban lazer niriida skanerlanadi. Yoritilganda zaryad zaryadlangan baraban yuzasiga mayda pudra shaklidagi manfiy va qog'ozga o'tadi. Sahifa pechat qilingandan keyin baraban toner qodiqalaridan tozalanadi.

Oqimli (струйный) printerlarda turli rangdagi siyohlar kichik teshikchalardan qog'ozga otiladi. Masalan oqim-ko'pikli printerlarda chiqadigan siyoh kamerada qaynatiladi va ko'pik shakliga teshikda otilib chiqadi. Hamma siyohsi sarflagandan keyin, kamerada vakuum hosil bo'ladi va vakuum yangi porsiya siyohni kabul qiladi.

### 18.8. Grafik port

Grafik plata kompyuterga shina orqali ulanishi mumkin (masalan PCI). Lekin, ko'pincha sistema platada AGP (Accelerated Graphics Port – tezkor grafik port) deb nomlangan, grafik plata uchun maxsuslashtirilgan slot mavjud. Bu port 32 razryadli bo'lib, PCI ga nisbatan yuqori tezlikka ega. Variantlari: AGP1x, AGP2x, AGP4x, AGP8x. AGP 1x-asosiy standart, uning o'tkazuvchanligi 264 Mb/s. AGP8x da esa 2Gb/s.

### 18.9. Grafik shakllarni qayta ishlanishi.

Kompyuterda 3 o'lchamli obyektlar ko'plab kichkina ko'pburchaklardan (asosan uchburchaklardan) tashkil topgan yuzada hosil qilinadi. Obyektlarning *proyeksiya* va *perspektiva* sini aniqlashda tasvirning turli qismlarini ifodalovchi uchburchaklarning joylashuvini hisob-kitob qilish kerak. Keyin, turli murakkab algoritmlar bilan har bir uchburchakning rangi va soyasi aniqlanadi. Bunda yorug'lik manbaining qaysi tomondan tushib turganligi, uni turli yuzalarda qanday akslanish kabi ko'rsatkichlar alohida et'tiborga olinadi. Bu jarayonda muhim jihatlardan yana biri yuzalarni aniqlashtirishdir. Masalan g'isht devorning yuzasi yoki yog'ochning yuzasi kabi. Buning

uchun tekse (texel) lar ishlatiladi. Yuzaga tegishli uchburchaklar tekse bilan to'ldiriladi va o'lchamli ko'rinish hosil bo'ladi. Tasvirning ko'rinmaydigan qismlari kesib tashlash (*clipping*) bilan o'chiriladi. Tasvirning ranglari va tiniqligini qayta ishlash (*sampling*) deyiladi. Ana shu barcha jarayonlarning barchasini bajarib, uni ekranga foydalanuvchi ko'rishi uchun uzatish jarayoni esa vizualizatsiya (*rendering*) deyiladi.

Harakatli tasvirlarda bu hisob-kitoblar soniyasiga 20, 30, 40 marta bajariladi. Bu ko'rsatkich *kadr chastotasi* deyiladi. Yuqoridagi amallarni grafik plata bajara olishi uning T&L (*Transformations and Lighting* - o'zgartirish va yoritish) koeffitsienti bilan o'lchanadi. T&L koeffitsienti bir sekundda videokarta nechta uchburchakni proyeksiyalash, akslanish, yoritish, *sampling* qilishini belgilovchi ko'rsatkichdir. Hozirgi holatda o'rtacha T&L=10...30 mln uchburchak/sekundga teng.

Quyida RADEON VE (ATI Group firmasi) va GeForce 2MX (nVidia firmasi) ning grafik platalari xarakteristikalarini keltirilgan

18.1-jadval.  
RADEON VE va GeForce2 MX grafik platalari xarakteristikalarining taqqoslanishi

Ko'rsatkich	Qiymat	Qiymat
GPU mikroshemasi	GeForce2 MX	RADEON VE
Shina	AGP 4x	AGP 4x
Xotira hajmi va turi	256 MBayt SDR SDRAM	256 MBayt, DDR SDRAM
Rang	32 bit	32 bit (+8 bit kengaytirish uchun)
Piksellar soni	2048 x 1756	2048 x 1536
Koeffitsient T&L	800 Megatexel	30 Megatexel
Ekranni yangilanish chastotasi	183 marta/sek	75-200 marta/sek
Qo'shimcha imkoniyatlar	TV, VCR, DVD, HDTV, MPEG qo'llab-quvvatlanishi	TV, VCR, DVD, HDTV, MPEG qo'llab-quvvatlanishi

**Grafik platalarning dasturiy ta'minoti.** Grafik platalar ko'plagan murakkab funksiyalarni bajarishga mo'ljallangan. Uni bajarish uchun dasturiy mahsulot zarur. U esa har bir plata uchun alohida ishlab chiqilgan. aniq o'rnatilgan standart mavjud emas. Shu sababli kompyuterga yaxshi grafik plataning o'rnatishdan tashqari uning dasturiy

ta'minotini ham o'rnatish zarur. Hozirda OpenGL (Open Graphic Language - ochiq grafik kutubxona) standarti yaratilgan. Va oxirgi yillarda ishlab chiqarilayotgan grafik platalar shu standartga moslanmoqda.

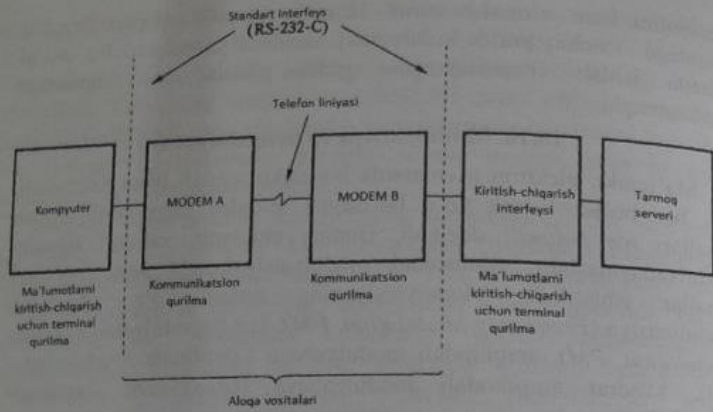
### 18.10. Modulyatsiya va demodulyatsiya

Ma'lumki, elektron sxemalarda bit elektr signali bilan kodlanadi. (tok bor bo'lsa=1, tok yo'q bo'lsa=0). Bunday signallarni uzatish kanallari *tor polosali* deyiladi. Buning alternativ varianti signalni modulyatsiyalashdir. Sinusoidal mudolyatsiyalashda keng polosali kanallar ishlatiladi. Modulyatsiyaning turlari ko'p: chastotali modulyatsiya (*Frequency Modulation, FM*), fazali modulyatsiya (*Phase Modulation, PM*), amplitudali modulyatsiya (*Amplitude Modulation - AM*), kvadrat amplitudali modulyatsiya (*Quadrature Amplitude Modulation*).

Ma'lumotlarni uzatishda signal parametrlari sakrovchan o'zgaradi. Bunday modulyatsiya usuli manipulyatsiya (*keing*) deyiladi. Yuqoridagilarga mos ravishda chastotali manipulyatsiya (*Frequency-Shift Keying*), fazali manipulyatsiya (*Phase-Shift Keying*) va h.k manipulyatsiyalar mavjud.

Bir taktida uzatiladigan signal konfiguratsiyasi *simvol* deyiladi. Chastotali modulyatsiyada ikkita simvol ishlatiladi. Ular  $f_1$  va  $f_2$  sinusoidal signallar bilan ifodalanadi. Kvadrat modulyatsiyada to'rtta simvol qo'llaniladi. Signal amplitudasi va fazasi bilan aniqlanadi. Bir sekundda uzatiladigan simvollar soni bod (*baud*) deyiladi. Uni bir sekunddagi signal holati o'zgarishlari soni bilan ham xarakterlash mumkin. **Ikkilik modulyatsiya sxemasi yoki, chastotali modulyatsiya ishlatilgandagina bir sekundda uzatiladigan bitlar soni va bodlar soni teng bo'ladi.** Kvadrat modulyatsiyada bitli uzatish bodli uzatishdan ikki barobar katta, chunki bir simvol ikki bitga ega bo'ladi. 8, 16 modulyatsiya sxemalari ham mavjud. Masalan 16 modulyatsiya sxemasida bit bitni kodlash uchun 4 bit zarur, bitli uzatish tezligi esa bodlikidan 4 barobar yuqori bo'ladi.

Quyidagi rasmda kommunikatsion kanal oxirida modem bo'lib, signal o'zgartirishni bajaradi.



18.4-rasm. Tarmoqqa masofadan ulanish

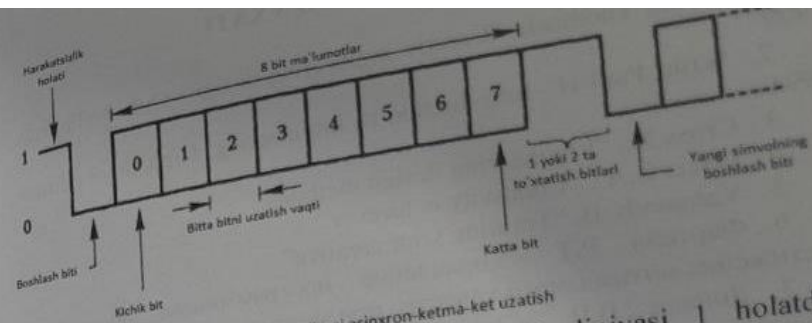
### 18.11. Dupleks ulanish

Kommunikatsiya ulanishi quyidagi uchta sxemada tashkil qilinishi mumkin:

- *Simpleks* ulanish – ma'lumotlar faqat bitta yo'nalishda uzatiladi;
- *Yarimdupleks* ulanish – ma'lumotlar ikki yo'nalishda uzatiladi, lekin bir vaqtda emas
- *Dupleks* ulanish – ma'lumotlar bir vaqtda ikki yo'nalishda uzatiladi.

### 18.12. Asinxron uzatish

Startstop texnologiyasini qo'llagan holda asinxron uzatish jarayonini ketma-ket aloqaning oddiy sxemasi deyish mumkin. Ishonchlilikni ta'minlash maqsadida ma'lumotlar 6 yoki 8 bitdan guruhlanadi (18.5-rasm).



18.5-rasm. Simvollarini asinxron-ketma-ket uzatish

Ma'lumotlar uzatilayotgan holatda uzatish liniyasi 1 holatda bo'ladi. Axborotni uzatish oldidan 0 uzatiladi, uni start biti deyiladi. Undan keyin 8 bit uzatiladi va 1 bit stop biti uzatiladi. Start biti qabul qiluvchiga uzatish boshlanayotganini bildiradi. Uning oldingi fronti uzatuvchi va qabul qiluvchining takt signallarini sinxronlash uchun ishlatiladi. Uzluksiz uzatish jarayonida stop bitlari keyingi porsiyani ajratish uchun xizmat qiladi. Uzatish yakunlangandan keyin oxirgi stop biti (1) ning xolati liniyada saqlab qolinadi. Startstop bitlarini qo'yish va olib tashlashni uzatuvchi-qabul qiluvchining sxemalari bajaradi.

## ADABIYOTLAR RO'YXATI

1. Jorma Tuomaala "Creativ engineering design" Oulu Nyliopisto 1999
2. Wriht Paul H., Introduction to Engineering, High shool edition 1994
3. Cross N., "Engineering design methods"
4. Davis G.A., "Creativity is forever"
5. Vangundy B., "Training your creative"
6. Фаргелл Р.Т «Управление программными проектами: достижение оптимального качества при минимальных затратах»
7. Липаев В.В. «Экономика производства программных продуктов»
8. Кантор М. «Управление программными проектами. Практическое руководство по разработке успешного программного обеспечения»
9. Barry Hyman "Fundamentals of Engineering design"
10. Breakly George C. "Engineering: An introduction to a creative profession"
11. Howell Steven K. "Engineers toolkit. A first course in Engineering: Engineering design and problem solving"
12. Bralla J. G. "Design for excellence"
13. Buhl H. A. "Creative engineering design"
14. Lindbek J.R. "Product design and manufacture"
15. Otto K.N., Wood K.L. "Product design: Techniques in Reserve Engineering and new product development"

## MUNDARIJA

Kirish .....	3
<b>1. Kompyuter arxitekturasi rivojlanish bosqichlari. kompyuterlarni tashkil etish tamoyillari va klassifikatsiyasi...</b>	4
1.1. Arxitektura.....	4
1.2. Klassik arxitektura.....	5
1.3. Ko'p protsessorli arxitektura.....	5
1.4. Parallel protsessorli arxitektura.....	5
1.5. Kompyuterlarni tashkil etish tamoyillari (prinsiplari) va klassifikatsiyasi.....	6
<b>2. Kompyuterlarni tashkil qilishning raqamli mantiqiy asoslari.....</b>	8
2.1. Arifmetik-raqamli qurilma (AMQ) strukturasi.....	8
2.2. AMQning tasniflanishi.....	9
2.3. AMQdagi amallar.....	10
2.4. Arifmetik-mantiqiy qurilmaning ishlashi.....	10
2.5. Arifmetik summatorlar.....	15
<b>3. Kompyuter tizimining tuzilish turlari, kompyuter arxitekturasi va tashkil qilishda qo'llaniladigan standartlar va vositalar, kompyuterlarning tashkiliy qismlari</b>	17
3.1. Komandalarni bajarilishi.....	18
3.2. Zamonaviy kompyuterlar ishlab chiqish tamoyillari.....	20
3.3. Superskalyar arxitektura.....	22
3.4. Vektorli kompyuterlar.....	23
3.5. Multiprotsessorlar.....	24
3.6. Multikompyuterlar.....	25
3.7. Asosiy xotira.....	25
3.8. Xotira adreslari.....	26
3.9. Kesh xotira.....	28
3.10. Xotira modullarini yig'ish va ularning tiplari.....	29
<b>4. Buyruqlar tizimi arxitekturasi tasniflash. operandlarning turlari va formatlari. buyruqlar turlari va formatlari.....</b>	31
4.1. Buyruqlar tizimi arxitekturasi. RISC va CISC protsessorlari.....	31
4.2. Operandlar tipi va formati.....	33
4.3. O'nlik sonlar.....	34
4.4. Mashxur kodlash jadvallari.....	36
<b>5. Adreslash turlari, oqimlarni boshqarish. buyruqlardagi</b>	



<b>manzillash usullari, uzilishlar</b> .....	38
5.1. Adreslash turlari, oqimlarni boshqarish.....	38
5.2. Teskari yozuvda formulalar hisob-kitobi.....	42
5.3. O'tish komandasi uchun adreslash usullari.....	43
5.4. PentiumII protsessorida adreslash usullari.....	43
5.5. Buyruq (komanda) lardagi manzillash usullari, uzilishlar.....	44
5.6. Ma'lumotlarni ko'chirish komandalari.....	44
5.7. Tenglash va shartli o'tish.....	45
5.8. Protseduralarni chaqirish komandalari.....	46
5.9. Sikllarni boshqarish.....	47
5.10. Ketma-ket boshqarish potoki va o'tishlar.....	47
<b>6. Kompyuterlarning asosiy ko'rsatkichlari: massiv-parallel tizimlar (massiv-parallel protsessorlar), simmetrik multiprotsessorli tizimlar (smp), xotiraga bir xil bo'lmagan ruxsatlilik tizimlari (numa)</b> .....	49
6.1. Kompyuterlarning asosiy ko'rsatkichlari.....	49
6.2. Informatsion modellar.....	50
6.3. Multikompyuterlar.....	51
6.4. SIMD kompyuterlari.....	53
6.5. Massiv parallel protsessorlar.....	53
6.6. Vektorli protsessorlar.....	54
6.7. Cray-1 vektorli superkompyuteri.....	55
6.8. Xotiraga bir xil bo'lmagan ruxsatlilik tizimlari (NUMA).....	57
6.9. CC-NUMA multiprotsessorlari.....	58
6.10. Stanford DASH multiprotsessori.....	60
6.11. COMA multiprotsessorlari.....	63
<b>7. Parallel vektorli tizimlar (pvp), klasterli tizimlar</b> .....	64
7.1. Parallel ishlash turlari.....	64
7.2. Parallel qayta ishlash tizimlari klassifikatsiyasi.....	65
7.3. Ma'lumotlarni matritsali qayta ishlash.....	65
7.4. Umummo'ljallangan multiprotsessor tizimlari arxitekturasi.....	66
7.5. Klasterli tizimlar.....	67
<b>8. Xotira ierarxiyasi, asosiy xotira, sinxron va asinxron xotira qurilmalari. operativ va doimiy saqlash qurilmalari. stekli va assotsiativ xotira</b> .....	70
8.1. Xotiraning tipik ierarxiyasi.....	70
8.2. Registrlarning asosiy xarakteristikalari.....	71
8.3. Xotira mikrosxemalari.....	74

8.4. Statik xotira.....	75
8.5. Asinxron xotira.....	76
8.6. Operativ va doimiy saqlash qurilmalari. Stekli va assotsiativ xotira.....	78
8.7. Sinxron xotira.....	80
8.8. DDR SDRAM.....	81
8.9. Katta hajmli xotira strukturasi.....	82
8.10. Dinamik xotiralar tizimi.....	84
8.11. Operativ va doimiy saqlash qurilmalari, stekli va assotsiativ xotira.....	85
8.12. Faqat o'qish uchun mo'ljallangan xotira.....	87
<b>9. Kesh xotira. kesh xotira xususiyatlari, asosiy va kesh xotiralarga murojaat qilish algoritmlari. aralash va bo'lingan kesh xotira. bir bosqichli va ko'p bosqichli kesh xotira</b> .....	92
9.1. Akslanish funksiyasi.....	94
9.2. To'g'ri akslanish.....	94
9.3. Assotsiativ akslantirish.....	95
9.4. To'plam-assotsiativ akslanish.....	96
9.5. Almashish algoritmlari.....	98
9.6. Aralash va bo'lingan kesh xotira. Bir bosqichli va ko'p bosqichli kesh xotira.....	98
9.7. To'g'ri akslanuvchi kesh.....	100
9.8. Kesh xotirani ikkita markaziy protsessorida tashkil qilinish uslubiyati.....	101
<b>10. Virtual xotira tushunchasi, tashqi xotira va ularning turlari. raid massivlar</b> .....	105
10.1. Xotiraning sahifali tashkil qilinishi.....	105
10.2. Xotirani sahifali tashkil qilish uslubiyati.....	107
10.3. Qattiq disk.....	109
<b>11. Flinn klassifikatsiyasi. parallel hisoblash tizimlari. konveyerli hisoblash. superskalyar hisoblashlar</b> .....	114
11.1. Parallel hisoblash tizimlari.....	116
11.2. Ommaviy parallel tizimlar.....	117
11.3. Konveyerli hisoblash. Superskalyar hisoblashlar.....	118
11.4. Superskalyar arxitektura.....	123
<b>12. Hyper-Threading texnologiyasi. Multitasking va ko'p oqimli tizimlar. VLIW, SMT texnologiyalari</b> .....	124
12.1. Texnologiyaning xarakteristikalari.....	124

12.2. Texnologiyaning yutuqlari va kamchiliklari.....	125
12.3. Ishlab chiquvchilarning asosiy vazifasi.....	128
12.4. Dasturiy ta'minot va apparat omillari.....	128
12.5. Hyper-Threadingning amaliy yordami.....	129
12.6. Uskuna ma'lumotlarini qayta ishlash.....	129
12.7. Multitasking va ko'p oqimli (potokli) tizimlar. VLIW, SMT.....	129
12.8. Multitasking va uning xususiyatlari.....	130
12.9. Multitasking tashkil qilishning murakkabliklari.....	131
12.10. Ko'pmasalali (multitasking) operatsion tizimlar tarixi.....	131
12.11. Pseudoparallel multitasking.....	132
<b>13. Kompyuterda ma'lumotlarning uzatilishi. shina: tushunchasi, turlari, iyerarxiyasi. shina hakami (arbitp), shina protokollari.....</b>	<b>136</b>
13.1. Kiritish/chiqarishni shinali tashkil qilish.....	136
13.2. Parallel shinalar.....	139
13.3. VLB.....	139
13.4. PCI Express.....	139
13.5. Razyomlari.....	140
<b>14. Shina hakami (arbitp), shina protokollari.....</b>	<b>142</b>
14.1. Markazlashgan hakamlik.....	142
14.2. Taqsimlangan hakamlik.....	144
<b>15. Kiritish-chiqarish tizimlarining manzil maydonlari. kiritish-chiqarish modullari. kiritish-chiqarishni boshqarish metodlari. kiritish-chiqarish kanallari va portlari, ularning xususiyatlari.....</b>	<b>148</b>
15.1. Uzilishlar.....	150
15.2. Kiritish-chiqarish kanallari va portlari, ularning xususiyatlari.....	152
15.3. Istisnolar.....	152
15.4. Xatolardan keyingi qayta tiklanish.....	153
15.5. Interfeys sxemalar.....	153
15.6. Parallel port.....	154
15.7. Ketma-ket port.....	156
15.8. Kiritish/chiqarish interfeysi standartlari.....	157
<b>16. Protessorlarning tuzilishlari va xususiyatlari. markaziy protessor qurilmasining tuzilishi. registrlar. protessorga kiruvchi va chiquvchi boshqarish signallari. mashina takti.</b>	<b>162</b>

<b>komanda sikli va mashina takti sikli.....</b>	<b>163</b>
16.1. Ma'lumotlarni bir registrdan boshqasiga yuborish.....	164
16.2. Xotiradan so'zni o'qish.....	166
16.3. Xotiraga so'zni yozish.....	166
16.4. Mashina takti. Komanda sikli va mashina takti sikli.....	166
<b>17. Protessorlar turlari: matritsali va vektorli protessorlar. Multiprotessorlar.....</b>	<b>170</b>
17.1. Matritsali protessorlar.....	170
17.2. Matritsali protessorlar xususiyatlari.....	171
17.3. Vektorli protessorlar.....	172
17.4. Klassifikatsiya.....	172
17.5. Dasturlashtirish.....	172
17.6. Intel Pentium protessorlari. Celeron protessorlari. AMD.....	174
17.7. Hyper-Threading texnologiyasi yoki bir CPU doirasida "ikki protessorlik".....	176
17.8. AMD kompaniyasi protessorlari.....	176
<b>18. Tashqi qurilmalarning turlari va vazifalari. tashqi xotira qurilmalari. tarmoq adapteri, strimmer.....</b>	<b>180</b>
18.1. Klaviatura.....	180
18.2. Sichqoncha.....	181
18.3. Trekbol, joystick va sensorli panel.....	182
18.4. Sensorli panel (tachpad).....	183
18.5. Skanerlar.....	184
18.6. Displeylar.....	185
18.7. Printerlar.....	185
18.8. Grafik port.....	185
18.9. Grafik shakllarni qayta ishlanishi.....	185
18.10. Modulyatsiya va demodulyatsiya.....	187
18.11. Dupleks ulanish.....	188
18.12. Asinxron uzatish.....	188
<b>ADABIYOTLAR RO'YXATI.....</b>	<b>190</b>

HAKIMOV Z.A.  
HAMRAYEVA S.I.

# KOMPYUTERNI TASHKIL ETILISHI

O'quv qo'llanma

Toshkent - "METHODIST NASHRIYOTI" - 2024

*Muharrir: Bakirov Nurmuhammad*

*Texnik muharrir: Tashatov Farrux*

*Musahhih: Saidova Nurshoda*

*Dizayner: Ochilova Zarnigor*

*Bosishga 1.04.2024.da ruxsat etildi.*

*Bichimi 60x90, "Times New Roman" garniturasida.*

*Ofset bosma usulida bosildi.*

*Shartli bosma tabog'i 13. Nashr bosma tabog'i 12,25.*

*Adadi 300 nusxa.*

*"METHODIST NASHRIYOTI" MCHJ matbaa bo'limida chop etildi.  
Manzil: Toshkent shahri, Shota Rustaveli 2-vagon tor ko'chasi, 1-uy.*



+99893 552-11-21

*Nashriyot rozilgisiz chop etish ta'qiqlanadi.*

ISBN 978-9910-03-214-1



9 789910 032141

