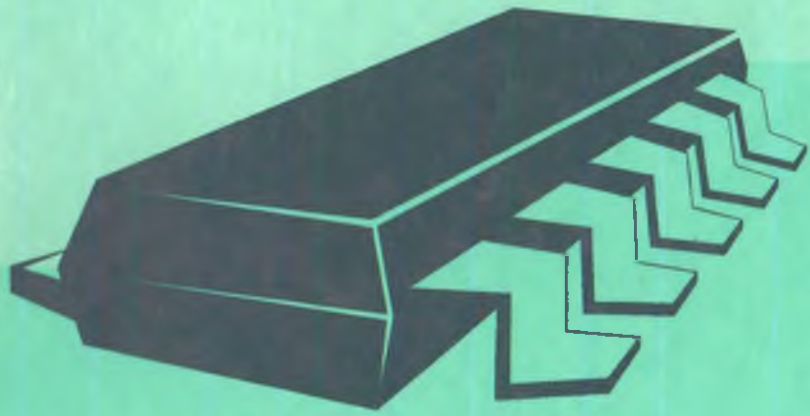
X.YU.ABASXANOVA,
U.B.AMIRSAIDOV

MIKROPROTSESSORLAR



TOSHKENT

O‘ZBEKISTON RESPUBLIKASI
OLIY VA O‘RTA MAXSUS TA’LIM VAZIRLIGI

X.YU.ABASXANOVA, U.B.AMIRSAIDOV

MIKROPROTSESSORLAR

O‘zbekiston Respublikasi Oliy va o‘rta maxsus ta’lim vazirligi tomonidan 5311300- «Telekommunikatsiya» bakalavr ta’lim yo‘nalishi talabalari uchun o‘quv qo‘llanma sifatida tavsiya etilgan

TOSHKENT – 2016

UO'K: 004.3 (075)
KBK 32.973.26-04
A-60

A-60 **X.Yu.Abasxanova, U.B.Amirsaidov. Mikroprotsessorlar –T.: «Fan va texnologiya», 2016, 272 bet.**

ISBN 978–9943–11–342–8

«Mikroprotsessorlar» o'quv qo'llanmasida raqamli qurilmalarni rivojlanish bosqichlari, mikroprotsessorlarni qurish tamoyillari va ishlash asoslari, tarmoq protsessorlarining tuzilishi, rivojlanish bosqichlari va arxitekturasi ko'rib chiqilgan. Turli mikrokontrollerlarning strukturalari keltirilgan, ular uchun dasturiy ta'minot yaratilish masalalari, bosqichlari va shular asosida dasturiy boshqariladigan qurilmalarni yaratilish jihatlari yoritilgan. Shuningdek, ma'lum qurilmalarga oid dasturlar tuzish misollari keltirilgan.

O'quv qo'llanma oliy o'quv yurtlari talabalariga, o'qituvchilariga hamda aloqa va axborotlashtirish sohasi xodimlariga mo'ljallangan.

***

В учебном пособии «Микропроцессоры» рассмотрены вопросы построения цифровых устройств, этапы их развития, принципы построения микропроцессоров и основы их работы, а также архитектурные особенности сетевых процессоров. Описаны различные структуры микроконтроллеров, принципы создания программного обеспечения микроконтроллеров, этапы создания программно-управляемых устройств и примеры программ для отдельных устройств.

Учебное пособие рассчитано для студентов, преподователей высших учебных заведений и работников отрасли связи и информатизации.

***

This study book 'Microprocessors» is intended to consider the issues of creating and designing the digital devices, development trends, and structural principles of microprocessors. Structural and functional features of network processors are described as well. Different structures of microcontrollers, principles of developing software for microcontrollers, stages to create software-controlled devices and case studies for different devices are described.

Study book will be useful for students, teachers of higher education institutions and all those interested in the field of communication and information technologies.

UO'K: 004.3 (075)
KBK 32.973.26-04

Taqrizchilar:
H.Q.Aripov – f.m.f.d., professor;
R.Jurayev – t.f.n., dotsent.

ISBN 978–9943–11–342–8

KIRISH

XXI asr bo‘sag‘asida O‘zbekistonda axborot kommunikatsion texnologiyalari barcha yo‘nalishlar bo‘yicha rivojlanmoqda. Ushbu jarayonni rivojlanishda 2001-yil may oyida Davlatimiz Prezidenti tomonidan Oliy Majlisning 5-sessiyasida ishlab chiqarish, ta’lim, insonlarni kundalik hayotiga axborot texnologiyalarini kiritishni yuqori darajaga ko‘tarish bo‘yicha aniq vazifalar qo‘yildi.

2002-yil 30 mayda O‘zbekiston Respublikasi Birinchi Prezidenti «Kompyuterlashtirish va axborot-kommunikatsion texnologiyalarni kiritishning keyingi istiqbollari haqida»gi qarori imzolandi. Mazkur qarorda kompyuterlashtirishni va axborot-kommunikatsion texnologiyalar (AKT)ni zamonaviy tizimlarini rivojlantirish va kiritish sohalaridagi birlamchi vazifalar aniqlangan. Ular ichiga quyidagilar kiradi:

– real iqtisod tarmoqlarida, boshqaruv, biznes, ilm va ta’lim sohalarida kompyuter va axborot texnologiyalarini keng kiritish, aholi turli qatlamlari uchun zamonaviy kompyuter va axborot tizimlaridan keng foydalanish uchun sharoitlar yaratish;

– maktab, kasb-hunar kollej, akademik litsey va oliy o‘quv yurtlari ta’lim jarayoniga zamonaviy kompyuter va axborot texnologiyalarini faol qo‘llashga asoslangan progressiv ta’lim tizimini joriy qilish;

– axborot-kommunikatsion texnologiyalar sohasida, shu jumladan dasturiy vosita, ma’lumotlar axborot bazalarini tashkil qilish, respublika, soha va lokal axborot-kommunikatsion tarmoqlarni shakllantirish, kompyuter va telekommunikatsion texnikani ishlab chiqish sohasida ishlash uchun yuqori malakali kadrlar tayyorlashni tashkil qilish;

– milliy va xalqaro axborot tizimiga yuqori tezlikdagi kirishni joriy etish, ularga aholi, shu jumladan qishloq punktlarini kiritishni ta’minlash;

– mamlakatning butun hududida axborot-kommunikatsion texnologiyalarni, shu jumladan mobil aloqa, IP texnologiya, telekommunikatsiya va ma'lumot yetkazishni boshqa zamonaviy vositalarini, axborot-kommunikatsion tarmoq va xizmatlarini konvergensiyasini jadal rivojlantirish.

Yangi texnologiyalar kiritish sharoitida mutaxassislar oldida texnologik jarayonlarni o'rnatish, tarkibiy qismlarini qo'llanishi, zamonaviy texnologiyalari asosida tarmoq yaratish kabi masalalar tadqiqoti dolzarb desa bo'ladi.

Tasdiqlangan qonunlarni bajarish jarayoni infokommunikatsion texnologiyalarni O'zbekistonda rivojlanishning tegishli qonunlarni bajarish uchun keng yo'l ochib berdi. Turli telekommunikatsion xizmatlar aholiga ko'rsatilmoqda. 2008-yildagi iqtisodiy inqiroz juda ko'p mamlakatlarda iqtisodiy o'sish jarayonini sekinlashtirishga olib keldi. Ushbu jarayonni I.A.Karimovning «Jahon moliyaviy-iqtisodiy inqirozi, O'zbekiston sharoitida uni bartaraf etishning yo'llari va choralari» nomli asarida ko'rsatilib, uni bartaraf etish uchun mavjud korxonalarni modernizatsiyalash, texnik va texnologik qayta tiklash va zamonaviy texnologiyalarni tadbiq etish eng dolzarb masalalardandir. Yuqorida keltirilgan masalalarni yechishda boshqarish qurilmalarini qurish, ularni tahlil etish jarayonlarini amalga oshiruvchi mutaxassislarni yetishtirish eng asosiy masalalardan biridir. Shuning uchun ushbu qo'llanmada zamonaviy boshqarish qurilmalarini qurishda mikroprotsessorli tizimlarni o'rni kattadir.

Oliy o'quv yurti mutaxassislarni tayyorlashda raqamli texnika va mikroprotsessorlarni bilish, dasturiy boshqariladigan mikroprotsessorlar tizimini yaratish, tahlil etish eng dolzarb muammolardan biridir. Mavjud davrda hisoblash texnikasining eng yuqori pog'onasini egallab turgan kompyuterlar juda tez odimlar bilan rivojlanib kelmoqda. Ushbu rivojlanish davriga mos mutaxassislarni yetishtirib chiqarish, ularni o'qitish va o'quv materiallarini tushunarli, davlat tilida izohlab berish davr talabidir. Bu masalalarni yechish faqat yangi rivojlanayotgan raqamli texnika va uning yuqori pog'onasi bo'lgan mikroprotsessorlar va ular asosida qurilgan boshqaruv qurilmalarining loyihalash, qurish va tahlil etish imkoniyatiga ega bo'lgan mutaxassislarni yetishtirish eng dolzarb muammolardan biridir. Ushbu texnologiyalar juda qisqa vaqt davomida o'zgarishi

mavjud fanni har doim o'zgartirishga, yangi materiallar bilan to'ldirib borishga majbur etadi.

Shuni ta'kidlash lozimki, hisoblash texnologiya nazariyasi va ularning element bazasi juda tezkorlik bilan rivojlanmoqda. Buni hozir rivojlanayotgan nanotexnologiya jarayoni bilan taqqoslash lozimdir. Nanotexnologiya asosida quriladigan boshqarish qurilmalari mavjud texnologiyalarni loyihalash va qurish jarayonlarini amalga oshirish usullari orqali amalga oshirilishi mumkindir. Shuning uchun ushbu fanning salohiyati boshqa fanlarga qaraganda o'zining tezkor o'zgarishi bilan farqlanadi. O'quv qo'llanmada bayon etilgan raqamli qurilmalar, tarmoq protsessorlari, mikroprotsessor tizimlari va mikrokontrollerlar yangi avlod mutaxassislarni tayyorlashda yordam beradi deb umid qilamiz.

Ushbu o'quv qo'llanmada III, IV bobni U.B. Amirsaidov tayyorlagan, kirish qismi, I bob, II bob, IV bob,ilovalarni X.Yu. Abasxanova tayyorlagan.

1. RAQAMLI QURILMALARNING EVOLYUTSIYASI

1.1.Raqamli texnika asoslari

Ixtiyoriy analog signal diskret signallar bilan tasvirlanishi mumkindir [1]. Raqamli qurilma diskret signallarga ishlov beradi (qayta ishlaydi). Raqamli qurilmalarda diskret signallar turli sanoq tizimlarida ifodalanadi.

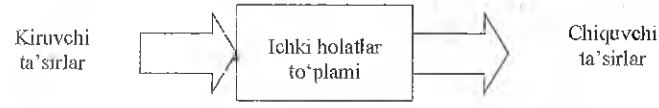
Raqamli qurilmalar ikkilik sanoq tizimida ishlashining matematik asosi bo'lib mantiq algebra yoki bul algebrasi tashkil etadi. Uni XIX asr o'rtasida Irland matematigi Djon Bul ishlab chiqqan [2]. Bul algebrasida ikki qiymatni qabul qiladigan o'zgaruvchilar qo'llanadi: rost hodisa va yolg'on hodisa. Ikkilik sanoq tizimida mazkur tushunchalarga alfavitning ikkita soni mos qo'yiladi: mantiqiy bir (rost hodisa) va mantiqiy nol (yolg'on hodisa). Ikkilik alfavit faqat ikkita simvoldan iborat, shuning uchun nafaqat kiruvchi o'zgaruvchilar, balki chiquvchi funksiya qiymatlari ham faqat ikkita qiymatni olishi mumkin. Ikkilik o'zgaruvchi funksiya Bul funksiyasi yoki mantiqiy funksiya deb ataladi.

Ixtiyoriy raqamli axborot jarayonlari va o'zgartirishlari, qanday murakkab bo'lmasin, natijada oddiy mantiqiy o'zgaruvchilar 1 va 0 ga olib kelinadi. Mantiqiy algebra funksiyalarini shakllantirish uchun mo'ljallangan qurilmalar mantiqiy qurilmalar deb nomlanadi. Ular ikki turg'un holatga ega. Bir holatga mos holda mantiqiy bir qo'yiladi. Ko'p hollarda bu yuqori kuchlanish holati. Boshqa holatga esa mos holda mantiqiy nol qo'yiladi – past kuchlanish holati.

Raqamli qurilmalarni ishlash jarayoni avtomatlar nazariyasi yordamida ta'riflanadi [8,10]. Raqamli avtomatlarni ta'riflash uchun ikki model ishlatiladi: abstrakt va strukturali. Raqamli avtomat abstrakt modelda uchta alfavit va ikkita tavsifiy funksiya orqali taqdim etiladi (1.1-rasm):

Kiruvchi alfavit $X=\{x_1,x_2,\ldots,x_n\}$, chiquvchi alfavit $Y=\{y_1,y_2,\ldots,y_m\}$ va ichki holatlar alfaviti $U=\{u_1,u_1,\ldots,u_k\}$ cheklidir.

O‘tishlar funksiyasi F(U, X) «kiruvchi so‘z-ichki holat» aloqasini tashkil etadi va U da X x U to‘plamini aks ettiradi.



1.1-chizma. Diskret avtomat modeli

Kiruvchi alfavit $X=\{x_1,x_2,\ldots,x_n\}$, chiquvchi alfavit $Y=\{y_1,y_2,\ldots,y_m\}$ va ichki holatlar alfaviti $U=\{u_1,u_1,\ldots,u_k\}$ cheklidir.

O‘tishlar funksiyasi F(U, X) «kiruvchi so‘z-ichki holat» aloqasini tashkil etadi va U da X x U to‘plamini aks ettiradi.

Chiqishlar funksiyasi $\Psi(U, X, Y)$ «chiquvchi so‘z – ichki holat» juftligini bog‘laydi va Y da X x U to‘plamini aks ettiradi.

Shunday qilib, diskret avtomat kiruvchi va chiquvchi alfavit, ichki holat, o‘tish va chiqish funksiyalarining $A=\{ X, U, Y, F, \Psi\}$ to‘plami bilan ta’riflanadi. Diskret (raqamli) avtomatlar diskret vaqtda ishlaydi va diskret axborotni qayta ishlashni amalga oshiradi.

Strukturali model esa mantiqiy elementlardan diskret avtomatni chekli sxemasini qurish uchun mo‘ljallangan.

Raqamli tizimlarning strukturali sxemasini qurishda mantiqiy qurilmalarni texnik o‘ziga xos tomonlarni hisobga olish maqsadida uchta model ishlatiladi: 1) mantiqiy model; 2) vaqtinchalik to‘xtashli model; 3) elektr tavsiflar va parametrlarni hisobga oladigan model.

Mantiqiy model mantiq algebra nazariya asoslariga tayanadi. U nisbatan past tezlikka ega bo‘lgan raqamli qurilmalarni ishini yetarlicha aniq ifodalaydi va 20% ga yaqin qurilmalarni ishlab chiqishda to‘g‘ri keladi. Kechikish holatlarining aniqlash hisobi ikkinchi modelda ishlaydi va o‘tish jarayonlari raqobatlashayotgan jarayonlarini ifodalash uchun zarur va aniqmas ishlashlarni, raqamli qurilma ishiga mos kelmaydigan signallar kombinatsiyasi paydo bo‘lishi holatlarini oldini oladi. Uchinchi modelni murakkab sxemalarni hisoblashda qo‘llash zarur bo‘lib, bunda bitta elementni chiqishiga boshqa ko‘plab elementlar kirishlari ulanadi, ishlatilayotgan quvvat, tok, 0 va 1 mantiqiy sathlar, aloqa tarmog‘idagi signallarni uzatish ishlari o‘ziga xosligini hisobga olgan holda tahlil etiladi.

Mantiqiy (raqamli) qurilmalar turli xususiyatlar bo'yicha sinflanadi.

Axborotni kiritish-chiqarish xususiyati bo'yicha: ketma-ket, parallel va ketma-ket-parallel (aralash).

Ketma-ket qurilmada kiruvchi va chiquvchi simvollar kirishga berilishi va ularning bir vaqtda bajarilmasligi, ya'ni ketma-ket, bir razryaddan so'ng keyingi razryadning bajarilishi asosida amalga oshiriladi.

Parallel qurilmalarda barcha kiruvchi o'zgaruvchilar kirishiga uzatiladi, bunda barcha chiquvchi razryad o'zgaruvchilar razryadlari bir vaqtda olinadi. Kirish va chiqishlar soni kiruvchi va chiquvchi so'zlar razryadlari orqali aniqlanadi.

Ketma-ket parallel qurilmalarda kiruvchi va chiquvchi o'zgaruvchilar turli shaklda taqdim etilishi mumkin. Kirishga ketma-ket ko'rinishda tushadi, chiqishdan esa parallel ko'rinishda olinadi, yoki aksincha.

Mantiqiy qurilmalar ishlash usuli bo'yicha ikki sinfga bo'linadi: kombinatsion va ketma-ket.

Kombinatsion qurilmalarda (xotirasiz avtomatlarda) chiquvchi so'z faqat joriy lahzada faoliyat ko'rsatayotgan kiruvchi simvollar kombinatsiyasiga bog'liq va kiruvchi signallarning oldingi holatlariga bog'liq emas.

Ketma-ket qurilmalarda (xotirali avtomatlarda) chiquvchi so'z nafaqat joriy vaqt lahzasidagi joriy so'zdan, balki oldingi ichki holatga, ya'ni kelib tushgan kiruvchi signallar ketma-ketligiga ham bog'liqdir. Ketma-ket qurilmalar, qurilmaning oldingi ishlashi to'g'risidagi ma'lumotlarni saqlaydi, ya'ni xotiraga egadir.

Xotira hajmi bo'yicha raqamli qurilmalar quyidagi turlarga bo'linadi:

– xotirasiz (kombinatsion qurilmalar);
– chekli xotirali;
– cheksiz xotirali.

Ideallashtirilgan avtomatlarga cheksiz xotirali qurilmalar kiradi. Bunday avtomatlar mavjud emas. Lekin bu model katta xotira va masala shartlari bo'yicha xotira kattaligi va to'lib qolishi mumkin emas bo'lgan hollarda raqamli qurilma ishini tahlil etish va hisoblashlarini sezilarli darajada soddalashtirish uchun qulaydir.

Chiquvchi signalni shakllantirish usuli bo'yicha Mur va Mili avtomatlari bilan farqlanadi.

Mur avtomatlarida chiquvchi Y signal kiruvchi X so'zga bog'liq emas, balki joriy vaqt lahzasidagi ichki U holatga bog'liq:

$$U(t+1)=F(U(t),X(t));$$
$$Y(t)=\Psi(U(t)).$$

Mili avtomatlarida chiquvchi Ysignal, ham ichki holat U, ham kiruvchi X so'z bilan aniqlanadi

$$U(t+1)=F(U(t),X(t));$$
$$Y(t)=\Psi(U(t),X(t)).$$

Agarda ishlash qonunini jadval ko'rinishida keltirilsa, Mili avtomati o'tishlar va chiqishlar jadvali ko'rinishida bo'ladi. Mur avtomatida chiquvchi signal kiruvchi signalga emas, balki ichki holatga bog'liq bo'lganligi sababli, Mur avtomati o'tishlar jadvali bilan ifodalanadi. Umumiy holda avtomatni bir ichki holatdan ikkinchisiga o'tishi kiruvchi signallar ta'siri ostida bo'ladi.

Matematik mantiqning asosiy qismlaridan biri - mantiq algebrasi raqamli qurilmalarning asosi hisoblanadi. Mantiq algebrasi fikrlar bilan ish ko'radi. Fikr deganda haqiqiy yoki yolg'onligi nuqtayi nazaridan bildirilgan har qanday tasdiq tushuniladi. Fikrning haqiqiyligi yoki yolg'onligidan boshqa alomatlari (yaxshi, yomon, nodir va h.k.) e'tiborga olinmaydi.

Mantiq algebrasida fikrlarning haqiqiyligi 1 bilan, yolg'onligi 0 bilan tenglashtirish qabul qilingan. Fikrlarning bu ikkilik tabiatiga mosligini hisobga olib, ularni mantiqiy o'zgaruvchilar deb atashadi. Fikrlar yoki mantiqiy o'zgaruvchilar *oddiy* bo'ladi va lotin alifbosining kichik harflari - x, y, z, x_1, x_2, a, b, . . . bilan belgilanadi.

Oddiy fikrlardan mantiqiy o'zgaruvchilarning ikkilik funksiyalari hisoblanuvchi *murakkab fikrlar* tuziladi. Murakkab fikrlar katta harflar A, B, C, D, E, F, ... bilan belgilanadi va ko'pincha mantiq algebrasining funksiyasi (MAF) deb ataladi.

Mantiq algebrasi elementar mantiqiy funksiyalar yordamida mantiq algebrasi funksiyalarini ifodalash va o'zgartirish bilan shug'ullanadi. MAF larini ifodalash va o'zgartirish masalalari raqamli qurilmalarini loyihalashda keng qo'llaniladi.

Elementar mantiqiy funksiyalar qatoriga avvalo bitta o'zgaruvchi x ning elementar funksiyalarini kiritish mumkin. Bu funksiyalar *haqiqiylik(chinlik) jadvali* deb ataluvchi jadvalda keltirilgan (1.1-jadval). Umuman, chinlik jadvali argumentlarning (mantiqiy o'zgaruvchilarning) mumkin bo'lgan to'plamlaridan har biriga mos funksiya qiymatini akslantiradi.

Chinlik jadvali

1.1-jadval

Funksiya	x argumentli funksiya qiymati		Funksiya belgisi	Funksiya nomi
	0	1		
f_0	0	0	0	doimo volg'on
f_1	0	1	x	o'zgaruvchi
f_2	1	0	$\bar{x}$	inkor
f_3	1	1	1	doimo haqiqiy

Ikkita x va u o'zgaruvchilarning elementar mantiqiy funksiyalarini ko'raylik (1.2-jadval). 1.2 -jadvaldagi funksiyalardan bir qismi trivial hisoblanadi. Masalan, $f_0=0$, $f_{15}=1$ va $f_3=x$, $f_5=y$. Ularning ichida ikkitasi elementar funksiyalardir - $f_{10}=\bar{y}$, $f_{12}=\bar{x}$. f_2 va f_4 funksiyalari esa mos holda u va x bo'yicha ta'qiqiy funksiyalari hisoblanadi.

x va u o'zgaruvchilarning elementar mantiqiy funksiyalari

1.2-jadval

Funksiya	xu argumentli funksiya qiymati				Funksiya belgisi	Funksiya nomi
	0	0	1	1		
	0	1	0	1		

f_0	0	0	0	0	0	doimo yolg‘on
f_1	0	0	0	1	$x \wedge y$	kon’yunksiya
f_2	0	0	1	0	$x\bar{y}$	*u* bo‘yicha ta’qiq
f_3	0	0	1	1	x	x doimo haqiqiy
f_4	0	1	0	0	$\bar{x}y$	x bo‘yicha ta’qiq
f_5	0	1	0	1	y	*u* doimo haqiqiy
f_6	0	1	1	0	$x \oplus y$	x va *u* ni 2 ning moduli bo‘yicha qo‘shish
f_7	0	1	1	1	$x \vee y$	diz’yunksiya
f_8	1	0	0	0	$x \uparrow y$	Pirs strelkasi
f_9	1	0	0	1	$x \sim y$	teng qiymatlilik
f_{10}	1	0	1	0	$\bar{y}$	*u* doimo yolg‘on
f_{11}	1	0	1	1	$x \rightarrow y$	implikatsiya
f_{12}	1	1	0	0	$\bar{x}$	x doimo yolg‘on
f_{13}	1	1	0	1	$y \rightarrow x$	implikatsiya
f_{14}	1	1	1	0	x/y	Sheffer shtrixi
f_{15}	1	1	1	1	1	doimo haqiqiy

Qolganlarini qisqacha tavsiflaylik:

– x va u mantiqiy o‘zgaruvchilarning diz’yunksiyasi. Qisqacha x va u ning diz’yunksiyasi. $x \vee u$ kabi belgilanadi. «x yoki u» deb o‘qiladi. Ta’rifi: x va u mantiqiy o‘zgaruvchilarning diz’yunksiyasi murakkab funksiya bo‘lib, u faqat x va u yolg‘on bo‘lgandagina yolg‘on hisoblanadi (1.3-jadval).

– x va u mantiqiy o‘zgaruvchilarning ***kon’yunksiyasi***. $x \wedge u$ kabi belgilanadi. «x ham u» deb o‘qiladi. Ta’rifi: x va u ning kon’yunksiyasi murakkab funksiya bo‘lib, u faqat x va u haqiqiy bo‘lgandagina haqiqiy hisoblanadi (1.4-jadval).

Mantiqiy o‘zgaruvchilarning dizyunksiyasi

1.3-jadval

$0 \vee 0 = 0$
$0 \vee 1 = 1$
$1 \vee 0 = 1$
$1 \vee 1 = 1$

Mantiqiy o‘zgaruvchilarning kon’yunksiyasi

1.4-jadval

$0 \wedge 0 = 0$
$0 \wedge 1 = 0$
$1 \wedge 0 = 0$
$1 \wedge 1 = 1$

- x va u mantiqiy o‘zgaruvchilarning teng qiymatliligi. $x \sim u$ kabi belgilanadi. «x u ga teng qiymatlik» deb o‘qiladi. Ta’rifi: x va u ning teng qiymatliligi murakkab funksiya bo‘lib, u faqat x va u haqiqiyliklari mos kelgandagina haqiqiy hisoblanadi (1.5.-javdal).

- x va u ni 2 ning moduli bo‘yicha qo‘shish. $x \oplus u$ kabi belgilanadi. «x ni u ga 2 ning moduli bo‘yicha qo‘shish» deb o‘qiladi. Ta’rifi: x va u ni 2 ning moduli bo‘yicha qo‘shish murakkab funksiya bo‘lib, u faqat x va u ning haqiqiyliklari mos kelmaganda haqiqiy hisoblanadi. Ba’zi adabiyotlarda bu funksiyani ***teng*** qiymatlilikning inkori deb ham atashadi(1.6.-javdal).

- x va u ning implikatsiyasi. $x \rightarrow u$ kabi belgilanadi. «Agar x, unda u» deb o‘qiladi. Ta’rifi: x va u ning implikatsiyasi murakkab funksiya bo‘lib, u faqat x haqiqiy, u yolg‘on bo‘lgandagina yolg‘on hisoblanadi (1.7-jadval).

Ekvivalentligi

1.5-jadval

$0 \sim 0 = 1$
$0 \sim 1 = 0$
$1 \sim 0 = 0$
$1 \sim 1 = 1$

MOD qo‘shish

1.6- jadval

$0 \oplus 0 = 0$
$0 \oplus 1 = 1$
$1 \oplus 0 = 1$
$1 \oplus 1 = 0$

Ta’kidlash lozimki, implikatsiya sabab va oqibat orasidagi bog‘lanish ma’nosiga ega emas, ya’ni x ning haqiqiyligidan u ning haqiqiylik sharti kelib chiqmaydi. Aksincha, implikatsiya yordamida tuzilgan murakkab fikrning haqiqiyligi uchun x ning yolg‘onligi kifoya f_{13} funksiya $u \rightarrow x$ ga mos keladi.

- x va u ning Sheffer shtrixi. x/u kabi belgilanadi. «x shtrix u» deb o‘qiladi. Ta’rifi: x va u ning Sheffer shtrixi murakkab funksiya bo‘lib, u faqat x va u haqiqiy bo‘lgandagina yolg‘on hisoblanadi (1.8-jadval).

- x va u ning Pirs strelkasi. $x \uparrow u$ kabi belgilanadi. «x Pirs strelkasi u» deb o‘qiladi. Ta’rifi: x va u ning Pirs strelkasi murakkab funksiya bo‘lib, u faqat x va u yolg‘on bo‘lgandagina haqiqiy hisoblanadi (1.9-jadval).

1.7-jadval
implikat-
siyasi

$0 \to 0 = 1$
$0 \to 1 = 1$
$1 \to 0 = 0$
$1 \to 1 = 1$

1.8-jadval
Sheffer
shtrixi

$0/0 = 1$
$0/1 = 1$
$1/0 = 1$
$1/1 = 0$

1.9-jadval
Pirs trel-
kasi

$0 \uparrow 0 = 1$
$0 \uparrow 1 = 0$
$1 \uparrow 0 = 0$
$1 \uparrow 1 = 0$

Yuqorida ko‘rilgan elementar mantiqiy funksiyalar yordamida ixtiyoriy MAFni tavsiflash mumkin. 1.10-jadvalda uchta o‘zgaruvchili mantiqiy funksiya uchun haqiqatlik jadvali keltirilgan.

Uchta o‘zgaruvchili mantiqiy funksiya uchun haqiqatlik jadvali

1.10-jadval

To‘plam tartib raqami	x_1, x_2, x_3 to‘plamlari	f funksiya qiymati
0	000	0
1	001	0
2	010	0
3	011	1
4	100	0
5	101	1
6	110	1
7	111	1

Mantiq algebrasi elementar funksiyalarining xususiyatlari:

1.3-jadvaldan ko‘rinib turibdiki, elementar funksiyalar o‘zaro ma’lum bog‘lanishlarga ega. Bu bog‘lanishlarni, hamda elementar funksiyalarning xususiyatlarini ko‘rib chiqaylik.

Kon’yunksiya, diz’yunksiya, inkor (VA, YOKI, EMAS) funksiyalari. Mantiq algebrasining asosiy qoidalaridan foydalanib, quyidagi aksiomalarning o‘rinli ekanligiga qanoat hosil qilish mumkin. Aytaylik, x - biror bir mantiqiy funksiya. Unda

1) $x=\bar{\bar{x}}$, mantiqiy ifodadan barcha qo‘shaloq inkorga ega bo‘lgan hadlarni chiqarib tashlab, ularni dastlabki qiymat bilan almashtirish imkoniyatini bildiradi;

2) $\left.\begin{matrix} x \vee x = x \\ x \cdot x = x \end{matrix}\right\}$, bunday o‘zgartirish qoidalari mantiqiy ifoda uzunligini qisqartirishga imkon beradi;

3) $x \vee 0=x$; 4) $x \vee 1=1$; 5) $x \cdot 0=0$; 6) $x \cdot 1=1$; 7) $x \cdot \bar{x}=0$; 8) $x \vee \bar{x}=1$ (mantiqiy haqiqiylik).

Diz’yunksiya va kon’yunksiya arifmetikadagi ko‘paytirish amallariga o‘xshash qator xususiyatlarga ega:

1) assotsiativlik xususiyati (uyg‘unlashish qonuni):

$x \vee (y \vee z)=(x \vee y) \vee z$,

$x(yz)=(xy)z$

2) kommutativlik xususiyati (ko‘chirish qonuni):

$x \vee y=y \vee x$,

$xy=yx$,

3) distributivlik xususiyati (taqsimlanish qonuni):

diz’yunksiyaga nisbatan kon’yunksiya uchun

$x(y \vee z)=xy \vee xz$,

kon’yunksiyaga nisbatan diz’yunksiya uchun

$x \vee yz=(x \vee y)(x \vee z)$

Bu xususiyatlarning o‘rinli ekanligini yuqoridagi aksiomalardan foydalanib isbotlash aytarlicha qiyin emas.

De Morgan qonunlari sifatida ma’lum quyidagi munosabatlarning haqiqatligini ham ko‘rsatish mumkin:

$$\left.\begin{matrix} \overline{xy} = \bar{x} \vee \bar{y}; \\ \overline{x \vee y} = \bar{x}\bar{y}. \end{matrix}\right\} \qquad (1.2)$$

Bu qonundan quyidagini yozish mumkin:

$$\left.\begin{aligned} \overline{xy} &= \bar{x} \vee \bar{y}; \\ \overline{x \vee y} &= \bar{x}\bar{y}. \end{aligned}\right\} \qquad (1.3)$$

demak, kon'yunksiyani diz'yunksiya va inkor orqali yoki diz'yunksiyani kon'yunksiya va inkor orqali ifodalash mumkin.

Mantiqiy funksiyalar uchun singdirish qonuni sifatida ma'lum quyidagi munosabatlar o'rnatilgan:

$$\left.\begin{aligned} x \vee (xy) &= x, \\ x(x \vee y) &= x, \end{aligned}\right\} \qquad (1.4)$$

2 ning moduli bo'yicha qo'shish funksiyasi quyidagi xususiyatlarga ega:

kommutativlik (ko'chirish qonuni)

$x \oplus u = u \oplus x$;

assotsiativlik (uyg'unlashish qonuni)

$x \oplus (u \oplus z) = (x \oplus y) \oplus z$;

distributivlik (taqsimlanish qonuni)

$x(u \oplus z) = (xy) \oplus (xz)$.

Bu funksiya uchun quyidagi aksiomalar o'rinli:

$x \oplus x = 0$; $x \oplus 1 = \bar{x}$;

$x \oplus \bar{x} = 1$; $x \oplus 0 = x$.

Aksiomalar va xususiyatlardan foydalanib VA, YOKI, EMAS funksiyalarni 2 ning moduli bo'yicha qo'shish funksiyasi orqali ifodalash mumkin:

$$\left.\begin{aligned} \bar{x} &= x \oplus 1; \\ x \vee y &= x \oplus y \oplus xy \\ x \cdot y &= (x \oplus y) \oplus (x \vee y). \end{aligned}\right\} \qquad (1.5)$$

Implikatsiya funksiyasi uchun quyidagi aksiomalar o'rinli:

$x \to x = 1$; $x \to \bar{x} = \bar{x}$;

$x \to 1 = 1$; $1 \to \bar{x} = x$;

$x \to 0 = \bar{x}$; $0 \to x = 1$.

Aksiomalardan ko‘rinib turibdiki, implikatsiya faqat ko‘rinishi o‘zgargan kommutativlik (ko‘chirish qonuni) xususiyatiga ega

$x \to u = \bar{u} \to \bar{x}$.

Bu funksiya uchun assotsiativlik xususiyati o‘rinsizdir.

VA, YOKI, EMAS funksiyalari implikatsiya funksiyasi orqali quyidagicha ifodalanadi:

$$\left.\begin{array}{l} x \vee y = \bar{x} \to y; \\ xy = \overline{xy} = x \to \bar{y}; \\ \bar{x} = x \to 0. \end{array}\right\} \qquad (1.6)$$

Sheffer shtrixi funksiyasi uchun quyidagi aksiomalar o‘rinli:

$x/x = \bar{x}$; $x/1 = \bar{x}$;

$x/\bar{x} = 1$; $\bar{x}/0 = 1$;

$x/0 = 1$; $\bar{x}/1 = x$.

Sheffer shtrixi funksiyasi uchun faqat kommutativlik (ko‘chirish qonuni) o‘rinlidir:

$x/u = u/x$,

VA, YOKI, EMAS funktsiyalari Sheffer shtrixi funktsiyasi orqali quyidagicha ifodalanadi:

$$\left.\begin{array}{l} xy = \overline{x/y} = x/y/x/y; \\ \bar{x} = x/x; \\ x \vee y = \overline{\bar{x} \vee \bar{y}} = \overline{xy} = \bar{x}/\bar{y} = x/x/y/y. \end{array}\right\} \qquad (1.7)$$

Pirs strelkasi funksiyasi uchun quyidagi aksiomalar o‘rinli:

$x \uparrow x = \bar{x}$; $x \uparrow 0 = \bar{x}$;

$x \uparrow \bar{x} = 0$; $x \uparrow 1 = 0$.

Pirs strelkasi funksiyasi uchun faqat kommutativlik (ko‘chirish qonuni) xususiyati o‘rinli:

$x \uparrow u = u \uparrow x$.

VA, YOKI, EMAS funksiyalarini Pirs strelkasi funksiyasi orqali quyidagicha ifodalash mumkin:

$$\left.\begin{array}{l} xy = (x \uparrow x) \uparrow (y \uparrow y); \\ x \vee y = (x \uparrow y) \uparrow (x \uparrow y); \\ \bar{x} = x \uparrow x. \end{array}\right\} \qquad (1.8)$$

$f(x_0,x_1,...,x_n)$ funksiyalar mantiqiy (bul) deb nomlanadi, agar uning argumentlari $x_0,x_1,...,x_n$ va funksiya qiymatlari faqat ikkita qiymatni qabul qila oladi: mantiqiy 0 va mantiqiy 1.

Mantiq algebrasi funksiyasini shakllantirish uchun, har bir boshqa funksiyalaridagidek, barcha mumkin bo'lgan kiruvchi argumentlar kombinatsiyalarini berish zarur. Agar argumentlar soni n ga teng bo'lsa, u holda agrumentlar qiymati kombinatsiyalari 2^n ga teng bo'ladi, argumentlarning funksiyalari soni esa 2^{2n}. n=1, bo'lganda funksiyalar soni 2^2 =4 bo'ladi, n=2, bo'lganda funksiyalar soni 2^4 =16 bo'ladi, n=3, bo'lganda funksiyalar soni 2^8 =512 bo'ladi.

Mantiqiy funksiyalarni shakllantirish usullari:

So'zlar orqali. Funksiya qiymatlari va uning argumentlari bog'liqligi so'z iboralari orqali ifodalanadi.

Jadvalli. Jadval usulda rostlik jadvali tuziladi, unda argumentlarning mumkin bo'lgan kombinatsiyalari va mos mantiqiy funksiyalar qiymatlari keltiriladi. Bunday kombinatsiyalar yakuniy bo'lganligi uchun, rostlik jadvali ixtiyoriy argumentlar uchun qiymatni belgilash imkoni yaratiladi. Matematik funksiyalar jadvallaridan farqli ravishda, barcha funksiyalarga qiymatni berish imkonini bermaydi.

Raqamli. Mantiq algebrasi funksiyasini o'nlik sonlar ketma-ketligidek aniqlanadi. Shuningdek, birlik yoki nollik funksiya qiymatlariga mos ikkilik kodi ekvivalentlarini ketma-ket yozilib chiqiladi

Analitik. Mantiq algebrasi funksiyalari analitik ifoda ko'rinishida yoziladi, bularda funksiya argumentlari ustidan bajariladigan mantiqiy amallar ko'rsatiladi.

Bir o'zgaruvchi mantiqiy funksiyalari:

Bir o'zgaruvchi 4 ta funksiyalar mavjud.

Bir o'zgaruvchi funksiyasining rostlik jadvali

1.11-jadval

X Argument	Funksiyalar			
	f_0	f_1	f_2	f_3
0	0	0	1	1
1	0	1	0	1

Bir o‘zgaruvchi funksiyalari argumentlari quyidagi analitik yozuvlar va nomlarga ega:

$f_0(x) = 0$ – nol konstantasi;

$f_1(x) = x$ – x ni qaytarilishi;

$f_2(x) = \overline{x}$ – x ni inkor qilish, EMAS, inversiya, «x emas» deb o‘qiladi;

$f_3(x) = 1$ – bir konstantasi.

f_0, f_1, f_3 bir o‘zgaruvchi funksiyalari texnik realizatsiya nuqtayi nazardan ahamiyatga ega emas. Amaliyotda faqat $f_2(x) = \overline{x}$ funksiyasi – inversiya ishlatiladi.

Ikki o‘zgaruvchi mantiqiy funksiyalari:

Ikki o‘zgaruvchi 16 ta funksiyalar mavjud.

Ikki o‘zgaruvchi funksiyasining rostlik jadvali

1.12-jadval

Argumentlar		Funksiyalar															
x_1	x_2	f_0	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Ikki o‘zgaruvchi funksiyalari argumentlari quyidagi analitik yozuvlar va nomlarga ega:

$f_0(x_1, x_2) = 0$ – 0 konstantasi;

$f_1(x_1, x_2) = x_1, x_2 = x_1 \wedge x_2 = x_1 \& x_2$ – mantiqiy ko‘paytirish, kon’yunksiya, mantiqiy VA;

$f_2(x_1, x_2) = x_1 \Delta x_2$ – x_1 x_2 bo‘yicha man etish; x_1, $x_2$emas;

$f_3(x_1, x_2) = x_1$ – x_1 ni qaytarilishi;

$f_4(x_1, x_2) = x_2 \Delta x_1$ – x_2 x_1 bo‘yicha man etish; x_2, $x_1$emas;

$f_5(x_1, x_2) = x_2$ – x_2 ni qaytarilishi;

$f_6(x_1, x_2) = x_1$ x_2 – 2 modul bo‘yicha qo‘shish, teng ma’no emaslik, mustasno etuvchi YOKI;

$f_7(x_1, x_2) = x_1+x_2= x_1 \vee x_2$ – mantiqiy qo'shish, diz'yunksiya, mantiqiy YOKI;

$f_8(x_1, x_2) = \overline{x_1 \vee x_2} = x_1 \downarrow x_2$– Pirs strelkasi, YOKI inkori; YOKI–EMAS;

$f_9(x_1, x_2) = x_1 \leftrightarrow x_2$– teng ma'nolik,ekvivalentlik,mustasno etuvchi YOKI–EMAS;

$f_{10}(x_1, x_2) = \overline{X}_2$ – x_2 ni inkor etish;

$f_{11}(x_1, x_2) = x_1 \rightarrow x_2 = x_1 \cap x_2$ –implikatsiya; agar x_2,u holda x_1;

$f_{12}(x_1, x_2) = \overline{X}_1$ – x_1 ni inkor etish;

$f_{13}(x_1, x_2) = x_1 \rightarrow x_2 = x_1 \cap x_2$ –implikatsiya; agar x_1,u holda x_2; x_1 x_2 ni olib keladi; $x_1$ni x_2 implikatsiya qiladi;

$f_{14}(x_1, x_2) = x_1 | x_2 = \overline{X_1 X_2}$ - Sheffer shtrixi, VA inkori,VA–EMAS;

$f_{15}(x_1, x_2) = 1$ – 1 konstantasi.

Ikki o'zgaruvchi funksiyasidan quyidagilar amaliy ahamiyatga ega emas: f_0(konstanta 0), f_3($x_1$ni qaytarilishi), f_5(x_2 ni qaytarilishi), f_{15}(konstanta 1).

Ba'zi funksiyalarga so'zlar yordamida ta'rif beramiz.

Mantiqiy qo'shish. Diz'yunksiya. YOKI funksiyasi birlik qiymat qabul qiladi, agar kamida bir YoKI x_1, YOKI x_2 argumenti birga teng bo'lsa.

Mantiqiy ko'paytirish. Kon'yunksiya. VA funksiyasi birlik qiymatni qabul qiladi, agar bir vaqtda ikki VA x_1, VA x_2 argument birga teng bo'lsa.

Inversiya.EMAS funksiyasi x argumentiga teskari qiymatni qabul qiladi.

Mantiqiy funksiyani raqamli shaklini f_6 misolida ko'ramiz, u kiruvchi o'zgaruvchilar (x_1x_2) kiritishda ikkilik kodda birlik qiymatni qabul qiladi, bu 1;2 o'nlik ekvivalentga teng:

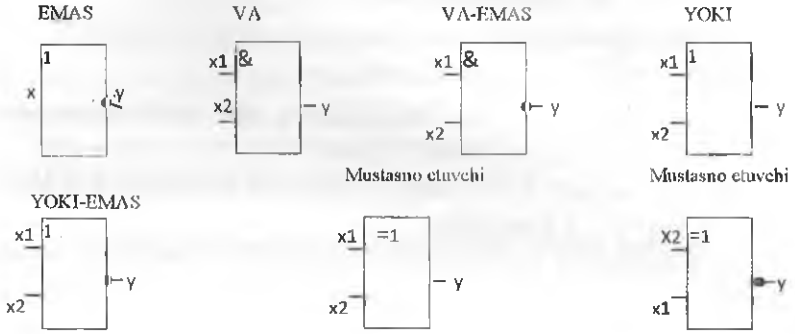
$$f_6(x_1, x_2) = \sum(1,2) = \vee(1,2). \qquad (1.9)$$

f_6 funksiyasi ikkilik kodda 00,11 kiruvchi qiymatlar (x_1x_2) to'plamida nol qiymatini qabul qiladi. O'nlik kodda bu 0;3ga mos:

$$f_6(x_1, x_2) = P(1,2) = \wedge(1,2).$$

Ikki va bir o‘zgaruvchilar mantiqiy funksiyalari elementar deb nomlanadilar. Ular faqat bir amalni bajarishni nazarda tutadilar.

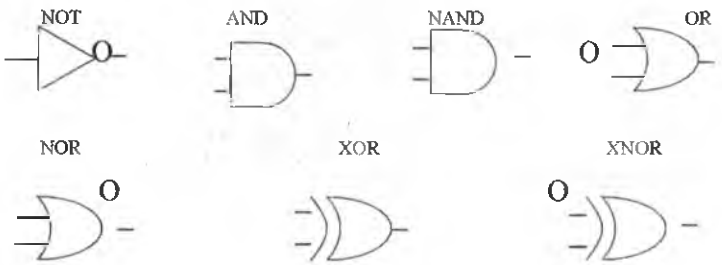
Raqamli qurilmalarda mantiqiy funksiyalarni mantiqiy elementlar amalga oshiradilar. Eng ko‘p tarqalgan EMAS, VA, YOKI, VA–EMAS, YOKI–EMAS, mustasno etuvchi YOKI, mustasno etuvchi YOKI–EMAS elementlari 1.2.-chizmada keltirilgan.



1.2-chizma. Mantiqiy elementlarning shartli belgilanishlari

Raqamli elementlar to‘g‘ri burchak asosida chiziladi. Funksional asosiy maydonning yuqori qismida ko‘rsatiladi. Kirishlar chapda x harfi bilan belgilab ko‘rsatiladi, chiqishlar esa o‘ng tarafda u harfi bilan belgilagan holda ko‘rsatiladi. Invers kirish yoki invers chiqishlar aylana bilan belgilanadi.

Chet el adabiyotlarida mantiqiy elementlarni boshqa ko‘rinishda belgilash qabul qilingan (1.3.-chizma).



1.3- rasm. Chet el adabiyotlarida mantiqiy elemetlarni belgilash

Barcha mantiqiy amallarni bajaruvchi mantiqiy elementlarni ishlab chiqish amaliyotda o‘z tasdig‘ini topdi. Bundan tashqari, o‘zgaruvchilar soni oshishi bilan mantiqiy funksiyalar juda kattalashmoqda. Keyinchalik mantiqiy funksiyalarni cheklangan elementlarni qo‘llagan holda murakkab mantiqiy funksiyani amalga oshirish yo‘li ko‘rsatiladi.

Yuqorida mantiqiy elementlarni ifodalashda jadval usulidan foydalangan edik. Jadval usulida o‘zgaruvchilar qiymatlarining har bir to‘plamiga haqiqiylik jadvalida mantiqiy funksiya qiymati to‘g‘ri kelar edi. Bu usul ixtiyoriy sonni o‘zgaruvchi funksiyalarini yozishga imkon bersada, bunday yozuv MAFlami tahlil etishda ixcham bo‘lmaydi. Formula ko‘rinishidagi analitik yozuv soddaroq hisoblanadi.

Mantiqiy algebra funksiyasi berilgan o‘zgaruvchilarning belgilangan to‘plami $\{x_1, x_2, ..., x_n\}$ ni ko‘raylik. Ixtiyoriy o‘zgaruvchi $x_i=\{0,1\}$ bo‘lganligi sababli o‘zgaruvchi qiymatlarining to‘plami aslida qandaydir ikkilik sondan iborat. To‘plamning tartib raqami ixtiyoriy ikkilik son i deb faraz qilib, quyidagini olamiz:

$$i=x_1\cdot 2^{n-1}+x_2\cdot 2^{n-2}+...+x_{n-1}\cdot 2^1+x_n.$$

Aytaylik, quyidagi $F_i(x_1, x_2, .., x_n)$ funksiya mavjud:

$$F_i = \begin{cases} 0, \text{ agar to‘plamning tartib raqami } i \text{ bo‘lsa,} \\ 1, \text{ agar to‘plamning tartib raqami } i \text{ bo‘lmasa,} \end{cases}$$

F_i funksiya term deb ataladi.

Diz’yunktiv term (maksterm) - to‘g‘ri va invers shaklda ifodalangan barcha o‘zgaruvchilarni diz’yunksiya belgisi bilan bog‘lovchi term (ba’zi adabiyotlarda «nulning konstituenti» atamasi ishlatiladi).

Masalan,

$F_1= \bar{x}_1 \vee x_2 \vee x_3 \vee x_4,$

$F_2= x_1 \vee \bar{x}_2,$

Kon’yunktiv term (minterm) - to‘g‘ri va invers shaklda ifodalangan barcha o‘zgaruvchilarni kon’yunksiya belgisi bilan bog‘lovchi term (ba’zi adabiyotlarda «birning konstituenti» atamasi ishlatiladi).

Masalan,

$F_1=x_1\ \bar{x}_2\ \bar{x}_3\ \bar{x}_4,$

$F_2 = \bar{x}_1\ \bar{x}_3\ \bar{x}_4$,

Termning darajasi r termga kiruvchi o'zgaruvchilar soni bilan aniqlanadi.

Masalan,

$F_1 = \bar{x}_1 x_2\ \bar{x}_3 x_4\ \bar{x}_5$, minterm uchun $r=5$,

$F_1 = \bar{x}_1 \vee x_2 \vee \bar{x}_3$, maksterm uchun $r=3$,

Yuqorida keltirilganlarga asoslanib, quyidagi teoremani ta'riflash mumkin:

Teorema. Jadval ko'rinishida berilgan ixtiyoriy MAF quyidagi ko'rinishda analitik ifodalanishi mumkin [ ]:

$$f(x_1,x_2,\ldots,x_n) = F_1 \vee F_2 \vee \ldots \vee F_n = \bigvee_i F_i \qquad (1.10)$$

bu yerda, i-funksiya 1 ga teng bo'lgan to'plamlarning tartib raqami; $\bigvee_i$ - 1 ga teng bo'lgan barcha F_i termlarni birlashtiruvchi diz'yunksiya belgisi. Haqiqatan, qandaydir to'plamda funksiya $f(x_1^*,x_2^*,\ldots,x_n^*)=1$ bo'lsa, $x \vee 1=1$ bo'lganligi sababli (1.10) ifodaning o'ng tarafida 1 ga teng bo'lgan element doimo topiladi; agar i-to'plamda funksiya $f(x_1^*,x_2^*,\ldots,x_n^*)=0$ bo'lsa, (1.10) ifodaning o'ng tarafida bitta ham 1 ga teng bo'lgan element topilmaydi, chunki $0 \vee 0 \vee \ldots \vee 0 = 0$.

Shunday qilib, $f_i=1$ bo'lgandagi har bir i-to'plamga $F_i=1$ bo'lgan element to'g'ri keladi, $f_i=0$ bo'lgandagi to'plamlarga esa bitta ham $F_i=1$ bo'lgan element to'g'ri kelmaydi. Shu sababli, haqiqiylik jadvali (1.11-jadval) ko'rinishidagi analitik yozuv orqali bir qiymatli akslantiriladi. (1.11-jadval) ifodani termlarning birlashtirilishi deb yuritiladi.

O'zgaruvchan darajali mintermlarni o'z ichiga oluvchi termlar birlashmasi diz'yunktiv normal shakl(DNSh) deb ataladi.

Teorema. Jadval ko'rinishida berilgan ixtiyoriy MAF quyidagi ko'rinishida analitik ifodalanishi mumkin:

$$f(x_1,x_2,\ldots,x_n) = F_1 \wedge F_2 \wedge \ldots \wedge F_k, \qquad (1.11)$$

bu yerda, k - $f=0$ bo'lgandagi ikkilik to'plamlar soni.

O‘zgaruvchan darajali makstermlarni o‘z ichiga oluvchi termlar birlashmasi kon’yunktiv normal shakl (KNSh) deb yuritiladi.

(1.11) teoremadan quyidagi xulosa kelib chiqadi: jadval ko‘rinishida berilgan ixtiyoriy MAF quyidagi analitik shaklda ifodalanishi mumkin:

$f(x_1,x_2,...,x_n)=F_1 \equiv F_2 \equiv ... \equiv F_k$,

bu yerda k - funksiyaning no‘llik qiymatlari soni.

Mintermlar (makstermlar) asosida MAF larning kanonik diz’yunktiv (kon’yunktiv) shakllari tuziladi.

DNSh (KNSh) kanonik deyiladi, agar ularning barcha elementar kon’yunksiyalari (diz’yunksiyalari) mintermlar (makstermlar) bo‘lsa. Har qanday MAF faqat bitta diz’yunktiv kanonik shaklga (DKSh) va faqat bitta kon’yunktiv kanonik shaklga (KKSh) ega bo‘ladi. Kanonik shakllar mukammal kanonik shakllar deb ham ataladi.

MAF ning mukammal diz’yunktiv normal shakli (MDNSh) va mukammal kon’yunktiv normal shakli (MKNSh) mos haqiqiylik jadvallar yordamida tuzilishi mumkin.

MDNSh - MAF ning qiymati 1 ga teng bo‘lgan to‘plamlarga mos keluvchi mintermlar diz’yunksiyasidir.

Masalan, 1.11 - jadvalda keltirilgan funksiyaga quyidagi MDNSh mos keladi:

$f = \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3$

MKNSh haqiqiylik jadvali yordamida quyidagicha aniqlanadi. Funksiyaning qiymati 0 ga teng bo‘lgan to‘plamlarning har biri uchun maksterm aniqlanadi. Bunda to‘plamdagi 0 qiymatli o‘zgaruvchiga o‘zgaruvchining o‘zi mos kelsa, 1 qiymatli o‘zgaruvchiga o‘zgaruvchining inkori mos keladi.

Masalan, 1.11-jadvaldagi funksiyaga quyidagi MKNSh to‘g‘ri keladi:

$f=(x_1 \vee x_2 \vee \bar{x}_3)(x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2 \vee x_3)$

Demak, mukammal normal shaklning normal shakldan farqi, undagi termlar faqat maksimal darajaga ega bo‘lishi va funksiyani bir qiymatli ifodalashga imkon berishidir.

Ixtiyoriy diz’yunktiv normal shaklga o‘tish quyidagicha amalga oshiriladi:

aytaylik, $f_{DNSh}=F_1$ bo‘lsin. Unda

$$f_{DNSh}=F_1 x_1 \vee F_1 \bar{x}_i, \qquad (1.12)$$

bu yerda x_i - berilgan F_1 termga kirmaydigan o'zgaruvchi.

Misol. Quyidagi DNSh da berilgan mantiqiy funksiyani MDNSh ga o'tkazish lozim:

$$f(x_1, x_2, x_3, x_4)=\underset{F_1}{x_1 \bar{x}_2} \vee \underset{F_2}{x_2 \bar{x}_3 \bar{x}_4} \vee \underset{F_3}{\bar{x}_1 \bar{x}_3 x_4} \vee \underset{F_4}{x_1 x_2 x_3 x_4}$$

Yechish. (112) o'zgartirishni navbat bilan barcha termlarga qo'llaymiz:

$$F_1= x_1 \bar{x}_2 (x_3 \vee \bar{x}_3)= x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3.$$

Olingan ifodadagi ikkala hadni $(x_4 \vee \bar{x}_4)$ ga ko'paytiramiz. Natijada quyidagini olamiz:

$$F_1=(x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3)(x_4 \vee \bar{x}_4)= x_1 \bar{x}_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4 \vee x_1 x_2 x_3 \bar{x}_4.$$

Xuddi shunday:

$$F_2= x_2 \bar{x}_3 x_4 (x_1 \vee \bar{x}_1)= x_1 x_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4;$$

$$F_3= x_1 \bar{x}_3 x_4 (x_2 \vee \bar{x}_2)= x_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4.$$

Soddalashtirishdan so'ng quyidagini olamiz:

$$f_{MDNSh}(x_1, x_2, x_3, x_4)= x_1 \bar{x}_2 x_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4 \vee \vee x_1 x_2 x_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 x_4 \vee \bar{x}_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4 \vee x_1 x_2 x_3 \bar{x}_4.$$

Agar funksiyaning maksimal darajasi r ga, j nchi termning minimal darajasi k ga teng bo'lsa (1.12) o'zgartirishni r-k marta qo'llash zarur.

Normal shakllarda ifodalashda elementar funksiyalarning chegaralangan sonidan foydalaniladi. Masalan, MDNSh uchun elementar funksiyalar sifatida «kon'yunksiya», «diz'yunksiya» va «inkor» ishlatiladi. Demak, ixtiyoriy murakkablikka ega bo'lgan mantiqiy funksiyalarni analitik ifodalovchi mantiq algebrasi funksiyalari sistemasi mavjud. Raqamli avtomatlarni loyihalash xuddi shunday funksiyalar sistemasiga asoslanadi.

Ta'rif. Mantiq algebrasi funksiyalarining funksional to'liq sistemasi – bazis deb shunday mantiqiy funksiyalar majmuasiga aytiladiki, bu majmua yordamida ixtiyoriy mantiqiy funksiyani ifoda ko'rinishida yozish imkoni bo'lsin.

Bazisga quyidagi funksiyalar sistemasi kiradi: VA, YOKI, EMAS (1-bazis); VA, EMAS (2-bazis); YOKI, EMAS (3-bazis);

Sheffer shtrixi (4-bazis); Pirs strelkasi (5-bazis). Bazislar ortiqchalik (1-bazis) va minimal (4, 5-bazislar) bo'lishi mumkin.

1-bazis ortiqchalik sistema hisoblanadi, chunki undan biror-bir funksiyani chiqarib tashlash mumkin. Masalan, De Morgan qonunidan foydalanib VA funksiyasini YOKI va EMAS funksiyalari yoki YOKI funksiyasini VA va EMAS funksiyalari bilan almashtirish mumkin.

Agar ifodalashning turli shakllari minimallik nuqtayi nazaridan taqqoslansa, ravshanki, normal shakllar mukammal normal shakllarga qaraganda tejamli hisoblanadi. Ammo, normal shakllar bir qiymatli akslantirishni bermaydi.

MAF larning sonli ifodalanishi. Mantiq algebrasi funksiyalarining yozilishini soddalashtirish maqsadida termlarni to'liq sanab o'tish o'rniga funksiya 1 qiymatini (MDNSh uchun) yoki 0 qiymatini (MKNSh uchun) qabul qiluvchi to'plamlar tartib raqamidan foydalaniladi. Masalan, 1.10-jadvalda keltirilgan funksiya quyidagi ko'rinishda yozilishi mumkin:

$$f(x_1, x_2, x_3) = 3 \vee 5 \vee 6 \vee 7 = \vee(3, 5, 6, 7)$$

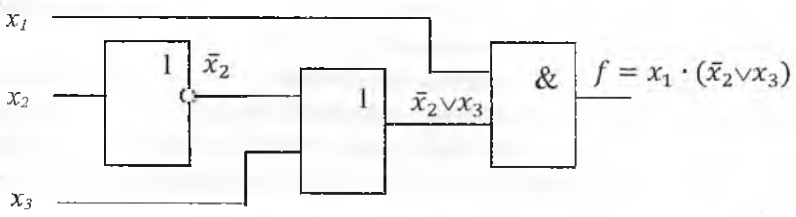
ya'ni funksiya faqat 3, 5, 6, 7-to'plamlarda birlik qiymatiga ega. Yoki

$$f(x_1, x_2, x_3) = 0 \wedge 1 \wedge 2 \wedge 4 = \wedge(0, 1, 2, 4)$$

ya'ni, funksiya faqat 0, 1, 2, 4-to'plamlarda nollik qiymatiga ega.

Mantiqiy elementlar yordamida ixtiyoriy murakkab MAF ni ifodalovchi kombinatsion sxemani tuzish mumkin.

Misol. $f = x_1 \cdot (\bar{x}_2 \vee x_3)$ funksiya uchun kombinatsion sxema 1.7. -chizmada keltirilgan.



1.7-chizma. $f=x_1\cdot(\bar{x}_2 \vee x_3)$ **funksiyaning kombinatsion sxemasi**

1.2. Hisoblash mashinalarning evolyutsiyasi

Zamonaviy hisoblash texnikasining (HT) hozirgi holati ko'p-yillik evolyutsiyaning mahsulidir. Oxirgi vaqtlarda HT ning rivojlanish masalasi olimlarning e'tiborini o'ziga tortdi. Natijada taraqqiyotning «Kompyuter rivojlanish nazariyasi» nomini olgan fanning yangi bosqichi rivojlandi. Nazariyani yaratgan olimlarning e'tiborini HT ning rivojlanish taraqqiyoti biologiyaning rivojlanish taraqqiyotiga o'xshashligi o'ziga tortdi. Yangi fanga quyidagi postulatlar asos bo'ldi:

«Tirik» hisoblash tizimi «O'lik» elementlardan o'zi hosil bo'ladi (biologiyada bu abiogenez nomi bilan ataladi)

Taraqqiyot bo'ylab ketma-ket surilish hisoblash mashinasining protoprotsessordan (bitta protsessorli) hisoblash tizimining poliprot-sessorigacha (ko'p protsessorli).

Hisoblash tizimining texnologiyasida foydali mutatsiya va variatsiya jarayoni hosil bo'ladi. Yagona tanlov natijasida eskirgan texnologiyalarning o'lchanishi.

Mutaxassislarning fikriga ko'ra kompyuter rivojlanish nazariyasi doirasida hisoblash mashinasi va tizimlarini rivojlanish qonunini o'rganishda amaliyotda sezilarli natijalarga biologiya kabi erishish mumkin.

Mur Qonuni. 1995-yil 19-aprelda «Electronics» jurnalida «Izla-nuvchi olimlar kelajakka nazar solishadi» nomli sonida jahonga mashhur Gordono Murning «Cramming more components onto integrated circuits» (ko'p miqdorli kompyuterlarning integral sxemalarda birlashishi) nomli maqolasida chop etildi.

Fairchild Semiconductors kompaniyasining qayta ishlash bo'limining direktori Murning (Intel korparatsiyasining asoschisi) bir maqolasi 6-yillik mikroelektronikaning rivojlanishi asosida 10-yilda mikroelektronikaning rivojlanishi kristaldagi elementlar miqdorini har 2-yilda mikrosxemada ko'payishi haqida edi.

Mur Qonunining bir qancha ko'rinishlari mavjud [7].

Eng qulay kristaldagi tranzistorlarning soni har -yilda 2 marotaba ortib boradi.

Chiqarilayotgan chiplardagi tranzistorlar soni har -yilda 2 marotaba ortib boradi.

Mikroprotssessorlarni taktli chastotasi har 18 oyda ikki barobar ortadi.

Texnikaning rivojlanishida dualizm. Inson va jamiyatning rivojlanishida texnikaning ahamiyati juda katta. Insonlarning fizik va hisoblash imkoniyatlarini ortishida ular tomonidan yaratilgan mashina va mashinalar tizimi katta rol o'ynaydi. Texnikaning rivojlanishida o'ziga xos dualizm o'rnatildi. U quyidagi ikki evolyutsion qator orqali ko'rsatilgan.

1.8-chizma. Fizik qator

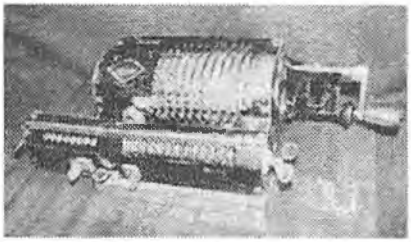
1.9-chizma. Hisob qatori

Hisoblash texnikasi tarixi ikki davrga bo'linadi:

Oddiy mexanik va elektromexanik qurilmalar va hisoblash mashinalari davri (qadimgi tarix).

EHM va parallel hisoblash tizimi davri (yangi tarix).

Mashinalarning mexanik asri.



1-rasm. Arifmometrlar

Arifmometr - (grekcha arithmos -son va metrov-o'lchov) qo'lda boshqariladigan to'rt xil arifmetik amallarni bajara oladigan mexanik hisoblash mashinasi.

1492-yil. Leonardo Da Vinchi o'zining kundaliklarida tishli g'ildirak asosida 30 raztyadli 10 lik jamlovchi qurilmani tasvirini chizgan.

1642-yil. Blez Paskal «Paskalin» deb nomlangan birinchi mexanik raqamli hisoblash qurilmasini taqdim etdi. Bu qurilma 10 lik sanog'ini 5 razryadigacha jamlay olgan.

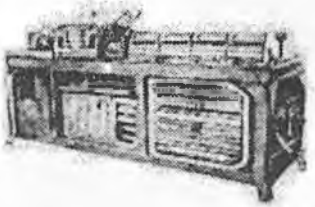
1673-yil. Gotfrid Vilgelm Leybnts to'rtta arifmetik amalni bajara oladigan 12 razryadli 10 lik jamlovchi qurilmani yaratdi.

1786-yil. Nemis harbiy muhandisi Iogann Myuller logorifmlarni tabulyatsiyalash uchun mo'ljallangan mashina ishlab chiqarish g'oyasini ilgari surdi.

1874-yil konstruksiyalashtirilgan arifmometr amaliyotda keng foydalanila boshlandi. Bu arifmometr peterburgli mexanik V.T.Odner tomonidan yaratilgan bo'lib, Rossiya va chet elda ishlab chiqarila boshlandi.

Odner arifmometri 1960-yilda ishlab chiqarilgan «Feliks» modeliga prototip bo'lib xizmat qildi.

Hisob - analitik mashinalari XIX asr oxiri XX asr boshlarida paydo bo'la boshladi. Bu mashinalar moliya - bank operatsiyalari, buxgalteriya hisobi, statistika va hisob matematikasi masalalarini yechish uchun ishlatildi. Bunday mashinalar bilan nafaqat hisob mexanizatsiyasini maksimal darajaga erishdi, balki sonlarni kiritish va turli operatsiyalarni amalga oshirishni avtomatlashtirish imkoniyatini berdi. Ularda ma'lumotlarni kiritish va ishni boshqarish uchun perfokartalardan foydalanildi.



2-rasm. Hisob-analitik mashinalari

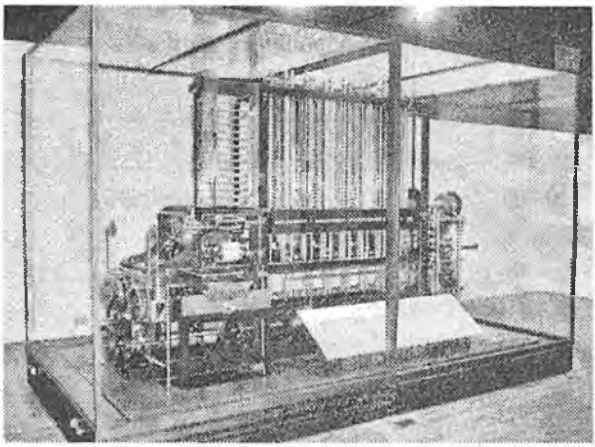
Hisob - analitik mashinalari quyidagilarni o'z ichiga oluvchi jumlasini hisoblanadi. Perfokartaga kiritilgan sonlar ustida amallarni bajaradigan mashina.

Jamlovchi mashinalar (tabulyatorlar). Ko'paytiruvchi mashinalar (ko'paytiruvchi perforatorlar yoki multipleyerlar). Mantiqiy-arifmetik operatsiyalarni amalga oshirish uchun mo'ljallangan mashinalar. Perforatorlar. Yordamchi mashinalar - kontrol apparatlari, bir kartadan boshqa kartaga o'tkazuvchi reproduktorlar.

Differensial tenglamalarni yechish uchun mo'ljallangan birinchi hisoblash mashinasi Rossiyada 1904-yil matematik, mexanik va kemasoz A.N.Krilov tomonidan yaratilgan.

Hisob-analitik texnikasining aniq kompleksi bir qancha qurilmalardan tashkil topgan, ammo eng asosiylari quyidagi to'rtta qurilma: kiruvchi perforator, kontrolnik sortlarga ajratuvchi mashina va tabulyator. Perforator perfokartadagi yoriqlarni teshish uchun ishlatiladi, kontrolnik esa bu yoriqlar to'g'ri teshilishini nazorat qiladi, ya'ni amaldagi hujjatdagi ma'lumotni perfokartaga to'g'ri o'tkazilishini nazorat qiladi. Odatda kontrolnik perforator asosiga teshuvchi qurilma bilan konstruksiya qilinadi. Sortlarga ajratuvchi mashinaning asosiy funktsiyasi kelgusida perfokartani tabulyatorda qayta ishlash uchun guruhlashdan iborat.

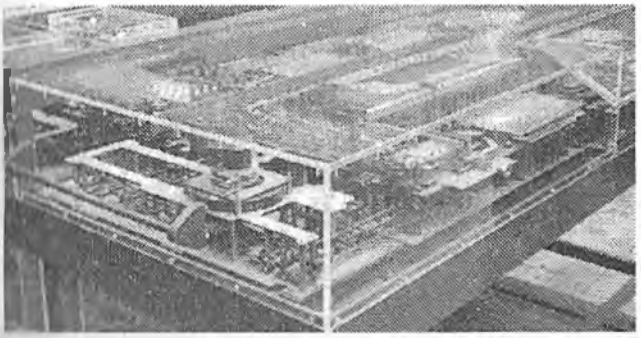
Bebbijning hisoblash mashinasi. Katta hisoblash mashinasini yaratish g'oyasi Kembrij Universiteti professori Charlz Bebbejga tegishli. U avtomatlashgan mexanik raqamli kompyuter yaratmoqchi bo'lgan. Hisoblash mashinasining loyihasi 1833-yilda ishlab chiqilgan.



3-rasm. Bebbijning hisoblash mashinasi

Bebbijning mexanik mashinasi funksional tizimi jihatdan birinchi elektron hisoblash mashinalariga yaqin. Hisoblash mashinalarini o'rganishda arifmetik va eslab qoluvchi qurilma, axborot kiritish va chiqarish hamda boshqaruv qurilmalari ko'zda tutilishi kerak. Avtomatlashgan hisob perfokartadagi ketma-ket kodlangan dasturga bog'liq holatda boshqaruv qurilmasi yordamida ta'minlangan. Bebbij mashinasi olingan natijalarga qarab dasturdagi qadamlarni o'zgartirish mumkin bo'lgan. Mashina bir necha ming hisob g'ildiraklaridan tashkil topgan bo'lib, hajmi 1000 50-razryadli songa ega eslab qoluvchi qurilmaga ega. Bu mashina bir necha kvadrat metr maydonga bemalol joylasha olgan.

1835-yilda oddiy konfiguratsiyali hisoblash mashinalari qurildi. Ular asosan algebraik tenglamalar va logarifmlarni yechish uchun ishlatildi.Bu mashinalar qidirilayotgan yechimni bir minut ichida yechgan (qachonki bu masala uchun tajribali matematik bir kun sarflagan). Bebbej loyihasi texnika va texnologiyalar imkoniyatlarini amalga oshirish ancha qimmat edi. Shu sababli Britaniya parlamenti 1842-yili bu loyihaga pul ajratishni to'xtatdi. Bebbej 30-yil davomida shu ish bilan shug'ullandi va 239 ta detal chizmasini ishlab chiqdi.



4-rasm. Konrad Suze hisoblash mashinasi

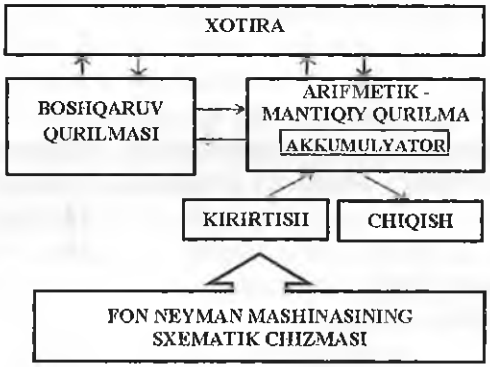
Z1 modeli 1938-yil qurilgan. Bu dunyodagi birinchi boshqaruv dasturiga ega raqamli mexanik kompyuter hisoblanadi. Z1 ning arxitektura xususiyati shundan iboratki, ikkilik kodlash va sonlarni suzuvchi vergul (yoki yarim logorifmik tizim) bilan ko'rsatish tizimi ko'rsatilgan. Bunda son uzunligi 21 razryad bo'lib, 1 razryad son belgisi bilan ajratilgan, 7 razryad tartib uchun qo'llanilgan, 13 razryad mantissalar uchun. Z1 hisoblash mashinasi test model bo'lib, hech qachon amaliy maqsadda qo'llanilmagan. Bu mashina 1980-yillarda Berlinda Suze tomonidan qayta konstruktorlashtirilgan. Hozirda Berlin transport va texnologiyalari muzeyida eksponat sifatida saqlanmoqda.

1940-yili Z2 modeli yaratildi. Unga birinchi marta elektromexanik rele qo'llanildi Z2 mashinasida arifmetik va boshqaruv qurilmalari releda amalga oshirildi, xotira esa mexanikligicha qolaverdi (Z1 xotirasi). Bunday gibrid hisoblash mashinasi yetarlicha ishonchli bo'lmagan va amalda deyarli qo'llanilmagan.

Z3 modeli-dunyodagi birinchi dasturiy boshqariluvchi ikkita elektro-mexanikli hisoblash mashinasi. Z3 mashinasini yaratish 1939-yillarda boshlandi va 1941-yil 5 dekabrda to'liq tugatildi. Z3 hisoblash mashinasining imkoniyatlarini ko'rib chiqamiz. Bunda Z3 mashinasini funksional tizimi va texnik xarakterini ko'rishimiz mumkin. Z3 mashinasi qo'shish, ayirish, bo'lish, ko'paytirish , kvadrat ildiz chiqarish va yordamchi funksiya (sonlarni ikkilik

o‘nlikka aylantirish) kabi operatsiyalarni bajarish uchun mo‘ljallangan. Sonlar ikkilik sistemasida taqdim etilgan. Sonlar uzunligi 22 lik razryad ulardan sonning belgisi birinchi razryad, tartibi 7 ta razryad mantissa 14-razryad. Tez harakatlanuvchi hisoblash mashinalari 3 ta yoki 4 ta operatsiya qo‘shilmasini 1 sekundda bajaradi, ko‘paytmasini esa 3-4 sekundda.

Fon Neyman printsipi. Fon Neyman arxitekturasi - kompyuter xotirasiga ma’lumot va dasturlarni saqlashning zamonaviy prinsipi sifatida keng tarqalgan. Hisoblash tizimining bunday turi (Fon Neyman mashinasi) atamasi bilan ataladi. Ammo bu tushunchalar bir xil emas. Umumiy holatda Fon Neyman arxitekturasi haqida gapirilganda ma’lumotlar va dasturlarni saqlash qurilmasi bilan protsessor moduli orasida farq mavjud.



1.8-chizma. Fon Neyman arxitekturasi

EHM funksional tizimlari fon Neyman nomi bilan bog‘liq. EHM tizimi o‘z ichiga arifmetik-mantiqiy qurilma (AMQ), xotira yoki esda saqlovchi qurilma(SQ), axborotni kiritish va chiqarish qurilmasi(KQ va ChQ), boshqaruv qurilmasi(BQ).

AMQ ma’lumotlar ustida arifmetik va mantiqiy operatsiyalarni bajarish uchun qo‘llaniladi(sonli,so‘zlar yoki harfli ketma-ketlik). SQ ma’lumot va dasturlarni saqlash uchun ishlatiladi. KQ ma’lumotlarni kiritish, ChQ esa EHMdan turli axborotlarni, hatto natijalarni chiqarish uchun ishlatiladi.

BQ dastur ishlaganda qolgan barcha qurilmalarni nazorat qiladi.

EHM konstruktsiyasi Neyman tomonidan o'rtaga qo'yilgan takliflarga asoslangan.

EHM qurilish printsipi. EHM ishida dasturiy boshqaruv. Dastur turli qadamlardan iborat- buyruq. Buyruq - axborotni qayta ishlash natijasida yuzaga keladigan akt. Shartli o'tish. Bu hisob protsessi davomida biror dasturga o'tish imkoniyatini beradi. Bunda natijani hisoblashdagi uzilishlarga qaramasdan hisobga olinmaydi. Saqlangan dasturlar printsipi operativ xotiraga kiritilgan ma'lumotlar bilan birga saqlanadi. EHM da axborotlarni taqdim etishda ikkilik tizimi ishlatiladi. EHM rivojlanishining dastlabki bosqichlarida tez harakatlanuvchi AU va operativ xotira bir-biriga to'g'ri kelmaydi. Operativ xotira ular o'rtasidagi muhim kompromis bo'lib xizmat qildi. Bu printsiplar xuddi bo'lishiga qaramasdan kibernetika va hisoblash texnikasi rivojiga katta hissa qo'shdi.

1.3. Zamonaviy kompyuterlar yaratish asoslari

Birinchi RISC (Reduced Instruction Set Command – qisqartirilgan to'plamli buyruqlar tizimi) kompyuterlari yaratilgandan beri yigirma yildan ortiq vaqt o'tganligiga qaramay, apparat ta'minoti texnologiyalarining hozirgi zamon holatini hisobga olib, ishlab chiqishning ba'zi asoslaridan hozir ham foydalansa bo'ladi. Agar texnologiyalarda keskin o'zgarish yuz bersa, barcha sharoitlar o'zgaradi. Shuning uchun ishlab chiquvchilar har doim kompyuter komponentlari o'rtasidagi balansga ta'sir ko'rsatishi mumkin bo'lgan ehtimoliy texnologik o'zgarishlarni hisobga olishlari kerak.

Universal protsessorlarni ishlab chiqaruvchilar RISC tamoyillariga imkon qadar amal qilishga harakat qiladilar. Ba'zi tashqi cheklashlar, masalan, boshqa mashinalar bilan moslashuvchanlik talablari sababli vaqt-vaqti bilan kompromissga borishga to'g'ri keladi, ammo bu – ko'pchilik ishlab chiquvchilar intiladigan maqsaddir. Quyida biz ularning ba'zilarini muhokama qilamiz.

Barcha oddiy komandalar bevosita apparat ta'minoti yordamida bajariladi. Ular mikrokomandalar tomonidan talqin qilinmaydi. Talqin qilish darajasining bartaraf etilishi ko'pgina komandalarning

yuqori tezlikda bajarilishini ta'minlaydi. *CISC* (Complex Instruction Set Command - to'liq to'plamli buyruqlar tizimi) tipidagi kompyuterlarda yanada murakkab komandalar keyinchalik mikrokomandalar ketma-ketligi sifatida bajariladigan bir necha qismlarga bo'linishi mumkin. Bu qo'shimcha operatsiya mashinaning ishlash tezligini kamaytiradi, ammo u kam uchraydigan komandalar uchun qo'llanishi mumkin.

Kompyuter bir nechta komandalarni bajarishni boshlashi kerak. Zamonaviy kompyuterlarda ishlab chiqarish unumdorligini ko'paytirish uchun turli xil usullardan foydalaniladi, ulardan eng asosiysi – sekundiga imkoni boricha ko'proq komandalar soniga murojaat qilish imkoniyati. 500-MIPS protsessori sekundiga 500 mln. komandani bajarishga qodir va bunda ushbu komandalarning bajarilishiga qancha vaqt ketganligi ahamiyatga ega emas (MIPS – bu Millions of Instructions Per Second - «sekundiga million komandalar»ning ingliz tilidagi qisqartmasi). Ushbu parallellik tamoyili unumdorlikni yaxshilashda asosiy o'rin tutadi. Ammo, ushbu tamoyilni qisqa vaqt oralig'ida bir necha komandalarni bajarishga imkon bo'lgan taqdirdagina amalga oshirish mumkin.

Qaysidir dasturning komandalari har doim ma'lum tartibda joylashgan bo'lsa ham, kompyuter ularni bajarishga boshqa tartibda ham kirishishi mumkin (chunki xotiraning kerakli resurslari band bo'lishi mumkin) va, bundan tashqari, ularni bajarishni ular dasturda joylashgan tartibiga teskari bo'lgan tartibda tugatishi mumkin. Albatta, agar 1-komanda registrni o'rnatsa, 2-komanda esa bu registrdan foydalansa, 2-komanda registr kerakli qiymatni yozib olmaguncha registrni o'qimasligi uchun alohida ehtiyotkorlik bilan harakat qilish kerak. Bunday xatolarga yo'l qo'ymaslik uchun ko'p miqdordagi tegishli yozuvlarni xotiraga kiritish kerak, ammo unumdorlik bir vaqtning o'zida bir nechta komandalarni bajarish imkoniyati tufayli baribir yuqori darajada bo'lib qolaveradi.

Komandalar osonlik bilan dekodlanishi kerak. Sekundiga chaqiriladigan komandalar sonining chegarasi ayrim komandalarni dekodlash jarayoniga bog'liq. Komandalarni dekodlash ular uchun qanday resurslar kerak ekanligi va qanday harakatlarni bajarish kerakligini aniqlash uchun amalga oshiriladi. Bu jarayonni soddalashtirishga yordam beradigan istalgan vositalar foydalidir.

Masalan, ma'lum uzunlikdagi va qismlari ko'p bo'lmagan komandalardan foydalaniladi. Komandalarning turli formatlari qanchalik kam bo'lsa, shunchalik yaxshi.

Xotiraga faqat yuklash va saqlash komandalari murojaat qilishi kerak. Operatsiyalarni alohida qadamlarga ajratishning eng oddiy usullaridan biri-ko'pchilik komandalar uchun operandalarning registrlardan olinishi va yana xuddi shu yerga qaytarilishini talab qilish. Operandalarni xotiradan registrlarga ko'chirish operatsiyasi turli komandalarda amalga oshirilishi mumkin. Xotiraga murojaat etish ko'p vaqt olganligi, bunday kechikish esa nomaqbul bo'lganligi sababli, bu komandalarning ishini boshqa komandalar bajarishi mumkin, agar ular registrlar va xotira o'rtasida operandalarni ko'chirishdan boshqa ishni bajarishmasa. Ush' u kuzatuvdan shunday xulosa kelib chiqadiki, xotiraga faqat yuklash va saqlash komandalari murojaat qilishi kerak (LOAD va STORE).

Registrlarning soni ko'p bo'lishi kerak. Xotiraga murojaat etish ancha sekinlik bilan amalga oshirilishi sababli, kompyutyerda registrlar soni ko'p bo'lishi kerak (kamida 32 ta). Agar so'z qachondir xotiradan chaqirilgan bo'lsa, registrlarning soni ko'p bo'lganligi uchun u kerak bo'lgunicha registrda bo'lishi kerak. So'zning registrdan xotiraga qaytarilishi va bu so'zning registrga yangidan yuklanishi nomaqbul. Ortiqcha ko'chirishlardan xalos bo'lishning eng yaxshi usuli – yetarlicha miqdordagi registrlarning bo'lishi.

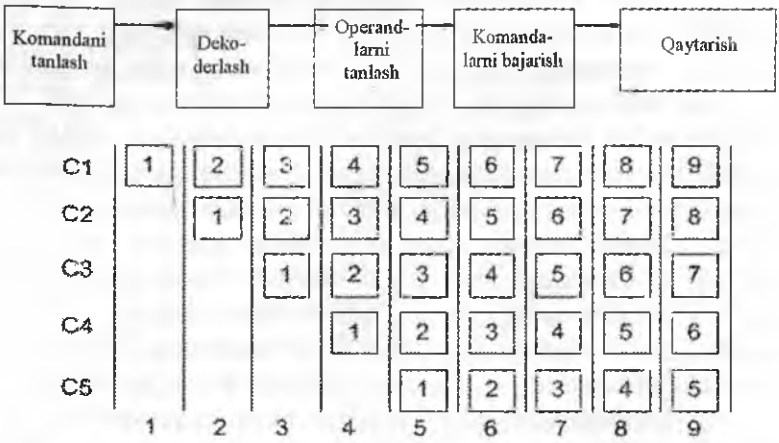
Komandalar darajasidagi parallelik.Kompyuterlarni ishlab chiquvchilar mashinalarning unumdorligini yaxshilashga intilmoqdalar. Protsessorlarni tezroq ishlashga majbur qiluvchi usullardan biri – ularning tezligini oshirish, ammo ayni vaqtda muayyan tarixiy davr bilan bog'liq texnologik cheklashlar mavjud. Shu sababli ko'pgina ishlab chiquvchilar protsessorning ushbu ishlash tezligida yaxshiroq unumdorlikka erishish uchun parallellik (bir vaqtning o'zida ikki yoki undan ortiq operatsiyalarni bajarishdan) tamoyilidan foydalanadilar.

Parallellikning ikkita asosiy shakllari mavjud: komandalar darajasidagi parallelizm va protsessorlar darajasidagi parallelizm. Birinchi holatda parallellik alohida komandalar doirasida amalga oshiriladi va sekundiga ko'p miqdordagi komandalarning bajarilishini ta'minlaydi. Ikkinchi holatda bir vaqtning o'zida bir vazifa ustida bir

nechta protsessorlar ishlaydi. Har bir yondashuv o‘z afzalliklariga ega.

Konveyerlar. Ko‘p-yillardan beri ma’lumki, komandalarni yuqori tezlik bilan bajarish yo‘lidagi asosiy to‘siq ularni xotiradan chaqirish hisoblanadi. Bu muammoni hal qilish uchun ishlab chiquvchilar komandalar kerakli vaqtda mavjud bo‘lishi uchun komandani xotiradan oldinroq chaqirish vositasini o‘ylab topdilar. Bu komandalarni oldindan tanlash buferi deb nomlanadigan registrlar to‘plamida joylashgan. Shu tariqa, muayyan komanda kerak bo‘lganda, u to‘ppa-to‘g‘ri buferdan chaqirilib, u xotiradan o‘qilishi uchun kutish kerak bo‘lmagan edi. Bu g‘oyadan 1959-yilda yaratilgan IBM Stretch kompyuterida foydalanilgan edi.

Amalda oldindan olish jarayoni komandaning bajarilishini ikki bosqichga ajratadi: chaqiruv va bajarish. Konveyer g‘oyasi ushbu strategiyani yanada ilgariroq siljitdi [7]. Endilikda komanda ikki bosqichga emas, bir necha bosqichlarga ajratildi, ularning har biri apparat ta’minotining muayyan qismi tomonidan bajarilar edi, ayni vaqtda ushbu qismlar parallel ishlay olar edilar.



1.9-chizma. 5 bosqichdan iborat konveyer (a); o‘tilgan sikllarning miqdoriga bog‘liq holda har bir bosqich holati (b). 9 ta sikl ko‘rsatilgan

1.9. *a*-chizmada bosqichlar deb nomlanadigan 5 ta blokdan iborat konveyer tasvirlangan. S1 bosqich komandani xotiradan chaqiradi va uni buferga joylaydi, bu komanda ushbu buferda kerak bo'lguncha saqlanadi. S2 bosqich ushbu komandaning turi va bu komanda ma'lum harakatlar bajaradigan operandalarning turini aniqlagan holda, ushbu komandani dekodlaydi. S3 bosqich operandalarning joylashgan joyini aniqlaydi va ularni registrlardan yoki xotiradan chaqiradi. S4 bosqich ma'lumotlar trakti orqali operandalarni o'tkazish yo'li bilan komandani bajaradi. Va nihoyat, S5 bosqich natijani qaytadan kerakli registrga yozadi.

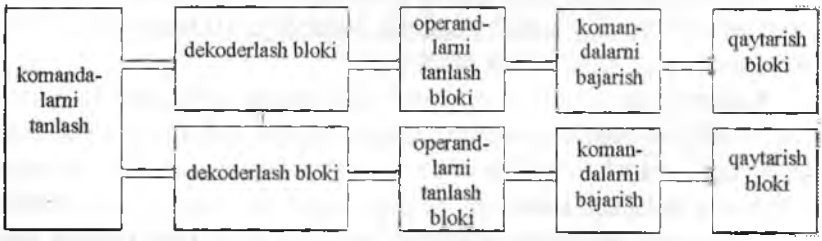
1.9. *b*-chizma biz konveyerlar vaqt davomida qanday harakat qilishini ko'rib turibmiz. 1-sikl vaqtida S1 1-komandani xotiradan chaqirib, uning ustida ishlaydi. 2-sikl vaqtida S2 bosqich 1-komandani dekodlaydi, ayni vaqtda S1 xotiradan 2-komandani chaqiradi. 3-sikl davomida S3 1-komanda uchun operandalarni chaqiradi, S2 bosqich 2-komandani dekodlaydi, S1 bosqich esa uchinchi komandani chaqiradi. 4-sikl davomida S4 1-komandani bajaradi, S3 2-komanda uchun operandalarni chaqiradi, 3-komandani dekodlaydi, S1 esa 4-komandani chaqiradi. Nihoyat, beshinchi sikl vaqtida S5 1-komandaning bajarilishini qaytadan registrga yozadi, ayni vaqtda boshqa bosqichlar keyingi komandalar ustida ishlaydi.

1.9.-chizmada tasvirlangan konveyerning unumdorligini hisoblaymiz. Bu mashinaning vaqt sikli 2 ns deb tasavvur qilamiz. U holda bitta komanda butun konveyerdan o'tishi uchun 10 ns talab qilinadi. Birinchi qarashda bunday kompyuter sekundiga 100 mln. komandani bajara oladigandek ko'rinadi, haqiqatda esa, uning ish tezligi ancha yuqori. Har bir sikl (2 ns) vaqtida bitta yangi komandaning bajarilishi tugallanadi, shuning uchun mashina sekundiga 100 mln. emas, balki 500 mln. komandani bajara oladi.

Konveyerlar kutish vaqti (bitta komandaning bajarilishi qancha vaqt oladi) va protsessorning o'tkazuvchanlik qobiliyati (protsessor sekundiga qancha million komandani bajara oladi) o'rtasida kelishuvni aniqlash imkonini beradi. Agar sikl vaqti T ns ni tashkil qilsa, konveyer esa n bosqichlardan iborat bo'lsa, u holda kutish vaqti nT ns, o'tkazuvchanlik qobiliyati esa – sekundiga 1000/T komandadan iborat bo'ladi.

Superskalyar arxitekturalar. Konveyer soni qancha ko'p bo'lsa tezlik ham oshadi. Ikkitalik konveyerga ega protsessoming sxemasi 1.10.-chizmada berilgan. Bu yerda komandalarni chaqiruvchi umumiy bo'lim xotiradan bir vaqtda ikkitadan komandani oladi va ularning har birini konveyerlardan biriga joylashtiradi. Har bir konveyer parallel operatsiyalar uchun AMQ (arifmetik mantiqiy qurilma) dan iborat. Parallel bajarilishi uchun ikkita komanda resurslar (masalan registrlar)dan foydalanilganda ziddiyatga bormasligi va ularning birortasi ham bittasini bajarish natijalariga bog'liq bo'lmasligi kerak. Bitta konveyer bilan bo'lgan holatdagidek, kompilyator nomaqbul vaziyatlar (masalan, apparat ta'minoti noto'g'ri natijalar berayotganda, agar komandalar bir-biriga mos kelmaganda) kelib chiqmasligini kuzatishi kerak yoki qo'shimcha apparat ta'minotidan foydalanish tufayli bevosita komandalarni bajarish vaqtida ziddiyatlar aniqlanadi va bartaraf qilinadi.

Avval konveyerlardan (ikkitalik hamda bittalik) faqat RISC kompyuterlarida foydalanilgan. Intel kompaniyasining protsessorlaridagi konveyerlar faqat 486-modelidan boshlab paydo bo'lgan. 486-protsessor bitta konveyerdan iborat bo'lgan edi, Pentium – esa besh bosqichli ikkita konveyerdan iborat. Shunga o'xshash sxema 1.10.-chizmada berilgan, ammo ikkinchi va uchinchi bosqichlar o'rtasida funksiyalarning ajratilishi (ular 1-dekodlash va 2-dekodlash deb nomlangan edi) biroz boshqacharoq edi. Bosh konveyer (u-konveyer) ixtiyoriy komandalarni bajara olgan. Ikkinchi konveyer (v-konveyer) faqat butun sonli oddiy komandalarni, shuningdek suzib yuruvchi nuqtali bitta oddiy komanda (FXCh)ni bajara olgan.

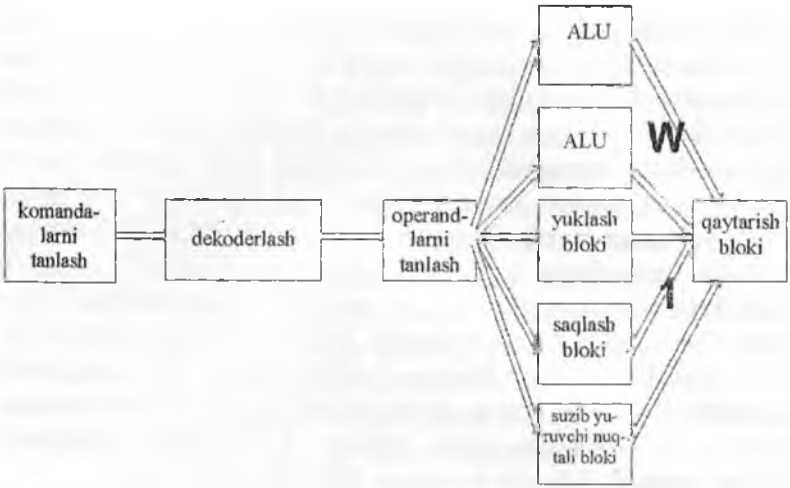


1.10-chizma. Komandalarni chaqiruvchi umumiy bo'limga ega besh bosqichdan iborat ikkitalik konveyer

Komandalar jufti parallel bajara olinishi uchun bir-biriga mos yoki mos emasligini aniqlovchi murakkab qoidalar mavjud. Agar juftlikka kiruvchi komandalar murakkab bo'lgan yoki bir-biriga mos kelmagan bo'lsa, ulardan faqat bittasi (u-konveyerdagisi) bajarilgan. Qolgan ikkinchi komanda keyingi komanda bilan juftlikni tashkil qilgan. Komandalar har doim tartib bo'yicha bajarilgan. Shu tariqa, Pentium bir-biriga mos bo'lgan komandalarni juftlikka birlashtirgan va avvalgi versiyalarga qaraganda tezroq bajariladigan dasturlarni yuzaga keltirishi mumkin bo'lgan alohida kompilyatorlarga ega bo'lgan. O'lchashlar shuni ko'rsatdiki, butun sonlar bilan operatsiyalarni bajaruvchi dasturlar Pentium kompyuterida 486-protsessorga qaraganda deyarli ikki marta tezroq bajarilgan, garchi 486-protsessorning taktli chastotasi xuddi shunday bo'lsa ham. Shubhasiz, tezlikdagi ustunlik ikkinchi konveyer tufayli paydo bo'lgan.

To'rtta konveyerga o'tishning imkoni bor, ammo katta apparat ta'minotining yaratilishini talab qilgan bo'lar edi. Buning o'rniga boshqa yondashuvdan foydalaniladi. Asosiy g'oya – 1.11.-chizmada ko'rsatilganidek, ko'p miqdordagi funksional bloklarga ega bitta konveyer. Masalan, Pentium II shunga o'xshash strukturaga ega. 1987-yilda bu yondashuvni belgilash uchun superskalyar arxitektura atamasi kiritildi. Ammo shunga o'xshash g'oya bundan 30-yil oldin CDC 6600 kompyuterida o'z aksini topdi. CDC 6600 har 100 ne ichida xotiradan komandani chaqirib, parallel bajarish uchun 10 ta funksional bloklarning biriga joylashtirar edi. Komandalar bajarilayotganda, markaziy protsessor keyingi komandani chaqirar edi.

3-bosqich 4-bosqich komandalarni bajara olish qobiliyatiga nisbatan tezroq komandalarni chiqaradi. Agar 3-bosqich har 10 ns ichida komandalarni chiqarsa, barcha funksional bloklar o'z ishlarini, shuningdek 10 ns ichida bajarganda edi, to'rtinchi bosqichda har doim faqat bitta blok ishlagan bo'lar edi, bu konveyer g'oyasining puchga chiqargan bo'lar edi. Haqiqatda to'rtinchi bosqichdagi ko'pchilik funksional bloklarga bitta sikl (bu xotiradan foydalana olish bloklari va suzib yuruvchi nuqtali operatsiyalarni bajarish bloki) egallagan vaqtdan ko'proq vaqt talab qilinadi. 1.11-chizmadan ko'rinib turganidek, to'rtinchi bosqichda bir nechta AMQ bo'lishi mumkin.



1.11-chizma. Beshta funksional bloklarga ega superskalyar protsessor

Protsessorlar darajasidagi parallelizm. Yanada yuqori tezlik bilan ishlaydigan kompyuterlarga bo'lgan talab oshmoqda. Astronomlar katta portlashdan keyingi birinchi mikrosekundda nima yuz berganligini bilishni xohlaydilar, iqtisodiyotchilar butun jahon iqtisodiyotini modellashtirishni istaydilar, o'smirlar Internet orqali o'zlarining virtual do'stlari bilan 3D interfaol o'yinlar o'ynashni xohlaydilar. Protsessorlar ishining tezligi oshib boradi, ammo ularda doimo axborotni uzatish tezligi bilan muammolar yuzaga keladi, chunki mis simlardagi elektromagnit to'lqinlarning va optik-tolali kabellardagi yorug'likning tarqalish tezligi avvalgidek 20 sm/ns bo'lib qolmoqda. Bundan tashqari, protsessor qanchalik tez ishlasa, u shunchalik tez qiziydi va uni qizib ketishdan saqlash kerak.

Parallellik komandalar darajasida qandaydir darajada yordam beradi, ammo konveyerlar va superskalyar arxitektura, odatda ishlash tezligini atigi 5-10 martaga oshiradi. Unumdorlikni 50, 100 va undan ko'pga oshirish uchun bir necha protsessorlarga ega kompyuterlarni ishlab chiqish kerak. Quyida biz bunday kompyuterlarning tuzilishi bilan tanishamiz.

Vektorli kompyuterlar. Fizik va texnik fanlardagi ko'p vazifalar vektorlarga ega, aks holda ular juda murakkab strukturaga ega

bo‘lardilar. Ko‘pincha aynan bir xil hisoblashlar ayni bir vaqtda ma’lumotlarning turli to‘plamlari ustida bajariladi. Bu dasturlar strukturasi komandalarning parallel bajarilishi tufayli ish tezligini oshirish imkonini beradi. Katta ilmiy dasturlarni tez bajarish uchun foydalaniladigan ikkita usul mavjud. Garchi ikkala sxema ko‘p tomonlama bir-biriga o‘xshash bo‘lsa-da, ulardan biri bitta protsessorning kengaytirilishi deb, ikkinchisi esa – parallel kompyuter bo‘lib hisoblanadi.

Massivli-parallel protsessor (array processor) ma’lumotlarning turli to‘plamlariga nisbatan komandalarning aynan bitta ketma-ketligini bajaradigan ko‘p sonli bir turdagi protsessorlardan iborat. Dunyodagi birinchi bunday protsessor ILLIAC IV (Illinoys universiteti) bo‘lgan edi. Dastlab har biri protsessor/xotira 8x8 elementlar panjarasidan iborat bo‘lgan to‘rtta sektorga ega mashinani konstruktsiyalash ko‘zda tutilgan edi. Har bir sektor uchun bitta nazorat bloki mavjud edi. Ushbu blok bir vaqtda barcha protsessorlar tomonidan bajariladigan komandalarni jo‘natar edi, bunda har bir protsessor o‘zining xotirasidagi o‘ziga tegishli ma’lumotlardan foydalanar edi (ma’lumotlarni yuklash initsializatsiya qilish vaqtida yuz berar edi). Narxi juda baland bo‘lganligi sababli faqat bitta shunday sektor qurilgan edi, ammo u sekundiga suzib yuruvchi nuqtali 50 mln. operatsiyalarni bajara olar edi. Agar mashinani yaratishda to‘rtta sektordan foydalanilgan bo‘lsa va u sekundiga suzib yuruvchi nuqtali 1 mlrd. operatsiyalarni bajara olganda edi, bunday mashinaning quvvati butun dunyodagi kompyuterlarning quvvatidan ikki marta ko‘proq bo‘lar edi.

Dasturchilar uchun vektorli protsessor (vector processor) massivli-parallel protsessor (array processor)ga o‘xshab ketadi. Massivli-parallel protsessor (array processor) kabi u ma’lumotlar elementlarining juftliklari ustida operatsiyalar ketma-ketligini bajarishda juda samarali. Ammo undan farqli ravishda barcha qo‘shish operatsiyalari konveyer strukturasiga ega bitta jamlash blokida bajariladi. Asoschisi Seymur Krey bo‘lgan Cray Research kompaniyasi Cray-1 (1974) modelidan boshlab va shu kungacha ko‘plab vektorli protsessorlarni chiqardi.

Protsessorlarning ikkala tipi ma’lumotlar massivlari bilan ishlaydi. Ularning ikkisi ham aynan bir xil komandalarni, masalan,

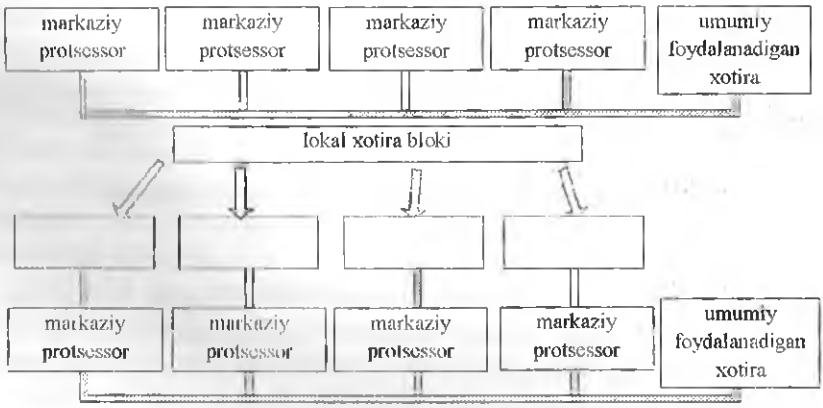
ikki vektor uchun elementlarni juftlab taxlaydigan komandalarni bajaradi. Ammo massivli-parallel protsessor (array processor)da faqat massivda qancha elementlar bo'lsa, shuncha jamlovchi qurilmalar mavjud bo'lsa, vektorli protsessor (vector processor) standart registrlar to'plamida iborat vektorli registrga ega. Bu registrlar ketma-ketlik bilan xotiradan bitta komanda yordamida yuklab olinadi. Qo'shish komandasi ikkita vektorli registrlardan ikkita vektorning elementlarini konveyer strukturasiga ega jamlovchi qurilmaga yuklab, ushbu ikki vektorlarning elementlarini juftlab taxlaydi. Natijada jamlovchi qurilmadan boshqa vektor chiqadi, bu vektor vektorli registrga joylashtiriladi yoki darhol vektorlar bilan boshqa operatsiyani bajarishda operanda sifatida foydalaniladi.

Massivli-parallel protsessorlar (array processor) haligacha chiqariladi, ammo kompyuter bozorining katta bo'lmagan qismini egallaydi, sababi ular turli ma'lumotlar to'plamlari ustida aynan bir xil hisoblashlarni bir vaqtda bajarilishini talab qiladigan vazifalarni hal qilishda samaralidir. Massivli-parallel protsessorlar (array processor) vektorli kompyuterlar (vector processor)ga nisbatan ba'zi operatsiyalarni tezroq bajara oladi, ammo ular ko'proq miqdordagi apparat ta'minotini talab qiladi va ular uchun dasturlar yozish murakkab. Vektorli protsessorlar (vector processor)ni, ikkinchi tomondan, oddiy protsessorga qo'shish mumkin. Natijada dasturning vektorli shakliga o'zgartirilishi mumkin bo'lgan qismlari vektorli blok tomonidan, dasturning qolgan qismi esa – oddiy protsessor tomonidan bajariladi.

Multiprotsessorlar. Massivli-parallel protsessor elementlari o'zaro bog'langan, sababi ularning ishini bitta boshqaruv bloki nazorat qiladi. Umumiy xotiraga birlashgan bir nechta parallel protsessorlar tizimi multiprotsessor deb nomlanadi. Har bir protsessor xotiraning istalgan qismidan axborotni yoza olishi yoki o'qishi mumkin bo'lganligi sababli qandaydir kesishishlarga yo'l qo'ymaslik uchun ularning ishi dasturiy ta'minot bilan muvofiqlashtirilishi kerak.

Bu g'oya amalga oshirilishining turli variantlari mavjud. Ularning eng oddiysi – bir nechta protsessorlarni va bitta umumiy xotirani birlashtiruvchi bitta shinaning mavjudligi. Bunday multiprotsessor sxemasi 1.12, *a*-chizmada ko'rsatilgan. Ko'pgina kompaniyalar bunday tizimlarni ishlab chiqaradilar.

Doimiy ravishda aynan bitta shina orqali xotiradan foydalana olishga urinayotgan ko'p sonli tez ishlaydigan protsessorlar mavjud bo'lganda nizolar kelib chiqishini tushunib yetish qiyin emas. Bu muammoni hal etish va kompyuterning unumdorligini oshirish uchun turli modellar ishlab chiqildi. Ulardan biri 1.13, *b*-chizmada tasvirlangan. Bunday kompyutyerda har bir protsessor boshqa protsessorlar foydalana olmaydigan o'z lokal xotirasiga ega. Bu xotiradan dasturlar va bir nechta protsessorlar o'rtasida bo'lish kerak bo'lmagan ma'lumotlar uchun foydalaniladi. Lokal xotiradan foydalanishda bosh shinadan foydalanilmaydi va shu tariqa, bu shinadagi axborot oqimi pasayadi. Muammoni hal qilishning boshqa variantlari ham mavjud (masalan, kesh-xotira).



1.13-chizma. Bitta shina va bitta umumiy xotiraga ega multiprotsessor (a); har bir protsessor uchun o'zining lokal xotirasiga ega bo'lgan multiprotsessor (b)

Multiprotsessorlar parallel kompyuterlarning boshqa turlariga nisbatan bir qator afzalliklarga ega, sababi yagona ajratilgan xotira bilan ishlash juda oson.

Multikompyuterlar. Soni ko'p bo'lmagan protsessorlarga ega multiprotsessorlarni yaratish qiyin emas, katta multiprotsessorlarni yaratish ba'zi qiyinchiliklarga ega. Murakkablik barcha protsessorlarni xotira bilan bog'lashdan iborat. Bunday muammolarga duch

kelmaslik uchun ko‘p ishlab chiquvchilar ajratilgan xotira g‘oyasidan voz kechdilar va har birining o‘z xotirasi mavjud va umumiy xotirasi bo‘lmagan ko‘p sonli o‘zaro bog‘langan kompyuterlardan iborat tizimlarni yarata boshladilar. Bunday tizimlar multikompyuterlar deb ataladi.

Multikompyuter protsessorlari bir-birlariga xabarlar yuboradilar (bu bir oz elektron pochtaga o‘xshab ketadi, faqat ancha tezroq). Har bir kompyuterni boshqa kompyuterlar bilan bog‘lash shart emas, shuning uchun, odatda, topologiyalar sifatida 2D, 3D, daraxtlar va halqalardan foydalaniladi. Xabarlar belgilangan joyga yetib borishi uchun ular bitta yoki bir nechta oraliq kompyuterlardan o‘tishi kerak. Shunga qaramay, uzatish vaqti atigi bir necha mikrosekund vaqtni oladi. Hozirgi kunda taxminan 10 000 ta protsessorlardan iborat multikompyuterlar ishga tushirilmoqda.

Multiprotsessorlarni dasturlash osonroq, multikompyuterlarni esa yaratish osonroq bo‘lganligi sababli, ikki turdagi mashinalarning afzalliklarini o‘zida birlashtirgan gibrid tizimlar yaratilmoqda.

1.4. Mikroprotsessorlarning turlari va rivojlanish bosqichlari

Birinchi MP 4004 mikroprotsessori Intel firmasi tomonidan 1971-yilda chiqarilgan. Hozirgi vaqtda bir necha yuzlab turli mikroprotsessorlar chiqarilmoqda, lekin eng ommaviy va keng tarqalgani Intel va Intel ga o‘xshash firmalarning mikroprotsessorlaridir [2-6].

Barcha mikroprotsessorlarni 3 ta guruhga bo‘lish mumkin:

– CISC asosidagi (Complex Instruction Set Command - to‘liq to‘plamli buyruqlar tizimi) mikroprotsessor;

– RISC asosidagi (Redused Instruction Set Command - qisqartirilgan to‘plamli buyruqlar tizimi) mikroprotsessor;

– MISC asosidagi (Minimum Instruction Set Command- minimal to‘plamli buyruqlar tizimi) mikroprotsessor.

IBM PC rusumidagi ko‘pchilik zamonaviy shaxsiy kompyuterlarda CISC asosdagi mikroprotsessor ishlatiladi.

I. MP 80386,80486 mikroprotsessorlarida SX, DX, SL va boshqa harfli o‘zgartirish kiritilganlari bor (80486SX, 80486DX), ular bazaviy modeldan shinalar razryadliligi, taktli chastota, ishlash

ishonchliligi, o'lchamlari, energiya iste'moli, kuchlanish amplitudasi va boshqa kattaliklar bilan farq qiladi:

– DX bazaviy model bilan deyarli mos keladi;

– SX va SL, xususan kichikroq shinalar razryadligiga ega;

– SL va ayniqsa SLE energiyani tejaydigan, ixcham ShK da (Lap Top, Notebook) ishlatishga mo'ljallangan.

80486DX – bu MP 80486 ning boshlang'ich versiyasidir. U sozlangan matematik soprotsessor va o'lchami 8 Kbayt bo'lgan birinchi darajali kesh-xotiraga ega. Uning uchun maksimal chastota – 50 MGts; chastotani yanada orttirish u vaqtda MP uchun ma'noga ega emas edi, chunki ko'pchilik tizimli platalar bunday tezliklarda ishlay olmas edilar.

486SX modeli DX ga o'xshash, lekin unda soprotsessor bloklangan. Bu ishlab chiqaruvchiga soprotsessorni testlash xarajatlaridan halos bo'lish va shu bilan mahsulot narxini kamaytirishga imkon bergan.

80486DX va undan yuqori mikroprotsessorlar ichki chastotasini ko'paytirib ishlashi mumkin. Ko'paytirilgan chastota bilan MP ning faqat ichki sxemalari ishlashi mumkin. MP ga nisbatan hamma tashqi sxemalar, shu jumladan tizimli platada joylashganlari ham, oddiy chastotada ishlaydi.

486DX4 bu 4-avlod mikroprotsessoridir (to'rt aynan shuni bildiradi); u DX2 dan ichki kesh-xotirani 16 Kbayt gacha ko'paytirilganligi, uch marta orttirilgan chastotada ishlay olish imkoniyati (486DX4 100) va 5 V m emas, balki 3,3 V kuchlanishli ta'minoti bilan farq qiladi.

Quyidagilarni ta'kidlash kerak:

• 80386 va undan yuqori MP da buyruqlar konveyerli bajariladi.

Buyruqlarning konveyerli bajarilishi – bu natijalarni MP ning bir qismidan boshqa qismiga bevosita uzatishda, MP ni turli qismlarida ketma-ket buyruqlarning turli taktlarini bir vaqtda bajarishdir. Buyruqlarning konveyerli bajarilishi ShK ning tezkorlilik bo'yicha samaradorligini 2–3 marta ortiradi;

• 80286 va undan yuqori MP ning hisoblash tarmog'ida ishlash imkoniyati;

• 80286 va undan yuqori MP ning ko'p masalalar bilan ishlash imkoniyati va bunga mos xotira himoyasi mavjud. Zamonaviy mikroprotsessorlar ikkita ish rejimiga ega:

• haqiqiy (bitta masalali), unda faqat bitta dastur bajarilishi mumkin va kompyuter asosiy xotirasining faqat 1024 Kbayti bevosita adreslanishi mumkin, qolgan (kengaytirilgan) xotiraga esa faqat maxsus drayverlar ulangandagina murojaat qilish mumkin;

• himoyalangan (ko'p masalali), bu rejimda birdaniga bir nechta dasturlarning bajarilishi, bevosita adreslash va ShK da bor bo'lgan barcha asosiy xotiraga to'g'ridan to'g'ri murojaat qilish (qo'shimcha drayverlarsiz), uning bajarilayotgan dasturlar o'rtasida avtomatik taqsimlanishi va mos ravishda uni, begona dasturlar tomonidan murojaat qilinishidan himoyalash ta'minlanadi;

• 80386 va undan yuqori MP larda virtual mashinalar tizimi rejimini qo'llab-quvvatlash.

Virtual mashinalar tizimi ko'p masalali ish rejimining yanada rivojlanishi bo'lib, unda har bir masala o'zining operatsiyaviy tizimi boshqaruvi ostida bajarilishi mumkin, ya'ni bitta MP da go'yo, parallel ishlaydigan va turli xil operatsiyaviy tizimlarga ega bo'lgan bir nechta kompyuterlar modellashtiriladi.

Pentium mikroprotsessorlari.

80586 (R5) mikroprotsessorlari Intel firmasi tomonidan patentlangan Pentium tovar markasi bo'yicha ko'proq ma'lumdir (boshqa firmalarning 80586 MP boshqacha belgilanishga ega: AMD firmasida K5, Cyrix firmasida MI va b.).

Bu mikroprotsessorlar besh pog'onali konveyerli strukturaga ega bo'lib, u ketma-ket buyruqlarning bajarilish taktlarini ko'p marotaba birgalikda ishlashini ta'minlaydi va yana boshqarishni shartli uzatish buyruqlari uchun kesh-buferga ega bo'lib, u dasturlarni tarmoqlanish yo'nalishini oldindan aytish imkonini beradi; samarali tezkorligi bo'yicha ular har bir buyruqni go'yoki bir takt ichida bajaradigan RISC MP lariga yaqinlashadi. Pentium 12razryadli adresli shinaga va 64-razryadli ma'lumotlar shinasiga egadir. Tizim bilan qiymatlarni almashish sekundiga 1 Gbayt tezlik bilan bajarilishi mumkin.

Hamma Pentium protsessorlarida har biriga 16 Kbaytdan alohida buyruqlar uchun, alohida ma'lumotlar uchun sozlangan kesh-xotira

va 2-darajali kesh-xotiraning sozlangan nazoratchisi bor, maxsuslashgan konveyerli apparatli qo'shish, ko'paytirish va bo'lish bloklari bor bo'lib, ular siljib yuradigan nuqtali amallarning bajarilishini jiddiy tezlashtiradi.

Pentium Pro mikroprotsessorlari.

1995-yil sentyabrda savdo markasi Pentium Pro bo'lgan 80686 (Rb) MP ning taqdimot marosimi bo'ldi va savdoga chiqarildi.

Mikroprotsessor 2 ta kristaldan: MP ni o'zidan va kesh-xotiradan tashkil topgan. Lekin u Pentium bilan to'liq mos kelmaydi va xususan, maxsus tizimli platani talab etadi. Pentium Pro 32-bitli ilovalarda yaxshi ishlaydi, 16-bitli ilovalarda esa hattoki Pentiumga birmuncha yutqazadi.

Yangi sxemotexnik yechimlar tufayli ular ShK lar uchun yanada yuqoriroq unumdorlikni ta'minlaydi. Bu yangiliklarning bir qismi "dinamik bajarilish" (dinamic execution) tushunchasi bilan birlashtirilishi mumkin, bu 14 ta pog'onali superkonveyerli struktura (superpi pelining), boshqarishni shartli uzatishlarda dasturning tarmoqlanishini oldindan aytish (branch prediction) va mo'ljallangan tarmoqlanish noli bo'yicha (speculative execution) buyruqlarning bajarish borligini bildiradi.

Ko'p masalalarni, ayniqsa iqtisodiy masalalarni yechish dasturlarida ko'p sonli boshqarishni shartli uzatishlar mavjud. Agar protsessor o'tish, tarmoqlanish yo'nalishini oldindan ayta olsa, u holda uning ish unumdorligi hisoblash konveyerlarini yuklashni optimallashtirish hisobiga sezilarli ortadi. Pentium Pro protsessorida oldindan to'g'ri aytish ehtimolligi 90%, Pentium da esa 80%.

256 - 512 Kbayt sig'imli kesh-xotira - Pentium protsessorlaridagi yuqori unumli tizimlarning majburiy xususiyatidir. Lekin ularda sozlangan kesh-xotira katta bo'lmagan (16 Kbayt) sig'imga ega, uning asosiy qismi esa protsessordan tashqarida asosiy platada joylashadi. Shuning uchun u bilan ma'lumotlar almashish MP ning ichki chastotasida emas, balki odatda 2-3 marta past bo'lgan taktli generator chastotasida amalga oshiriladi, bu esa kompyuterning umumiy tezkorligini pasaytiradi. Pentium Pro MP da 1-darajali kesh-xotira ham (8 Kbayt dan buyruqlar va qiymatlar uchun) va 256 yoki 512 Kbayt sig'imli 2-darajali kristall kesh-xotira ham bor bo'lib, ular

mikroprotsessorning o‘zini platasida joylashgan va MP ning ichki chastotasida ishlaydi.

Pentium MMXva Pentium II mikroprotsessorlari.

1997 yilda multimedia texnologiyasida ishlash uchun modernizatsiya qilingan va mos ravishda Pentium MMX (MMX - Multi Media eXtention) va Pentium II savdo markalarini olgan Pentium Pro mikroprotsessorlarining taqdimot marosimi bo‘ldi.

Pentium MMX MP audio va Video ma’lumotlarni qayta ishlashga mo‘ljallangan qo‘shimcha 57 ta buyruq, ikki marta kattalashgan (32 Kbayt gacha) kesh-xotira, Pentium Pro MP dan olingan tarmoqlanishlarni oldindan aytish yangi blokini o‘z ichiga oladi. Shuning hisobiga unda Pentium MP ga nisbatan 1 millionta tranzistorli element ko‘proqdir.

Bu mikroprotsessorlarni samarali ishlatish uchun barcha eski dasturlarga (shu jumladan Windows 95, Windows NT operatsion tizimlariga ham) moslashtiruvchi dasturli lavhalarni qo‘shish kerak; aslida esa, ularsiz ham Pentium MMX MP oddiy Pentium MP dan birmuncha unumliroqdir. Pentium MMX MP oddiy ilovalarni bajarishda Pentium MP ga qaraganda 10-15% tezkorroqdir, yangi 57 ta buyruqni ishlatib multimedia ilovalarini bajarishda esa u 30% tezkorroqdir

Pentium II MP boshqa hamma MP larga nisbatan o‘zgacha tuzilishga ega, xususan, u uncha katta bo‘lmagan plata-kartridj ko‘rinishida bajarilgan bo‘lib, unga protsessorning o‘zi (Pentium Pro da 5,5 mln ta tranzistor bo‘lsa, unda 7,5 mln ta tranzistor bor) va umumiy hajmi 512 Kbayt bo‘lgan ikkinchi darajali kesh-xotiraning to‘rtta mikrosxemasi joylashtirilgan. Protsessorning o‘z mikrosxemasida joylashgan 1-darajali kesh-xotira Pentium Pro MP da bor bo‘lgan 16 Kbayt o‘rniga 32 Kbayt sig‘imga ega, lekin 2-darajali kesh-xotira MP ning ichki chastotasida emas, balki ikki marta kichik chastotada ishlaydi.

Pentium II MP 0,35 mikronli texnologiya asosida ishlab chiqariladi va 2,8 V ta’minot kuchlanishini ishlatadi. Uning uchun, tabiiyki, boshqa barcha Pentium larga nisbatan o‘zgacha tizimli plata talab etiladi.

Pentium Pro (P6) protsessorlari. Bu protsessorda komandaning dinamik bajarilishi qo‘llanilgan, ya’ni ko‘pgina ma’lumotlarning

o‘tish taxlili va virtual bajarilishlarni taxmin qilish vositalari kombinatsiyalari, bunda protsessorda komandalar dastur kodida ko‘rsatilgan tartibda bajarilmasligi mumkin. Bunda oldingi operatsiyalar natijalariga bog‘liq bo‘lmagan komandalar o‘zgartirilgan tartibda bajarilishi mumkin, ammo natijalarni xotiraga yozish va portlar ketma-ketligi dastlabki dastur kodiga mos bo‘ladi. Komandani avvaldan bajarish (spekulyativ bajarish) imkoniyati, komanda bir konveyerda boshqasiga qaraganda tezroq bajarilgan holatda komandani qayta tartiblash, dinamik bajarilishda o‘tishlarni oldindan tahmin qilish hisoblashlar samaradorligini oshiradi.

Protsessor funksional qo‘shimchalar FRC yordamida testlash rejimini qo‘llaydi. Protsessor arxitekturasi simmetrik multipleksorli tizimda to‘rttagacha protsessorni bir shinada birlashtirishga imkon beradi. Umumiy o‘tkazish qobiliyati ikkita mustaqil shina hisobiga oshirilgan. Bir shina (tizim) protsessor yadrosini asosiy xotira va interfeys qurilmalar bilan o‘zaro ishlash funksiyasini bajaradi, ikkinchisi kesh-xotira bilan ma’lumot almashuv funksiyasini bajaradi. Birinchi shina protsessor takt chastotasida ishlaydi, ikkinchisi tizim chastotasida. Shinaning bunday taqsimlanishi xotira bilan protsessor o‘rtasidagi almashinuvni uch marta tezlashtiradi. Buning natijasida alohida kesh-xotira zarurati yo‘qoladi. Mikroprotsessor ma’lumotlar va komandalar uchun har biri 8 Kbayt bo‘lingan birinchi sath kesh-xotirasi (L1) va birlashtirilgan ikkinchi sath kesh-xotirasi (L2) ni o‘z ichiga oladi. Birinchi sath ma’lumotlar kesh-xotirasi bir taktda bitta yozish operatsiyasi va bitta o‘qish operatsiyasini qo‘llaydi. Ikkinchi sath kesh-xotira interfeysi protsessor takt chastotasida ishlaydi va bir taktda 64 bit uzatishi mumkin. Ikkinchi sath kesh-xotira hajmi 256 Kbayt (0,6 mkm texnologiyasida) va 512 Kbayt (0,35 mkm texnologiyasida).

Pentium MMX (P55C) protsessorlari – multimedia, 2D- va 3D-grafik va kommunikatsion qo‘llanishlarga yo‘naltirilgan protsessorlar. Pentium MMX arxitekturasiga qo‘shimcha ravishda quyidagilar kiritilgan:

- 8 ta 64-razryadli MMX-registrlari;
- ma’lumotlarning to‘rtta yangi toifasi;

• 57 ta qo‘shimcha komanda, quyidagi guruhlarga bo‘linadi: MMX registrlari va butun sonli registrlar yoki xotira orasida ma’lumot almashinuvi, arifmetik (murakkab va turli rejimlarda hisoblash, ko‘paytirish, ko‘paytirish va bo‘lish kombinatsiyalari), qiyoslash, formatlarni o‘zgartirish, mantiqiy, siljish (mantiqiy va arifmetik), MMX registrlarini tozalash;

• komanda kesh-xotirasi va ma’lumotlar kesh-xotirasi hajmi ikki marta oshirilgan (16 Kbayt);

• o‘tishlarni taxmin qilish logikasi yaxshilangan;

• kengaytirilgan konveyerlash;

• xotirani chuqurroq buferizatsiya qilish.

Pentium MMX protsessorida arifmetik va mantiqiy operatsiyalar 64-razryadli MMX registrida joylashgan so‘z yoki ikkilik so‘z har bir bayti ustida parallel bajariladi. Bunday holatda qayta ishlashning SIMD (Single Instruction – Multiple Data) modeli yaratiladi. SIMD komandalarda 64 razryad bir vaqtda qayta ishlanadi. MMX komandaga bog‘liq holda, sakkizta bir baytli operand, to‘rtta ikki baytli, ikkita to‘rt baytli yoki bitta sakkiz baytli operand traktlanadi. SIMD qayta ishlash multimedia algoritmlari bajarilishini tezlashtiradi, bu bir turli ma’lumotli katta massivlar ustida identifik operatsiyalarni bajarishga tegishli. MMX komandalardan foydalanish multimedia operatsiyalarining bajarilish tezligini xuddi shunday chastotali birinchi avlod Pentium protsessorlariga qaraganda 60 % ga oshirish imkonini beradi. Boshqa komandalarda Pentium bilan muvofiqlik ta’minlanadi. MMX komandalari bayroqlarga ta’sir ko‘rsatmaydi, uzilishlarni yuzaga keltirmaydi va protsessor ishining ixtiyoriy rejimi uchun ochiq.

O‘tishlarni taxmin qilish logikasini mukammallashtirish komandalarni oldindan tanlash (BVO) buferlari sonini oshirish bilan ta’minlanadi. Pentium MMX to‘rtta 16-razryadli BVO ga ega.

Kengaytirilgan konveyerlash konveyerda butun sonli dasturlarning bajarilish bosqichini oltitagacha oshirish bilan erishiladi. Bu dastlabki tanlov bosqichi (PF) va komandani dekodlash (D1) bosqichi orasiga tanlov bosqichi (F) ni kiritish hisobiga amalga oshiriladi. Tanlov bosqichida komanda uzunligini dekodlash bajariladi. Bu Pentiumning dastlabki modellarida D1 bosqichida bajarilgan.

Pentium II protsessori 36-razryadli adreslar shinasiga ega, bu 64 Gbayt hajmdagi fizik xotirani adreslash imkonini beradi. Protsessor yadrosining chastotasi 233, 266, 300 va 450 MGts, shina chastotasi 66,66 va 100 MGts ni tashkil etadi. Protsessorning yuqori samaradorligiga unda komandalarning dinamik bajarilishi, MMX kengaytmasi va ikkita mustaqil shinalarni qo'llash bilan erishilgan.

Birinchi sath kesh-xotirasi (L1) 32 Kbayt gacha oshirilgan (16 Kbayt – komanda kesh-xotirasi, 16 Kbayt – ma'lumot kesh-xotirasi). Ikkinchi sath kesh-xotirasi (L2) protsessor bilan bitta plataga joylashtirilgan.

Pentium III protsessori SSE (Streaming SIMD Extensions) texnologiyasi bo'yicha qurilgan. Pentium III da 70 ta yangi SIMD komandalari qo'llanilgan, bu komandalar 128-razryadli registrlar XMM0-XMM7 bilan ishlaydi. Bunday holatda ikkita registr ustida operatsiya bajarilganda, SSE to'rt juft son ishlatadi. Buning natijasida protsessor bir vaqtda to'rttagacha operatsiyani bajarishi mumkin, bu quyidagi qo'llanishlarda samara beradi:

- uch o'lchamli grafika va suzuvchi nuqtali format hisoblashlaridan foydalanib modellashtirish;
- signallarni qayta ishlash va jarayonlarni parametrlarning keng diapazonli o'zgarishi bilan modellashtirish;
- real vaqt dasturlarida uch o'lchamli tasvirlarni generatsiya qilish;
- Video signallarni kodlash va dekodlashda blokli qayta ishlash algoritmlarida;
- ma'lumotlar potoki bilan ishlovchi raqamli filtrlash sonli algoritmlari.

Ko'rib chiqilgan Intel firmasi protsessorlaridan tashqari Pentium bilan bir xil samaradorlikka ega bo'lgan boshqa kompaniyalarning protsessorlari mavjud – AMD (AMD K5 PR 75/90/100, AMD K6), CYRIX (Cx86 (M1), CYRIX 6x86, P120+, P133+, P150+, P166+, P200+, CYRIX 6x86MX, CYRIX MediaGX), Hewlett Packard (Merced (P7), PA RISC, PA-8000), DEC (Alpha 21068, 21164, 21264).

AMD K5 PR75/90/100/120/133/166 protsessorlari CISC/RISC gibrid arxitekturasi bo'yicha qurilgan va parallel ishlaydigan beshta qayta ishlash bloklari murakkab konveyeriga ega. Pentium dan farqli ravishda bu protsessorlarda bir vaqtda suzuvchi nuqtali, zagruzka (saqlash) va o'tish komandalari bajarilishi mumkin. Registrlarni qayta nomlashning katta to'plami va imkoniyati va zagruzka (saqlash) blokining mavjudligi ikkita operatsiyani xotiradan tanlovning bir tsiklida bajarish imkonini beradi. Protsessorda ajralish va komanda bajarilish tartibini o'zgarishini taxmin qilishdan foydalaniladi. Komandalar kesh-xotirasi hajmi 16 Kbayt; ma'lumotlar kesh-xotirasi hajmi 8 Kbaytdan iborat.

AMD-K6 MMX protsessorlari 32 Kbayt hajmdagi ma'lumotlar va komandalar uchun ajratilgan kesh-xotiraga ega. Ma'lumotlar kesh-xotirasi teskari yozishni qo'llaydi. Komanda kesh-xotirasi komandalarni dastlabki dekodlash uchun qo'shimcha sohaga ega. Har bir komanda qayta kodlash bitlariga ega, ular kesh-xotiradagi navbatdagi komanda boshiga siljishni ko'rsatadi. Ikkinchi sath ichki kesh-xotira mavjud emas. Protsessor 8192 elementdan iborat ajralishlar tarixi jadvalidan foydalanib, ajralishlarni taxmin qilish logikasini qo'llaydi; o'tish adreslari kesh-xotirasi va qaytish steki o'tishni to'g'ri taxmin qilish ehtimolligini 95% dan oshishiga imkon beradi.

Intel P54 vaP55 protsessorlaridan farqli ravishda AMD-K6 MMX protsessori multiprotsessor tizimini qo'llovchi ichki vositalarga ega emas. Unda shina operatsiyalarini tekshiruvchi (BUSChK) signal va zondlangan rejim yo'q, shuningdek, to'xtash nuqtasi signali va samaradorlik monitoringi chiqarilmaydi. Protsessorda komanda ketma-ketliklarini o'zgarishi, ma'lumotlarni dastlabki uzatish, registrlarni qayta nomlash mavjud bo'lgan spekulyativ bajarishdan foydalaniladi.

3DNow texnologiyali AMD-K6-2 protsessorlari 0,25 mkm texnologiya asosida qurilgan. Ular 256 Kbayt hajmli ikkinchi sath ichki kesh-xotirasining va 64 Kbayt hajmli birinchi sath kesh-xotirasining tezkorligi xarakterlanadi. Protsessor effektiv RISC-arxitekturaga va suzuvchi nuqtali hisoblashlarning mukammallashtirilgan blokiga ega. Protsessor yadrosi chastotasi 300 dan 400 MGts gacha. Shina chastotasi – 100 MGts.

Cyrix 6x86 (M1) protsessorlari registrlarni dinamik qayta nomlash, komandalar bajarilish tartibini o'zgartirish, spekulyativ bajarish, o'tishlarni taxmin qilish imkoniyatlariga ega. Protsessorlar ikkita kesh-xotirani o'z ichiga oladi: komandalar va ma'lumotlar uchun foydalaniladigan 16 Kbayt hajmli kesh-xotira va qo'shimcha 256 baytli komandalar kesh-xotirasi. Komandalar uchun alohida kesh-xotira umumiy kesh-xotiraga komandalar va ma'lumotlar uchun murojaat bo'lganda yuzaga kelishi mumkin bo'lgan konfliktlarni bartaraf qilish imkonini beradi. Protsessor butun sonli komandalar va suzuvchi nuqtali komandalarni bir vaqtda bajarish qobiliyatiga ega. Komandalar bajarilish sikli bosqichlari:

- komandalarni tanlash F;
- komandani dekodlash (1 stadiya) D1;
- komandani dekodlash (2 stadiya) D2;
- adresni hisoblash (1 stadiya) ACC1;
- adresni hisoblash (2 stadiya) ACC2;
- bajarilish EX;
- natijani yozish WB.

Dekodlash va adresni hisoblash bosqichlari konveyerlangan, oqibatda EX va WB bosqichlarida komandalarni qayta tartiblash imkoniyati ta'minlangan.

Mobil foydalanish uchun protsessorlarning kichik voltli versiyalari mavjud.

Cyrix 6x86MX protsessori - M1 protsessorining mukammallashtirilgan varianti, qo'shimcha ravishda 57 ta multimedia komandalari to'plamini bajarish, MMX kengaytmasi bilan muvofiqlik imkoniyati kiritilgan va takt chastotasi oshirilgan. Protsessor ikkita kesh-xotirani o'z ichiga oladi: teskari yozishli 64 Kbayt hajmdagi 4-kirishli assotsiativ kesh-xotira va 256 bayt hajmdagi yuqori tezlikli komandalar kesh-xotirasi. Mobil qo'llanilish uchun energiya iste'moli effektiv boshqaruv tizimi ko'rib chiqilgan.

6-razryadli mikroprotsessor arxitekturalarining o'ziga xosligi.

1997-yilda Intel va Hewlett Packard kompaniyalari yangi mikroprotsessor arxitekturasi EPIC (Explicitly Parallel Instruction Computing – aniq parallel hisoblash qo'llanmasi) ni yaratdilar, bu 64-

razryadli mikroprotsessorlar IA-64, McKinley, Itanium, Itanium 2 lar asosida ishlab chiqilgan.

EPICarxitekturasining o'ziga xosligi quyidagilar:

• umumiy vazifali registrlarning katta soni. MP IA-64 butun sonli operatsiyalar uchun 128 ta 64-razryadli registrlar, kasr sonlar uchun 12880 ta registrlarni o'z ichiga oladi;

• komandalararo bog'liqliklar qidiruvi, bunda qidiruv protsessor bilan emas, kompilyator bilan bajariladi. MP IA-64 komandalari kompilyator tomonidan 128 bit uzunlikdagi «bog'lanishda» guruhlanadi. Bog'liqlik 3 ta komanda va komandalar orasidagi (ya'ni, k2 komandasi k1 komandasi bilan parallel ishga tushishi mumkinligi yoki k2 faqatgina k1 dan keyin bajarilishi kerakligi aniqlanadi), shuningdek boshqa bog'liqliklar orasidagi (c1 bog'liqlikdagi k3 komandasi s2 bog'liqlikdagi k4 komandasi bilan parallel ishlashi mumkinligi) bog'liqlik ko'rsatilgan shablonni o'z ichiga oladi;

• arxitekturaning masshtabliligi, ya'ni komandalar to'plamining ko'p sonli funksional qurilmalarda ishlatila olinishi. Masalan, uchta komandadan iborat bitta bog'liqlik protsessorning uchta funksional qurilmalar to'plamiga mos keladi. IA-64 protsessorlari bunday funksional qurilmalari soni turlicha bo'lishi mumkin, bunda dastur kodiga moslik saqlanadi;

• predikatsiya (Predication). Predikatsiya bu shartli ajralishlarni qayta ishlash usuli hisoblanadi. Shartli shoxlanishda turli shoxdagi komandalar predikat maydonlar (shart maydoni) bilan belgilanadi va parallel bajariladi, ammo ularning natijalari predikat registrlarning qiymatlari aniqlanmaguncha yozilmaydi. Agar sikl oxirida shoxlanish sharti aniqlansa, predikat registrlarning biri «to'g'ri» shoxga mos bo'ladi va unga bir o'rnatiladi, ikkinchisiga ega nol. Yozuvlarni kiritishdan oldin protsessor predikat maydonini tekshiradi va faqatgina predikat maydoni bir bo'lgan komandalar natijalarinigina yozadi;

• taxmin bo'yicha yuklash (Speculative loading). Bu mexanizm protsessorning nisbatan sekin ishlovchi asosiy xotiradan komandalarni yuklashni kutishi bilan bog'liq bo'sh turishlarini kamaytirish uchun mo'ljallangan. Kompilyator komandalarni asosiy xotiradan

ular tezroq bajarilishlari uchun yuklab qo'yadi. Bunda MP uchun xotiradan qandaydir komanda kerak bo'lsa, kutib qolmaydi.

0,18 mkm texnologiya bo'yicha qurilgan Itanium 2 protsessori bitta mashina siklida oltita komandani bajarish imkoniyatiga ega. Takt chastotasining va tizim shinasi o'tkazish qobiliyatining oshirilishi (6,4 Gb/s, shina chastotasi - 400 MGts, shina razryadi-128) birinchi Itanium ga qaraganda 1,5-2 marta katta samaradorlikni ta'minlaydi. Protsessor kristalda qurilgan va yadro chastotasida ishlaydigan katta hajmli (3 Mbayt) uchinchi sath kesh-xotirasiga ega.

2. MIKROPROTSESSORLI TIZIMLAR

2.1. Mikroprotsessorli tizim strukturalari va ishlash asoslari

Mikroprotsessor tizimlari asosini tashkil qiladigan asosiy tushunchalar quyidagilardan iborat [11-12]:

Elektron tizim – har qanday elektron uzel bo'lib, ma'lumotlarni qayta ishlovchi blok, jihoz yoki kompleksdir.

Masala – elektron tizimga bog'liq bo'lgan funksiyalar to'plami.

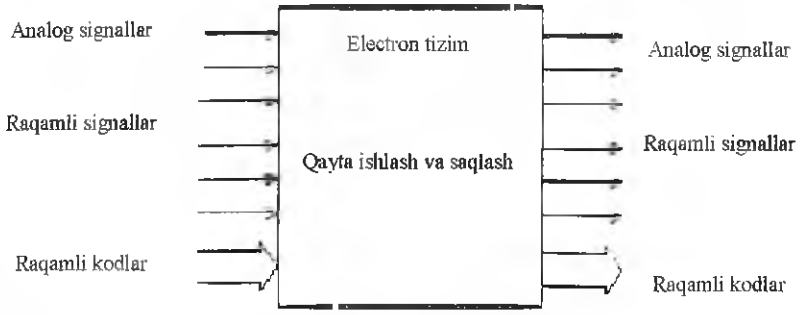
Tezkorlik – elektron tizim funksiyalarining bajarilish tezligi ko'rsatkichi.

Moslashuvchanlik – Tizimning turli masalalarga moslashuvchanligi.

Interfeys – mantiqiy va konstruktiv qurilmalararo ma'lumot almashish moslashmasi.

Mikroprotsessor elektron tizimning bir qismi bo'lib, kiritish va chiqarish signallarini qayta ishlash qurilmasi sifatida qabul qilingan (2.1.- chizma).

Kiritish va chiqarish signallari sifatida analog signallar, raqamli signallar, raqamli kodlar, raqamli kodlar ketma-ketligi qabul qilingan.



2.1-chizma. Elektron tizim

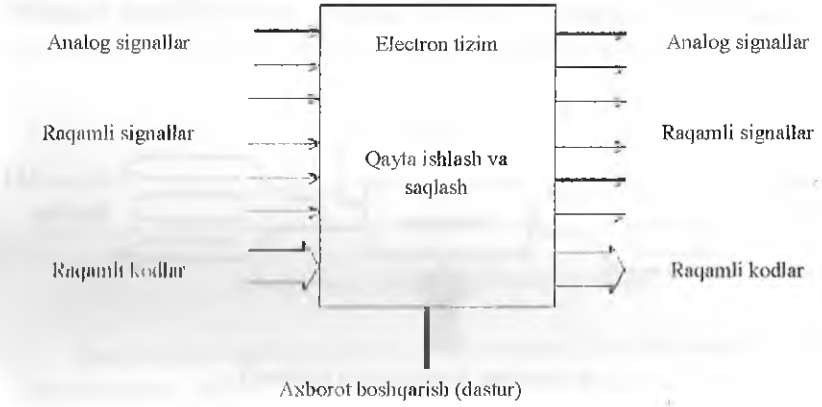
Tizim ichida ma'lumotlar yoki signallar saqlanadi. Agar tizim raqamli bo'lsa, analog signallar analog – raqam o'zgartirish qurilmalari asosida raqamli signallarga o'zgartiriladi yoki aksi.

Ma'lumotlarni saqlash va qayta ishlash raqamli ko'rinishda bo'ladi. Ma'lumotlarni saqlash va qayta ishlash sxemotexnik tizimlarga uzviy bog'liq.

Har qanday tizim maxsus bir vazifani yechish uchun mo'ljallangan bo'ladi.

Maxsus tizim har bir elementi to'liq ishlaydi. Maxsus tizim maksimal tezkorlikni ta'minlaydi. Eng asosiy kamchiligi, har bir vazifa uchun qaytadan loyihalash va tayyorlash kerak. Ushbu masalani hal qilish uchun shunday tizim qurish kerakki, ushbu tizimda qurilmalarni har doim o'zgartiravermaslik kerak.

Dasturiy boshqariladigan tizim ushbu masalani hal qiladi.U larni mikroprotsessor tizimlari ta'minlaydi (2.2.-chizma).



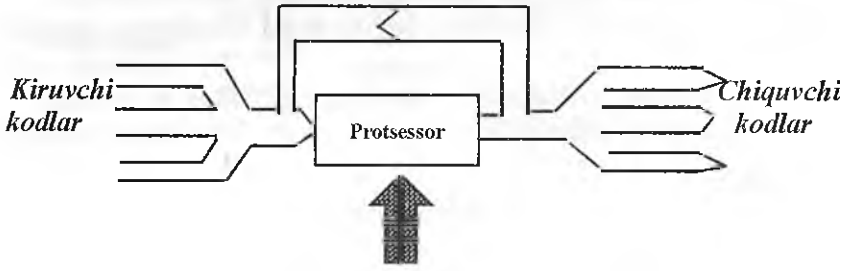
2.2-chizma. Dasturiy boshqarish tizimi

Universallik bir qancha muammolarni keltirib chiqaradi. Maksimal murakkab vazifalarni yechimini topish oddiy vazifalar yechimini topishga qaraganda ko'p vositalarni talab qiladi. Shuning uchun universal tizim shunday bo'lishi kerakki, murakkab vazifalarni yechishda barcha vositalardan maksimal foydalanish, oddiy vazifalarni yechishda esa kerakli vositalardan foydalanish kerak.

2.2. Mikroprotsessorning umumiy strukturasi

Har qanday mikroprotsessorli tizimlarning yadrosi mikroprotsessor yoki protsessor (processor) hisoblanadi. Boshqacha qilib «qayta ishlagich», aynan mikroprotsessor - bu shunday uzelki, hamma ma'lumotlarni qayta ishlashni MPT ichida amalga oshiradi. Qolgan uzellari yordamchi funksiyalarni bajaradi: ma'lumotlarni saqlash, tashqi qurilmalar bilan aloqa, foydalanuvchi bilan muloqot. MP arifmetik va mantiqiy amallarni, kodlarni vaqtinchalik saqlash, MPT lari uzellararo ma'lumot almashish va boshqa vazifalarni bajaradi. Protsessorni tizim miyasi bilan solishtirsa bo'ladi.

Protsessor hamma operatsiyalarni ketma-ket bajaradi. Birinchi bo'limdan ma'lumki MP operatsiyalarni parallel bajarishi mumkin. Ma'lumotlarni ketma-ket bajarish afzalligi shundan iboratki bir takt jarayon ichida murakkab operatsiyalarni bajarish mumkin. Lekin operatsiyalarni bajarilishi ularning oson yoki murakkabligiga bog'liq. Bundan kelib chiqadiki, MP har qanday operatsiyalarni bajaradi, lekin hamma operatsiyalarni yagona uzeldan o'tkazadi (2.3.-chizma).

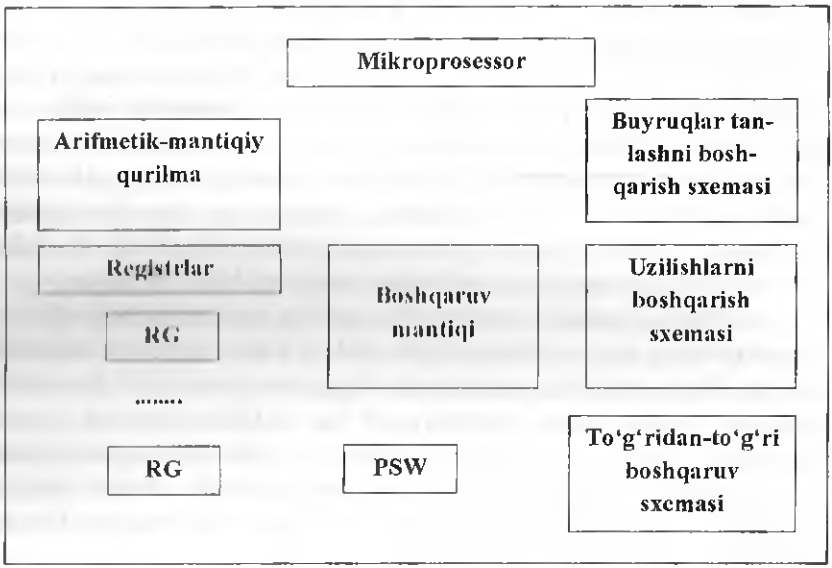


2.3-chizma. MP da axborotlar oqimi

Protsessor asosini buyruqlar tizimi tashkil qiladi. Buyruqlar tizimi va strukturasi protsessorning tezligi, moslanuvchanligi, foydalanish uchun qulayligini belgilaydi.

MP buyruqlari o'ntadan bir necha yuztagacha bo'lishi mumkin. Buyruq kodlari turli razryad uzunliklariga ega bo'ladi. Har bir buyruqning o'z bajarilish vaqti bor, shuning uchun dastur bajarilish

vaqti faqat dasturga emas, balki qanday buyruqlar bajarilishiga bog'liq. Protsessor strukturasiga: registrlar, arifmetik mantiqiy qurilmalar, multipleksorlar, buferlar va registrlar kiradi. Hamma uzellar ish jarayoni protsessorni umumiy tashqi takt signali orqali amalga oshiriladi. Protsessor murakkab raqamli qurilmani tashkil qiladi (2.4.-chizma).



2.4-chizma. Oddiy MP ning tuzilishi

Buyruqlarni tanlashni boshqarish sxemasi buyruqlarni o'qiydi va deshifratsiya qiladi. Birinchi mikroprotsessorlarda bir vaqtning o'zida oldingi buyruqni o'qish hamda keyingi buyruqni tanlash imkoniyatlari mavjud emas edi, chunki protsessor bu operatsiyalarni o'z ichiga olmagan edi. Lekin hozirgi 16-razryadli protsessorlarda bu imkoniyat mavjuddir, unda konveyer (ketma- ketlik) nomli buyruq mavjud bo'lib, bu buyruq orqali bir buyruq bajarilish jarayonida keyingi buyruqlarni tanlash, amalga oshirish imkoniyati mavjuddir. Bu orqali esa ish jarayoni tezlashadi. Konveyer o'z ichiga uncha katta bo'lmagan protsessor xotirasi (tashqi shinani ozod qilish bilan)ni o'z ichiga oladi, ya'ni qisqa harakat orqali bu xotiraga bir qancha

buyruqlarni yozib olishi mumkin. Bu buyruqlar konveyer kabi protsessordan o‘z holati bo‘yicha o‘qiydi. Lekin, bajarilayotgan buyruq xotira yacheykasiga o‘tadi, xotiradagi ko‘plab qolib ketayotgan (navbatda) buyruqni rad etadi. Lekin bu buyruqlar dasturlarda kam uchraydi.

Konveyerning takomillashuvi tufayli unga kesh-xotiraning qo‘shilishi bo‘ldi. Bu bilan protsessor buyruqlarni bajarish jarayonida, keyingi bajarilishi lozim bo‘lgan buyruqlarni o‘z ichida saqlaydi. Kesh- xotira qanchalik katta bo‘lsa, protsessorning keyingi o‘tish jarayonida navbatda turgan buyruqlarni saqlanish imkoniyati bor. Kesh xotirada protsessorning ayni damda bajarilishi lozim bo‘lgan buyruqlar saqlanadi. Buyruqlarni yanada tezroq qayta ishlash uchun zamonaviy protsessorlarda tanlash va deshifratsiyalash majmuasi, parallel buyruqlar konveyeri holati ishlatiladi. Bu bilan buyruqlarning o‘tish davri va boshqa usullarini bilish mumkin.

Arifmetik-mantiqiy qurilma. Bu qurilma protsessor buyruqlarini bajarilishidagi ma’lumotlarni qayta ishlash kabi vazifalarni bajaradi. Qayta ishlash misolida mantiqiy operatsiyalarni misol qilib ko‘rsatish mumkin (Misol uchun: mantiqiy «VA», «YOKI» kabilar), hamda operandlar ustida bitli operatsiyalar va arifmetik operatsiyalar (ko‘paytirish, bo‘lish kabilar) shular jumlasidandir. Bajarilayotgan buyruq turi qaysi kodlar bilan operatsiya bajarilishi hamda ularning natijalarini o‘z ichiga oladi.

Ma’lumotlarni qayta ishlash ketma- ketligi tartibi – arifmetik yoki mantiqiy funksiyalar bilan tanishib chiqamiz. Ko‘p hollarda bu hol juft operandlar bilan ishlaydi, ya’ni bajaruvchi operand dest (destination) va operand manba src(source). Bu yo‘riqnomaning odatiy ishlash sxemasi quyidagicha:dest=F(dest,serc), bu yerda, F- ikkita o‘zgaruvchidan bir nechta funksiyalardir. Bu esa protsessorning yo‘riqnomani (registr, xotira, konstanta yo‘riqnomada) bajarishidagi yuqoridagi ko‘rsatkichlardan ikkilik soniga o‘zgartiradi va ular ustida bajarilgan natijalarni dest (destination) qismidan biriga yozib qo‘yadi.Yana xuddi shu funksiyani boshqa bir juftli son uchtadan operatsiya bajarishi uchun boshqa bir operandlar juftligi kerak bo‘ladi. AMQ (Arifmetik mantiqiy qurilma) ning tezkorligi protsessorning ishchanligini ko‘rsatib beradi. Faqatgina AMQ ning takt chastotaning taktli signali muhim bo‘lmasdan, taktning soni ham

u yoki bu buyruqni bajarish uchun muhimdir. Ishchanlikni oshirish maqsadida ishlab chiqaruvchilar buyruqni bajarish vaqtini bir taktga tenglashtiradilar, shu bilan AQM ning yuqori chastota ishlashini ta'minlaydi.Buni amalga oshirishning yo'llaridan biri, AMQ dagi bajarilishi mumkin bo'lgan buyruqlar sonini kamaytirishdan va protsessorlar yaratilishida uning tarkibida mavjud bo'lishi kerak bo'lgan buyruqlar sonini kamaytirishdan iboratdir (RISC – protsessorlari). Yana bir boshqa yo'li bir vaqtda buyruqlarni bajaruvchi AMQ lardir.

Maxsus murakkab o'zgaruvchan operatsiyalar uchun esa protsessor tizimlarida oddiy buyruqlar va maxsus ichki dasturlar bilan dasturlangan, lekin keyinchalik maxsus hisoblovchi qismlar yaratildi. Matematik soprotsessorlar, ya'ni vaqti bilan shu soprotsessorlarga almashtirish mumkin. Zamonaviy mikroprotsessorlarda matematik soprotsessorlar mikroprotsessorning tarkibiy qismiga kiradi.

Protsessor registrlari – bu tezkor xotira va vaqtinchalik turli xil kodlarni saqlash uchun, ya'ni ma'lumotlar, manzillar va ishchi kodlardir. Bu kodlar orqali bajarilayotgan operatsiya sezilarli darajada protsessorda tez bajariladi, umumiy holda protsessor tarkibida bunday registrlarning mavjudligi ijobiy natijalarga olib keladi. Tez harakatli protsessor registr razryadiga uzviy bog'liqdir. AMQ da razryadli registrlar tashqi razryadlar bilan mos kelmasligi mumkin.

Bajarish funksiyasi bo'yicha ichki registrlar ikki turga bo'linadi. Birinchisi, Intel kompaniyasining registrlari, bu turdagi registrlar aniq holdagi tarkibiy javoblarni mujassamlashtiradi. Bir tomondan bunday funksiyalar bu turdagi registrlarni ishlatayotgan korxonalar uchun ish ko'lamini yengillashtiradi, buyruqlar bajarilish vaqti qisqaradi. Boshqa tomondan esa, bu registrlar protsessorning barqarorligini pasaytirib, dasturning ishlashini sekinlashtiradi. Masalan, qurilmalardagi ayrim arifmetik kirish va chiqish operatsiyalari bitta registr – akkumulyator orqali amalga oshiriladi, natijada ayrim jarayonlarning bajarilishi bo'yicha registrlar orasida sakrash amalga oshirilishi lozim bo'ladi. Ikkinchi turi bo'lsa, hamma (deyarli hammasi) registrlarning bir xil vazifani bajarishi, ya'ni DEC firmasinig T-11, 16-razryadli protsessorlaridir. Bu yo'l bilan yuqori

barqarorlikga erishiladi, lekin protsessor sxemasini murakkablashtiradi. Bundan tashqari, oraliq protsessor turlari ham mavjud.Bu turga Motorola kompaniyasining MS68000 turli protsessoridagi mavjud bo'lgan registrlarning yarmi ma'lumotlar uchun ishlatilgan, lekin ular o'zaro almashinuvga ega. Qolgan yarmi esa manzillar uchun, bu ham o'zaro almashinuvga ega.

Holat (bayroq) registri-ham muhim ahamiyatga ega, lekin bu ham protsessor tarkibiy qismi hisoblanadi.Bu registrning tarkibida ma'lumotlar to'g'risida yoki manzillar to'g'risida ma'lumotni ichiga olmaydi. U o'z ichiga protsessorning holati so'zini (PXS) ichiga oladi. Bu so'z (bayroq)dagi har bir bit tarkibida bajarilgan buyruqning natijasi to'g'risida ma'lumot bo'ladi. Masalan, nolinchi natijaning biti mavjud, bu natija qachonki bajarilgandan so'ng buyruqning natijasi nolga teng bo'ladi. Bu bit(bayroq)lar shartli o'tishlarda bajariladi. Yana bu registrlarda gohida boshqarish buyruqlari bo'ladi, bu ayrim buyruqlarning o'tish rejimini aniqlaydi.

Uzilishlarni boshqarish sxemasi – bu protsessorga kelib tushgan uzilishlar to'g'risidagi so'rovlar, dasturdagi uzilishlar boshlang'ich manzilini aniqlaydi. PSW (Processor Status Word) – protsessor holati so'zi. Masalan, nol natijali holat bor deb faraz qilsak, agar oldingi bajarilgan buyruqning natijasi nolga teng bo'lsa yoki holat noldan farq qilsa unda protsessor xotirasidan o'chiriladi. Bu bitlar buyruqlar orqali shartli o'tishda ishlatiladi. Masalan, nolli natija bo'lgandagina buyruqlar o'tish holatiga o'tadi.Bu registrda gohida ayrim buyruqlarning rejimini aniqlash maqsadida boshqaruv bayroqlariga ega bo'ladi.

Uzilishlarni boshqarish sxemasi- bu sxema protsessorga kelib tushayotgan uzilishlar to'g'risidagi so'rovlarni qayta ishlaydi,dastur boshidagi uzilishlar manzilini qayta ishlaydi (uzilishlar manzili vektori). Bu esa dasturga mavjud bo'lgan buyruqlarni qayta ishlab keyingi holatga o'tishiga va xotirada protsessorning o'z holatini saqlab qolishga yordam beradi. Dastur so'ngida protsessor uzilishlarini qayta ishlovidan dastur yakunigacha xotiradan tiklangan ichki birlik registri bilan o'tadi.

To‘g‘ridan-to‘g‘ri boshqaruv sxemasi - bu sxema xotiraga protsessorning vaqtinchalik tashqi shinadan o‘chirilishiga va protsessorning vaqtinchalik to‘g‘ridan-to‘g‘ri qurilmaga kirishiga ruxsat berishi uchun uzilishdir.

Boshqaruv mantiqi – protsessorning hamma uzellarini o‘zaro harakatini amalga oshiradi, ma’lumotlarni qayta uzatadi, protsessorni tashqi signallar bilan sinxronizatsiyalaydi va axborotni kirish , chiqishiga javob beradi. Bu texnik tomondan mikroprotsessorning «qattiq mantiq» uslubidir.

Bu holda, protsessorning ish jarayoni buyruqlarni tanlash sxemasi ketma-ketlik bilan xotiradan olinadi, keyin buyruqlar bajariladi, zarur holatda esa ma’lumotlarni qayta ishlash uchun AMQ ishlatiladi. AMQ kirishiga xotiradan yoki ichki registrlardan qayta ishlangan ma’lumotlar uzatilishi mumkin. Ichki registrlar xotirasida qayta ishlanishi zarur bo‘lgan manzillar kodlari saqlanishi mumkin. AMQ dagi ma’lumotlar qayta ishlovi to‘g‘risidagi ma’lumotlarni holat registrining holatlarini o‘zgartiradi va bu to‘g‘risida ichki xotiraga yozadi. Zarur bo‘lgan holatda ma’lumot xotiradan, ichki registrdan qayta yozilishi mumkin.

Ammo, mikroprotsessor tizimlari dasturchi uchun mikroprotsessor ichki holatlari tizimlari ahamiyatga ega emas. Dasturchi protsessorga «qora quti» sifatida ahamiyat berishi kerak. Bu bilan kiruvchi va boshqaruvchi kodlarni chiquvchi kodlarga o‘zgartiradi. Dasturchiga buyruqlar tizimi, protsessor ish rejimi va protsessorning tashqi qurilmalar bilan o‘zaro aloqasini bilish talab etiladi. Protsessorning ichki holati tizimi haqida esa protsessorning u yoki bu holatlarda buyruqlarni holati yoki rejimlari ishlashini bilgan holatda amalga oshirishi mumkin.

Mikroprotsessor (MP), boshqachasiga central processing unit (CPU), - dasturli boshqariladigan, axborotni qayta ishlaydigan funksional tugallangan qurilma bo‘lib, u bitta yoki bir nechta katta (KIS) yoki juda katta (JKIS) integral sxemalar ko‘rinishda tayyorlangan.

Mikroprotsessor quyidagi vazifalarni bajaradi:

• asosiy xotiradan (AX) buyruqlarni o‘qish va deshifrlash (ochish);

• ma’lumotlarni AX dan va tashqi qurilmalar (TK) adapterlarining registrlaridan o‘qish;

• so‘rovlarni va buyruqlarni adapterlardan TQ larga xizmat ko‘rsatish uchun qabul qilish va qayta ishlash;

• ma’lumotlarni qayta ishlash hamda ularni AX ga va TQ, adapterlarining registrlariga yozish;

• ShK ning barcha boshqa uzellari va bloklari uchun boshqaruvchi signallarni ishlab chiqish.

Mikroprotsessor ma’lumotlart shinasining razryadliligi ShK ning razryadliligini aniqlaydi; MP adreslar shinasini razryadliligi uning adres kengligini aniqlaydi.

Adres kengligi – bu asosiy xotira yacheykalarining maksimal soni bo‘lib, u bevosita mikroprotsessor tomonidan adreslanishi mumkin.

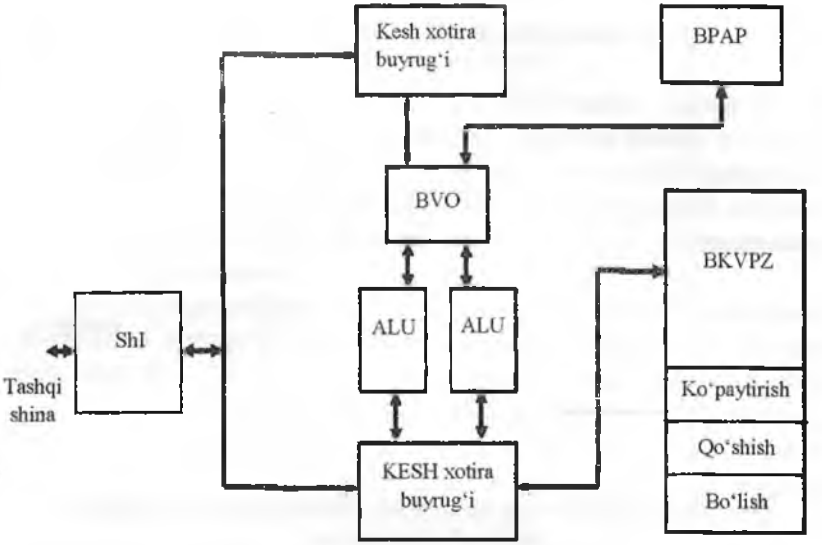
2.3. Pentium protsessorlarining umumiy strukturasi

Pentium protsessorining umumlashtirilgan strukturali sxemasi quyidagilarni o‘z ichiga oladi (2.4.chizma):

- ShI – 64-razryadli shina interfeysi;
- ikkita 32-razryadli butun sonli AMQ;
- komandalarning kesh-xotirasi;
- ma’lumotlarning kesh-xotirasi;
- umumiy vazifalarga mo‘ljallangan registrlar;
- ilgarilanma tanlamalar buferi ;
- o‘tish adreslarini oldindan bilish bloki ;
- suzuvchi nuqtali konveyer hisoblashlar bloki.

Shina interfeysi protsessor tashqi shinasini ichki shinasi bilan moslashtirish uchun mo‘ljallangan.

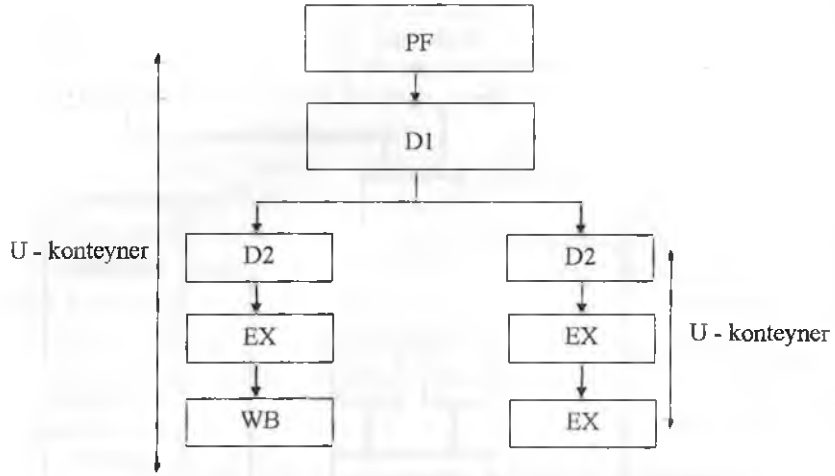
Kengaytirilgan 64-razryadli ma’lumotlar shinasi. Pentium MP shinalar siklining bir necha turlariga ega, ularga bitta siklda 256 bit ma’lumot kesh-xotiraga uzatiladigan paketli rejim ham kiradi. Bu i486DX protsessoriga qaraganda uzatish tezligini sezilarli darajada oshiradi. Masalan, 66 MGts chastotali shinaga ega Pentium MP ning uzatish tezligi 528 Mbayt/s, 50 MGts chastotali shinaga ega i486DX MP da esa 160 Mbayt/s.



2.5-chizma. Pentium mikroprotsessorining umumlashgan struktura sxemasi

Superskalyar arxitektura. «Superskalyar» atamasi bittadan ko'p hisoblash blokiga ega bo'lgan mikroprotsessor arxitekturasiga nisbatan qo'llaniladi. Pentium protsessori ikkita komandani bir vaqtda bajarishi mumkin bo'lgan ikkita konveyerga ega, U-konveyer komandalarning to'liq to'plamiga, V-konveyer esa to'liq bo'lmagan to'plamga ega. 3.16-rasmda konveyerlar ikkita butun sonli ALU, RON, BVO bilan soddalashtirib tasvirlangan. Bitta konveyerli i486 protsessori kabi ikkita konveyerli Pentium protsessori butan sonli komandalarni beshta bosqichda bajaradi (2.6.-chizma):

1. Oldindan taxmin qilingan komandalarni xotiradan olish (oldindan tanlov) - PF (PreFetch).
2. Komandani dekodlash – D1.
3. Komandani dekodlash – D2.
4. Komandani bajarish - EX.
5. Natijani yozish uchun mo'ljallangan buferga saqlash – EW.



2.6-chizma. Pentium protsessorida butun sonli buyruqlarni bajarish bosqichlari

Birinchi bosqich to‘rtta 32-razryadli buferga ega bo‘lgan BVO bloki bilan bajariladi. Ikkita bir-biriga bog‘liq bo‘lmagan tanlov buferlari jufti o‘tish bo‘lishi yoki bo‘lmasligini taxmin qiluvchi BPAP bloki bilan birga ishlaydi. Agar o‘tish taxmin qilinmasa, tanlov davom ettiriladi, agar taxmin qilinsa, boshqa bufer ishlashiga ruxsat beriladi va u o‘tish nuqtasidan tanlovni boshlaydi. Agar taxmin qilingan o‘tish amalga oshmasa, komanda konveyerlari tozalanadi va tanlov boshidan boshlanadi. Ikkinchi stadiyada komandani dekodlash xotiradagi operandlar adreslarini shakllantiradi. Har bir konveyer o‘zining bir taktda to‘lishi mumkin bo‘lgan, yozish uchun mo‘ljallangan 64-razryadli buferiga ega. Hech qanday o‘qish so‘rovlari buferda mavjud bo‘lgan yozish so‘rovlari tartibini buzmaydi. Pentium MP yozishning qat’iy ketma-ketligini qo‘llaydi.

Yuqori samarali arifmetik soprotsessor BKVPZ 8-taktli konveyer va ko‘paytirish, bo‘lish arifmetik operatsiyalarini bajaruvchi apparat vositasini o‘z ichiga oladi. Suzuvchi nuqtali operatsiyalarning katta qismi bitta butunsonh konveyerda bajarilishi mumkin, so‘ng ular suzuvchi nuqtali hisoblash konveyeriga tushadi. Pentium ichki joylashtirilgan arifmetik soprotsessorining samaradorligi FPU-486

(Floating-Point Unit) soprotsessorining samaradorligini 2-10 marta oshiradi.

Ikkita konveyerdan foydalanish bir necha komandalarga bajarilishning turli bosqichlarida bo'lish imkonini beradi va konveyerni komandalar bilan to'liq egallanishi hisobiga MP samaradorligini qo'shimcha ravishda oshiradi. Pentium protsessorida arifmetik operatsiyalarni apparatli bajarishdan foydalaniladi, bu ham protsessor samaradorligini oshiradi.

Kesh-xotirani komanda va ma'lumotlarga bo'lish. Pentium mikroprotsessori komanda va ma'lumotlar uchun bo'lingan kesh-xotirasiga ega. Bu i486 protsessoridan farqli ravishda bitta komanda uchun tanlov jarayoni va boshqa komanda uchun ma'lumotlarga murojaat jarayonlari o'rtasidagi konfliktni bartaraf qilishga imkon beradi. Kesh-xotira komandalar va ma'lumotlar uchun bo'linganda ikkala komanda bir vaqtda bajarilishi mumkin. Pentium protsessorida komanda kesh-xotirasi va ma'lumotlar kesh-xotirasi hajmi bir xil, 8 Kbaytni tashkil qiladi. Komanda va ma'lumot kesh-xotirasi ikki kirishli assotsiativ kesh-xotira sxemasi bo'yicha amalga oshiriladi. Ma'lumotlar kesh-xotirasi ikkita interfeysga ega (har bir konveyer uchun bittadan), bu esa bitta mashina sikli davomida ikkita alohida komandani ma'lumot bilan ta'minlashga imkon beradi.

Ma'lumotlar kesh-xotirasi kechiktirilgan yozuv bilan ishlaydi (tashqi shina bo'shatilgunga qadar) va teskari yozuv rejimiga sozlanadi. Oxirgi holatda ma'lumotlar kesh-xotiradan olinadi, bundan keyin esa asosiy xotiraga yoziladi. Keshlashning bunday usuli protsessor bir ma'lumotni bir vaqtda ham kesh-xotiraga, ham asosiy xotiraga yozadigan oddiy keshlashga qaraganda yuqori samaradorlikka ega. Ma'lumotlar kesh-xotirasi MESI protokolini qo'llaydi. Bu protokol boshqa protsessorlarning kesh-xotiralariga murojaat qilish imkoniyatini beradi. Protokolning nomlarini MESI kesh-xotira qatorlari holatlari nomlanishidan kelib chiqadi: M (Modified), E (Exclusive), S (Shared), I (Invalid). Kesh-xotira qatorlari holatlari quyidagicha tavsiflanadi:

M-holat – ko'rib chiqilayotgan protsessorning faqat kesh-xotirasida mavjud bo'lgan qator. Qator modifikatsiya qilingan, ya'ni asosiy xotira tarkibidan farq qiladi. Unga yozuvlar kiritish tashqi murojaat siklini generatsiya qilmay amalga oshiriladi;

E-holat - ko‘rib chiqilayotgan protsessorning faqat kesh-xotirasida mavjud bo‘lgan, ammo modifikatsiya qilinmagan qator. Qatorga yozuv kiritish tashqi murojaat siklini generatsiya qilmay amalga oshiriladi. Qatorga yozuv kiritilganda u M-holatga o‘tadi;

S-holat – qator ko‘rib chiqilayotgan protsessor kesh-xotirasida va boshqa protsessorlar kesh-xotiralarida bo‘lishi mumkin. Undan o‘qish tashqi siklni generatsiya qilmay amalga oshiriladi, yozuv kiritish esa boshqa protsessorlar kesh-xotirasidagi mos yozuvlarni bekor qilinishiga olib keluvchi asosiy xotiraga kiritish bilan amalga oshiriladi;

I-holat – kesh-xotirada mavjud bo‘lmagan qator, uni asosiy xotiradan o‘qish kesh-xotirani qatorlar bilan to‘ldirish siklini generatsiya qilinishiga olib keladi. Kesh-xotira qatoriga yozuv kiritishda tashqi shinadan foydalaniladi.

Multiprotsessor ish rejimini qo‘llash. Pentium arxitekturasi ikki va undan ortiq Pentium protsessorlarini multiprotsessor tizimida ishlatishi mumkin. Pentium MP ikkinchi avlodidan boshlab simmetrik arxitekturali ikki protsessorli tizimni qurish arxitekturasini qurish interfeysi qo‘llangan.

Xotira sahifalari hajmini berish vositasi. Pentium protsessori xotira sahifasi hajmini tanlash uchun optsiyaga (boshqaruv maxsus biti) ega: an’anaviy (4 Kbayt) va kengaytirilgan (4 Mbayt).

Funksional qo‘shimchalar yordamida xatoni aniqlash va testlash vositasi. Pentium protsessorida ishonchlilikni oshirish maqsadida ichki qurilmalar va tashqi shina interfeysi xatoliklarini aniqlash (ichki nazorat pariteti), adreslar shinalari nazorat pariteti, funksional qo‘shimchalar yordamida testlash ko‘rib chiqilgan. Ichki xatolarni aniqlash komanda va ma’lumot kodlariga juftlik bitlarini qo‘shishdan iborat, bu tizim va foydalanuvchi uchun xatolarni aniqlash imkonini beradi. Funksional qo‘shimchalar yordamida testlash dasturiy ilovalarda foydalaniladi. Funksional qo‘shimchalar yordamida testlash ikkita Pentium protsessorlarining asosiy/nazorat qiluvchi (master/checker) konfiguratsiyasidagi ishiga asoslanadi. Bunday konfiguratsiyada asosiy protsessor odatiy bir protsessorli rejimda ishlaydi. Nazorat qiluvchi protsessor ham shu operatsiyalarni bajaradi, ammo shinani boshqarmaydi va asosiy protsessor chiqish signallarini o‘zi generatsiya qilayotgan signallar bilan solishtiradi. Olingan natijalar

mos kelmagan holatda tizimda uzilish sifatida qayta ishlanishi mumkin bo'lgan xatolik signali shakllanadi. Bunday usul 99% dan ko'proq xatolarni aniqlashga imkon beradi. Bundan tashqari, testlash vositasi ichki o'rnatilgan test BIST (Built In Self Test) ni bajarish imkonini beradi, bu mnemokodlar, dasturlanayotgan mantiqiy matritsalar xatoliklarini aniqlash, komanda va ma'lumot kesh-xotiralarini adres buferlarini va doimiy xotira qurilmasini testlashni ta'minlaydi. Umuman olganda o'z-o'zini testlash protsessorning 70% tugunlarida bajariladi. Barcha protsessorlar JTAG raqamli qurilmalarni testlash ketma-ket interfeysi yordamida o'z-o'zini testlash uchun IEEE 1149.1 standart test portiga ega.

Pentium protsessorlarining o'ziga xosliklari quyidagilar:

- bir necha yangi komandalarning mavjudligi, shu jumladan protsessor modelini aniqlash;
- energiya ta'minotini boshqarish vositasining mavjudligi;
- shina sikllarini konveyer adreslanishini qo'llash;
- komandani bajarish vaqti (taktlar soni) qisqartirilgan;
- komandalar trassirovkasi va samaradorlik monitoringi;
- virtual rejim imkoniyatlarini kengaytirish – uzilish virtual bayrog'ining mavjudligi;

Yangi qo'shimcha sozlash vositalari:

- zondlangan rejim (Probe Mode), ichki registr, kiritish-chiqarish va protsessor tizim xotirasiga murojaatni ta'minlaydi. Bu rejim protsessor holatini ichki sxema emulyatori imkoniyatlari kabi imkoniyatga ega bo'lgan dastur yordamida tekshirish va o'lchash imkonini beradi;
- kengaytirilgan sozlamalar (DE, Debug Extensions), kiritish/chiqarish komandalari adresi bo'yicha nazorat nuqtalarini o'rnatish imkonini beradi;
- ichki hisoblagichlar, samaradorlikning joriy nazorati va hodisalar sonining hisobi uchun foydalaniladi;

Arxitekturani kengaytirish. 32-razryadli Pentium protsessorlarining bazaviy arxitekturasiga qo'shimcha MSR (Model Specific Registers) modeli uchun maxsus registrlar to'plamiga ega. MSR registrlar to'plami MP ning turli modellaridan turlicha bo'ladi, bu ularning mumkin bo'lgan mos tushmasliklariga olib keladi. MSR

registrlarida foydalaniladigan dasturiy ta'minot CPUID komandalari yordamida olingan protsessor haqidagi ma'lumotlardan foydalanishi kerak.

MSR registrlar tarkibiga quyidagilar kiradi:

- test registrlari TR1 – TR12;
- samaradorlik monitoringi vositasi;
- mashina xatosi nazoratini chaqiruvchi adres va ma'lumotlar sikli registri.

Test registrlari protsessorlarning ko'pgina funksional tugunlarini boshqarishni, ularning ishga yaroqliligini testlash imkoniyatini ta'minlaydi. TR12 registr bitlari yordamida yangi arxitektura xususiyatlarini, shuningdek, kesh-xotira ishini taqiqlash mumkin.

Samaradorlik monitoringi vositasi apparat va dasturiy ta'minotni dastur kodida potensial «tor joy»larni paydo bo'lishi hisobiga optimizatsiya qilish imkonini beradi. Ishlab chiqaruvchi ichki protsessor hodisalari taktlarini kuzatishi mumkin, bu o'qish va yozuvlar kiritish, kesh-xotiraga «omadli» va «omadsiz» murojaatlar, uzilish, shinadan foydalanish operatsiyalari samaradorligiga ta'sir ko'rsatadi. Bu dastur kodining effektivligini baholash va dasturiy ilovaning maksimal samaradorligiga erishish imkonini beradi. Samaradorlik monitoringi vositasi real vaqt taymeri va hodisalar hisoblagichi hisoblanadi. Taymer TSC (Time Stamp Counter) 64-razryadli hisoblagich bazasida qurilgan, tarkibi protsessor ishining har taktida inkrementatsiya qilinadi. Uning tarkibini o'qish uchun RDTSC komandasidan foydalaniladi. 40-razryadli hodisalar hisoblagichlari CTR0, CTR1 shina operatsiyalari, komandaning bajarilishi, konveyer, kesh-xotira, nazorat nuqtalari ishi va boshqalarga bog'liq bo'lgan turli klass hodisalarini hisoblashga dasturlanadi. Hodisa turini bildiruvchi olti bitli maydon har bir hisoblagichga mustaqil ravishda katta ro'yxatdagi hodisalarni hisoblash imkonini beradi. Bundan tashqari tashqi liniyalar RM1-RM0 mavjud, ular mos hisoblagichlarning ishlashi va to'lib qolish omillari ko'rsatkichlariga dasturlanadi.

Mashina xatosi nazoratini chaqiruvchi adres va ma'lumotlar sikli registri nomi turli klasslar (Pentium va Pentium Pro) uchun yoki xatto protsessor turli modellari uchun mos tushmasligi mumkin. Ulardan

foydalanayotgan dastur CPUID komandasi bilan protsessor haqidagi ma'lumotlarga murojaat qilishi kerak.

Pentium protsessorlari ishlamayotgan rejimda energiya ta'minotini kamaytirish imkoniyatiga ega. STOPCLK# signali bilan protsessor buferdan kechiktirilgan yozuvni yuklaydi va Stop Grant rejimiga o'tadi, bunda ko'pgina protsessor tugunlarining takti kamayadi, bu esa energiya ta'minotini taxminan 10 martaga kamaytiradi. MP bu holatda komandalarni bajarishni to'xtatadi va uzilishlarga xizmat ko'rsatmaydi, lekin ma'lumotlar shinasini kuzatishni davom ettiradi. Protsessor bu holatdan STOPCLK# signalining to'xtashi bilan chiqadi. SMM rejimidan foydalanib STOPCLK# rejimini boshqarish ta'minotni kengaytirilgan boshqaruv mexanizmi APM (Advanced Power Management) tomonidan amalga oshiriladi. Quvvatga talabni proporsional kamaytirish bilan protsessorni sekinlashtirish uchun STOPCLK# signali davriy impulslardan iborat bo'lishi kerak.

Protsessor kamaytirilgan energiya ta'minoti Auto HALT Power Down holatiga HALT komandasini bajarish vaqtida o'tadi. Bu holatda protsessor barcha uzilishlarni boshqaradi va shuningdek, shinani kuzatishda davom etadi.

Tashqi sinxronizatsiyani to'xtatish rejimida protsessor minimal quvvat iste'mol qiladi, ammo hech qanday funksiya bajarmaydi. Sinxronizatsiya signallarining ketma-ket uzatilishi RESET signaliga mos bo'lishi kerak.

2.4. Ko'p yadroli protsessorlar

Mur qonunida aytilishicha, yarim o'tkazgichli mikrosxemaga joylashtirilgan tranzistorlar soni har ikki-yilda ikki martaga ko'payadi, bu esa bir tomondan unumdorlikni oshishiga, boshqa tomondan mikrosxemalarni ishlab chiqarish bahosining pasayishiga olib keladi. Bu qonunning muhimligi va haqqoniyligiga qaramasdan, ko'p-yillar davomida, kelgusida rivojlanish istiqbollarini bahalagan holda, vaqti-vaqti bilan uning muvaffaqiyatsizligidan qutilib bo'lmasligi aytilar edi.

Kelgusidagi rivojlanish yo'lidagi to'siqlarga quyidagi omillar sabab bo'lishi mumkin: fizik o'lchamlarni cheklanganligi, energiya

iste'molining keskin ravishda o'sishi va ishlab chiqarishning haddan tashqari yuqori xarajatlari. Ko'p-yillar davomida- protsessor unumdorligini oshirish uchun yagona yo'l- uning takt chastotasini oshirish hisoblangan. Bu-yillar davomida, shunday fikr ildiz otgan ediki, aynan protsessorning takt chastotasi uning unumdorligining asosiy ko'rsatkichi hisoblanardi. Zamonaviy bosqichda takt chastotasini oshirish- bu asosiy vazifa emas. Mikroprotsessorlarning takt chastotalari poygasining tugashiga toklar yo'qolishining yechilmagan muammolari va mikrosxemalarning issiqlik ajralib chiqishining nomaqbul o'sishi tufayli erishildi.

Protsessor unumdorligi (Performance)- bu dasturiy kodning bajarilgan yo'riqnomalari umumiy sonining ularni bajarilish vaqtiga nisbati yoki bir soniyada (Instructions rate) bajariladigan yo'riqnomalar sonidir:

$$\text{Unumdorlik} = \frac{\text{yo'riqnomalar soni}}{\text{bajarilish vaqti}}$$

Protsessorning asosiy tavsifi uning takt chastotasi bo'lgani uchun chastotani protsessorning unumdorligi formulasiga kiritamiz. Sur'at va mahrajini yo'riqnomalar bajarilgan taktlar soniga ko'paytiramiz:

$$\text{Unumdorlik} = \frac{\text{yo'riqnomalar soni}}{\text{taktlar soni}} \quad \frac{\text{taktlar soni}}{\text{bajarilish vaqti}}$$

Olingan ko'paytmaning birinchi qismi- bir taktda bajarilgan yo'riqnomalar soni (Instuction Per Clok, IPC), ko'paytmaning ikkinchi qismi vaqt birligida protsessor taktlari soni (protsessorning takt chastotasi, F yoki Frequency). Shunday qilib protsessor unumdorligi nafaqat uning takt chastotasiga, taktda bajariladigan yo'riqnomalar soniga ham bog'liq (IPC)

$$\text{unumdorlik} = (\text{IPC})\,(\text{F})$$

Olingan formula protsessor unumdorligining oshishiga ikkita turli yondashishni belgilaydi. Birinchisi-protsessorning takt chastotasining oshirish, ikkinchisi protsessor bir taktida bajariladigan

dasturiy kod yo'riqnomasi sonini oshirishdir. Takt chastotasining oshirish cheksiz bo'lishi mumkin emas va bu protsessorning ishlab chiqish texnologiyasi bilan aniqlanadi. Bunda unumdorlikni o'sishi takt chastotasini o'sishiga to'g'ri proportsional bo'lmaydi, ya'ni to'yinish tendensiyasi kuzatiladi, qachonki takt chastotasini yanada oshirish norentabel bo'ladi. Bir takt vaqtida bajariladigan yo'riqnomalar soni protsessorning mikroarxitekturasiga : ijro bloklarining soniga, konveyerning uzunligiga va uning to'lish samaradorligiga, bloklariga, protsessorning ushbu mikrosxemasiga dasturiy kodning optimallashtirilganligiga bog'liq bo'ladi. Shuning uchun protsessor unumdorligini uning takt chastotasi asosida taqqoslash faqat bitta arxitektura chegarasida (IPC protsessorlar-bir soniyada bajariladigan operatsiyalar sonining bir xil qiymatida) mumkin.

Takt chastotasi asosida turli arxitekturali protsessorlar unumdorligini taqqoslash g'ayri qonuniydir. Masalan, takt chastotasiga asosan protsessorlar unumdorligini 1, 2 darajadagi xotira keshining turli o'lchamlari bilan yoki Hyper Threading texnologiyasini ta'minlaydigan va ta'minlamaydigan protsessorlar unumdorligini taqqoslash noto'g'ridir.

Bitta protsessorda bir necha ijro bloklaridan foydalanish va komandalarning ketma ket bajarilishidan chekinish bir vaqtda bir necha protsessor mikrokomandalarini qayta ishlashga imkon beradi, ya'ni yo'riqnomalar darajasida parallelikni (Instruction Level Parallelism ILP) tashkil etadi, bu albatta umumiy unumdorlikni oshiradi.

Ushbu muammoni hal etishga yana bir yondashish VLIW/EPIC-IA-64 arxitekturasida (juda uzun komandalar) amalga oshirilgan, bunda muammoni bir qismi apparaturadan kompilyatorga yuklatilgan va barcha ishlab chiqaruvchilar unumdorlikka erishish uchun arxitektura muhimligini tan oladilar.

Mikrosxema funksional bloklarining orasidagi masofa uzun bo'lganda signallarni tarqalish tezligiga bog'liq bo'lgan muammolar yuzaga keladi. Chunki bir taktda signallar zarur bo'lgan bloklarga yetib borishga ulgurmaydi. Ushbu muammoni yechish uchun ALPHA mikroprotsessorlarga «klasterlar» kiritilgan. Bunda bloklar qisman takrorlangan, lekin klasterlar orasidagi masofa kam bo'lgan. Ko'p yadroli mikroprotsessorlarni qurish g'oyasi klasterlar g'oyasini

rivojlantirish hisoblanadi deb aytish mumkin, lekin bu holatda protsessor yadrosi butunligicha takrorlanadi.

Intel-Hyper Threading texnologiyasini ko‘p yadroli yondashishning boshqa o‘tmishdoshi deb hisoblash mumkin, bunda umumiy yadrodan foydalanuvchi yo‘riqnomalarning ikkita oqimidan foydalanish va apparaturalarning katta bo‘lmagan takrorlanishi mavjud .

Ko‘p yadroli protsessor ikkita yoki ko‘p «ijroli yadrolarga» ega. Protsessorning yadrosi deb uning ma’lumotlarga ishlov berish uchun mo‘ljallangan ijro moslamalari tizimini (arifmetik-mantiqiy moslamalar to‘plami) aytish mumkin. Operatsion tizim ijro yadrolaridan har birini barcha zaruriy hisoblash resurslari bilan diskret protsessor sifatida qabul qiladi. Shuning uchun protsessorning ko‘p yadroli arxitekturasi tegishli dasturiy ta’minoti asosida bir necha dasturiy oqimlarni parallel holda to‘liq bajarishni amalga oshiradi.

2006-yilda mikroprotsessorlarning barcha yetakchi ishlab chiqaruvchilari ikki yadroli protsessorlarni yaratdilar. Birinchi bo‘lib ikki yadroli RISC-protsessorlar Sun Microsystems (Ultra SPARC 4) va HP (PA-8800 va PA-8900). AMD va Intel firmalari x86 arxitekturali ikki yadroli protsessorlarni ishlab chiqarilganligi to‘g‘risida deyarli bir vaqtda e’lon qildilar.

Protsessorlar arxitekturasi yetarlicha yuqori murakkablikka erishdi, shuning uchun ko‘p yadroli protsessorlarga o‘tish hisoblash tizimi unumdorligini oshirishning asosiy yo‘nalishi bo‘lib qoladi.

Zamonaviy mikroprotsessorlar arxitekturasini rivojlanishining asosiy yo‘nalishi ularning unumdorligini oshirishga intilish bilan aniqlanadi. Unumdorlikni oshirish mumkin, masalan takt chastotasini oshirish bilan (yoki) bir taktda bajariladigan komandalar sonini ko‘paytirish bilan. Ushbu muammoni yechishning biri –TLP(Thread Level Parallelism) «treydlar (oqimlar) darajasida parallelizm» konsepsiyasini amalga oshirishga bog‘liq. Agar dasturiy kodlar barcha yoki ko‘pgina funksional qurilmalarni ish bilan yuklashi mumkin bo‘lmagan hollarda, protsessorga bittadan ortiq vazifani (trend yoki oqimni) bajarishga ruxsat berish kerak. Ushbu holda qo‘shimcha oqimlar bo‘sh turgan funksional qurilmalarni band etadi. Bunda ko‘p vazifali operatsion tizim bilan o‘xshashligini ko‘rish qiyin emas: protsessor vazifani kutish holatida bo‘lganda (masalan kirish- chiqishni tugashi) protsessor to‘xtab turmasligi uchun

operatsion tizim protsessorni boshqa vazifani bajarishga qayta ulaydi. Bundan tashqari, operatsion tizimdagi ayrim despetcherlash mexanizmlari ko'p oqimli arxitekturaga (MTA- MultiThreading Architecture) o'xshashligi mavjud. Shubhasiz, oqimlar darajasida parallelizmni ta'minlaydigan arxitektura (TLP) , treydlar bir vaqtda bitta bir xil resurslarni ishlatmasligini kafolatlashi kerak, buning uchun qo'shimcha apparat vositalari talab etiladi. Lekin zamonaviy superskalyar protsessorlar bazasida MTA amalga oshirish mumkin va bu katta bo'lmagan apparatni tamomiga yetkazish talab etiladi, loyihalovchilar nazarida MTAning jozibadorligini keskin oshiradi.

Mikroprotsessor TLP darajasida parallellikni amalga oshirish uchun kamida ikkita apparat ta'minotiga ega bo'lishi kerak. Bular umumiy mo'ljallangan registrlar, komandalar hisoblagichi, protsessor holatining so'zi va shunga o'xshashlar. Vaqtning har qanday paytida, faqat bitta oqim (tred) ishlaydi. U muayyan vaziyat yuz bergunicha bajariladi (masalan/ma'lumotlar kesh-xotirada yo'q bo'lganda registrni yuklash komandasini bajarish). Bu holda protsessor boshqa oqimni bajarishga qayta ulanadi. Kesh-xotirada ma'lumot topilmagan holatda, uni xotiradan olish operatsiyalari uchun protsessorning yuzta taktlarigacha vaqt ketishi mumkin. Ushbu holda protsessor kutib qoladi. Zamonaviy protsessorlar bunday vaziyatlarda boshqa komandalarni bajarishni davom ettirishi mumkin, lekin amaliyotda mustaqil komandalar soni tez tugaydi va protsessor to'xtaydi.

Treadlarni bir vaqtda bajaradigan arxitektura- SMT (Simuitaneous Multi Threading) bir necha oqimlarni bir vaqtda bajarishga ruxsat etadi. Bu holda har bir yangi taktda har qanday oqim komandasi qandaydir ijro qurilmasiga bajarishga yo'naltirilishi mumkin. SMT uchun quyidagi apparat vositalari zarurdir:

- Bir nechta komandalar hisoblagichlari (oqimga bittadan), har bir taktda ularning har birini tanlash imkoniyati bilan;
- Oqim bilan komandalarni bog'laydigan vositalar (xususan o'tishlarni oldindan aytib boradigan va registrlarni qayta nomlaydigan registrlar ishlashi uchun);
- Qism dasturdan qaytish stek adreslari (oqimga bittadan), (qaytish adreslarini oldindan aytib berish uchun);

• Buferdan navbatdan tashqari bajarilgan komandalarni olib tashlash jarayoni uchun protsessorda joylashgan maxsus qo‘shimcha xotira (har bir oqim hisobida).

Ko‘pgina zamonaviy protsessorlardagi SMT xususiyatlaridan biri- registrlarni qayta nomlashdir, qachonki mantiqiy (arxitekturali) registr fizik registr sifatida aks ettiriladi va ular bilan real ish olib boriladi. Registrlarni qayta nomlash usuli registr fayllarini to‘g‘ridan-to‘g‘ri takrorlanishidan asraydi.

Hyper-Threading texnologiyasi.

Hyper-Threading texnologiyasi ko‘p oqimli ishlov berishga misol bo‘ladi. Hyper-Threading texnologiyasi bitta real fizik protsessor asosida ikkita mantiqiy protsessorni tashkil etishga imkon beradi. Operatsion tizim nazarida ikkita protsessor mavjud. Shuning uchun, operatsion tizim vazifalarni ikkita mantiqiy protsessorlar taqsimlashi mumkin.

Hyper-Threading texnologiyasida parallel rejimda yo‘riqnomalarni bajarishi mumkin (ketma- ketlikda emas), ya’ni ishlov beriish uchun barcha yo‘riqnomalar ikkita parallel oqimlarga bo‘linadi. Bu esa bir vaqtda ikkita turli ilovalarni yoki bitta ilovaning ikkita turli oqimlariga ishlov berishga imkon beradi va shu bilan protsessorning IPC (soniyada protsessor bilan bajariladigan yo‘riqnomalar soni) ko‘rsatkichini oshiradi, bu uning unumdorligini o‘sishiga olib keladi.

Konstruktiv planda Hyper-Threading texnologiyasi asosida protsessor ikkita mantiqiy protsessorlardan tashkil topib, ularning har birida o‘z registri va uzilish kontrolleri (Architecture State, AS) bo‘ladi. Shunday ekan, ikkita parallel ijro qilinadigan vazifalar o‘zining xususiy mustaqil registrlari va uzilish kontrollerlariga ega. Lekin vazifalar bitta real fizik protsessordan foydalaniladi. Aktivlashtirilgandan keyin mantiqiy protsessorlardan har biri o‘z vazifasini mustaqil va boshqa protsessorga bog‘liq bo‘lmagan holda bajarishi, uzilishlarga ishlov berishi mumkin. Shunday qilib, ushbu texnologiya real ikki protsessorli konfiguratsiyadan shu bilan farqlanadiki, ikkita mantiqiy protsessorlar umumiy resurslardan (xotira, shinalar) foydalaniladi. Ikkita mantiqiy protsessorlarni qo‘llash zamonaviy operatsion tizimlarda va yuqori samarali ilovalarda amalga oshirilgan oqimlar darajasida parallelizm jarayonini kuchaytirishga imkon beradi. Protsessor yadrosi bir nechta ijro

modullaridan foydalanishi hisobiga ikkita oqimni parallel holda bajarishga qodir.

Hyper-Threading texnologiyasi g'oyasi Pentium 4 protsessorining NetBurst mikroarxitekturasi bilan chambarchas bog'liq va qandaydir ma'noda uning mantiqiy davomi bo'lib hisoblanadi.

Intel NetBurst mikroarxitekturasi yo'riqnomani bir oqimini bajarishda unumdorlikning maksimal yutug'iga ega bo'lishga imkon beradi. Lekin dasturni maxsus optimizatsiyalagan holda ham, protsessorning ijro modulining hammasi ham har bir takt sikli davomida to'liq ishlatilmaydi. IA-32 komandalarini bajarishda protsessorlarning ijro resurslari, o'rtacha 35% ishlatiladi, protsessorlarni ijro resurslarining 65% esa bo'sh turadi, bu protsessorning imkoniyatlaridan samarasiz foydalanishni ko'rsatadi. Protsessor ishlashini shunday amalga oshirish kerakki, har bir takt siklida uning imkoniyatlaridan maksimal foydalanish mumkin bo'lsin. Aynan shu g'oyani Hyper-Threading texnologiyasi parallel vazifalarni bajarishga protsessorning bo'sh turgan resurslarini jalb etgan holda amalga oshiradi.

Vazifalarni yanada samarali parallel holda bajarish jarayonini amalga oshirish uchun ikki va undan ortiq yadroni bitta mikroprotsessorda integrallashdir. Bitta kristalldagi bunday ko'p yadroli konfiguratsiya ko'p protsessor tizimlarida tashqi shinalar, kommutatorlar va hokazolarda foydalanishdan ko'ra yadrolar orasidagi almashinishning eng yuqori tezligini ta'minlaydi.

Ko'p yadroli arxitektura protsessorlar unumdorligini oshirish uchun ikkita yoki undan ko'p resurslarni to'liq funksional to'plamidan iborat bo'ladi. Ko'p yadrolik va 6,5 nm texnologik jarayon elektr iste'molini yetarlicha iqtisod qilishga va unumdorlikni oshirishga imkon berdi.

2.5. Mikroprotsessor registrlari va xotira segmentlari

MP tizimidagi xotira maydoni odatda bir qancha asosiy maxsus funksiyalar uchun ajratilgan. Boshlang'ich yuklanish dasturi doimiy xotira qurilmasi yoki flesh-xotirada yozilgan bo'ladi. Protsessor manbaaga ulanishi bilan aynan shu boshlang'ich yuklanish dasturi orqali o'z ishini boshlaydi va RESET tugmasini bosish bilan o'z ishini tugatadi. Ma'lumki, qurilmaning boshlang'ich yuklanishini

mikroprotsessor xotiraning nolinchi manzilidan boshlab amalga oshiradi. Kiritish va chiqarish qurilmasini uchta maxsus guruhga bo‘lish mumkin:

- foydalanuvchi qurilmasi interfeysi (foydalanuvchi axborotni kiritishi va axborotni foydalanuvchiga chiqarish);
- axborotni uzoq saqlash uchun kiritish \ chiqarish qurilmasi;
- taymer qurilmasi.

Mikroprotsessor ichki xotiradagi kiritish \ chiqarish qurilmasiga yoki shinali qurilma tizimiga o‘zining tizim xotirasiga murojaat qilish kabi imkoniyati mavjud.

Manzillar xotirasini va kiritish \ chiqarish qurilmasini ajratishda odatda ikki asosiy yo‘nalish mavjud:

- kiritish \ chiqarish qurilmasi uchun umumiy manzillar joyini belgilash ;
- xotira manzillar joyini va kiritish \ chiqarish qurilmasi to‘liq ajratish;

Bundan birinchi holat shunisi bilan yaxshiki, protsessor kiritish \ chiqarish qurilmasiga murojaat qilganida xotiraga murojaat qilgan buyruqlar orqali bog‘lanishi mumkin. Lekin manzilli xotira joyi kiritish \ chiqarish qurilmasi xotirasidagi joy bilan to‘g‘ri kelishi lozim. Masalan, 16 - razryadli shina manzili 64K adres bo‘lishi mumkin.Undan 56K adreslar manzillar o‘rniga mos keladi, 8K esa katta manzillar - kiritish \ chiqarish qurilmasi osti joyiga tegishli bo‘ladi.

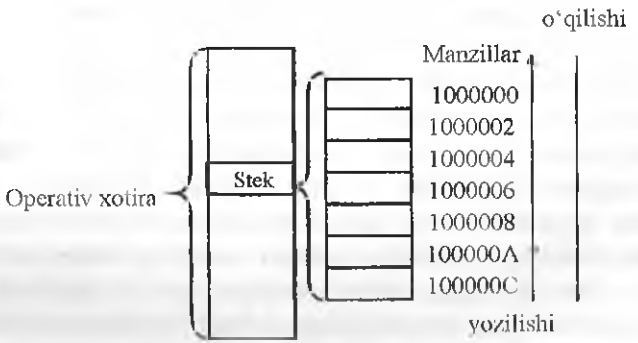
Ikkinchi usulning farqi shundaki, bunda xotira mikroprotsessor tizimida manzilli joyning barchasini egallaydi. Kiritish \ chiqarish qurilmasi bilan bog‘lanish uchun shina boshqaruvining magistralida maxsus buyruqlar va axborot almashinishning maxsus stroblari orqali bog‘laniladi. Personal kompyuterlarda bu usul aynan shunday amalga oshiriladi. Ammo bu usul xotira hisobida olinganda kiritish \ chiqarish qurilmasi bilan imkoniyatlari chegaralangan.

Fon Neyman arxitekturasida xotira buyruqlar va ma’lumotlar uchun umumiy bo‘ladi. Garvard arxitekturasida esa, buyruqlar xotirasi CSEG(Code Segment) va ma’lumotlar xotirasi DSEG(Data Segment) alohida ajratilgan bo‘lib, ularda alohida manzillar joyi va unga kirish holati mavjud.

Barcha zamonaviy MP larda alohida «stek» maydoni ajratilgan bo'lib, ular protsedura parametrlarini uzatishda va adreslarni saqlab ularga qaytishda ishlatiladi. Stek MP maxsus ichki maydonida yoki operativ xotirada joylashishi mumkin. Keltirilgan holatda DSEG adres maydonining qismida joylashgan. Stek LIFO(Last in –first out) rejimida ma'lumotlarni vaqtinchalik saqlash uchun foydalaniladi. Stekning afzalligi, belgilangan va o'zgartirilmaydigan adreslash usuli hisoblanadi.

Stekdan ma'lumotlarni o'qishda stek ko'rsatkichidagi adreslar orqali o'qiladi va stek ko'rsatkichi inkrementlanadi. Natijada LIFO usulidagi xotira tashkillashtiriladi, ya'ni birinchi keldi oxirgi ketdi (2.7.-chizma).

Masalan, stek ko'rsatkichining joriy holati 1000008, unga ikkita so'z yozish kerak. Birinchi adres 1000006(yozishdan oldin stek ko'rsatkichi ikkitaga kamayadi), ikkinchi adres 1000004. Yozish natijasida stek ko'rsatkichi-1000004 ga teng bo'ladi. Agar stekdan ikkita so'zni o'qish kerak bo'lsa, birinchi bo'lib 1000004 adresdan so'z o'qiladi, keyin stek ko'rsatkichi 1000006 ga teng bo'ladi. Ikkinchi bo'lib 1000006 adresdan so'z o'qiladi, stek ko'rsatkichi 1000008 ga teng bo'ladi. Hammasi o'z holatiga qaytdi. Birinchi yozilgan so'z ikkinchi bo'lib, ikkinchi so'z birinchi bo'ladi.



2. 7-chizma. Stekning ishlash prinsipi

Ushbu keltirilgan adreslash dasturni bir necha marta joylashtirishda qulaydir. Masalan asosiy dastur bajarilish jarayonida 1 chi qism dastur chaqiriladi. Agar asosiy dastur holatini saqlash kerak

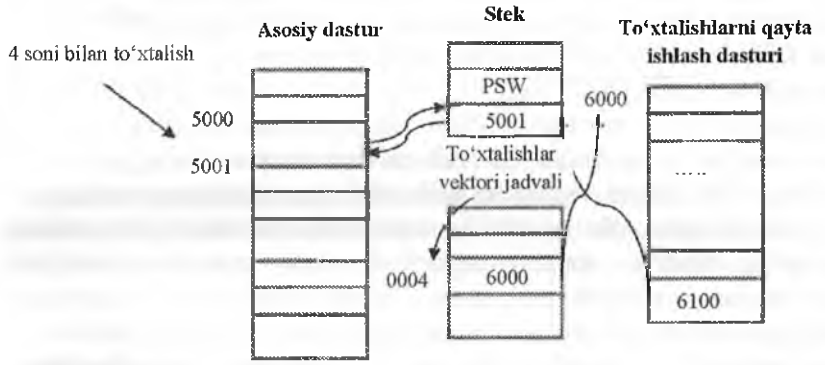
bo‘lsa stekda saqlaymiz va qism dastur bajarilgandan so‘ng stekdan chaqirib olishimiz mumkin.

Stek tizimidagi ma’lum vaqt ichida ko‘p pog‘onali uzilishlarni tashkillashtirishda foydalanadi. Akkumulyator vazifasini stek cho‘qqisiga uzatuvchi stek arxitekturasi mavjud. Stekli tashkillashtirish qisqa uzunlikka ega bo‘lgan adressiz buyruqlardan foydalanish imkoniyatini yaratadi. Adressiz buyruqlar stek cho‘qqisi va undan keyingi joylashgan ma’lumotlar ustida amallar bajaradi. Operatsiyalar bajarilish jarayonida joriy operandlar stekdan chaqirilib,natija stek cho‘qqisiga uzatiladi. Stekli arxitektura yuqori hisoblash effektivligiga ega. Yuqori pog‘onali til FORT mavjud bo‘lib, u adressiz buyruqlar asosida yaratilgan. Bunday arxitektura yuqori o‘tkazuvchanlikka ega bo‘lgan protsessorlarda, xususan RISC protsessorlarida qo‘llaniladi. Keyingi maxsus xotira maydoni DSEG-bu uzilish vektorlari jadvalidir.

Uzilishlar - bu faqat tashqi qurilmalardan so‘rovlarni tashkil qilish emas, balki protsessor ketma-ketlikdagi ishining buzilishidir. Masalan arifmetik operatsiyalarning noto‘g‘ri bajarilishi natijasida (nolga bo‘lish)uzilishlar bo‘lishi, yoki uzilishlar dasturiy bo‘lishi masalan qism dasturdan chiqish va qaytishdagi uzilishlar bo‘lishi mumkin. Har qanday uzilishlar vektor uzilishlar jadvali asosida qayta ishlanadi. Ushbu jadvalda uzilishlarni qayta ishlovchi dastur boshlang‘ich adresi joylashgan bo‘lib, vektor deyiladi. Jadval uzunligi katta bo‘lishi mumkin(bir necha yuz element). Odatda uzilishlarni tashkil qilish vektori xotiraning bosh maydonida joylashgan. Har bir vektorning adresi uzilish raqamini belgilaydi. Apparat uzilishlarda uzilish raqami qurilmalar orqali beriladi. Protsessor apparat uzilishlarni olgandan so‘ng joriy buyruqni bajarishni to‘xtatadi va xotira maydonidagi uzilishlar vektori jadvaliga murojaat qiladi. Protsessor o‘z navbatida ushbu qator tarkibini o‘qiydi va ushbu vektor orqali berilgan xotira adresiga o‘tadi. Undan tashqari ushbu adresdan boshlab belgilangan raqamga ega bo‘lgan uzilishlar dasturi joylashadi. Ushbu jarayon bajarilayotganda protsessor parametrlari stekda saqlanadi.

Masalan, protsessor (2.8.-chizma) xotira adresi 5000 da joylashgan buyruq va asosiy dasturni bajaryapti. Shu vaqtda vektor adresi 4 ga teng uzilishlar so‘rovini oladi. Protsessor 5000 adresdan buyruq

bajarishni tugallaydi. Keyin (5001) buyruq hisoblagichi joriy qiymatini stekda va PSW «registr so'z holati» ni saqlaydi. Undan keyin protsessor 4 adresdan uzilish vektor kodini o'qiydi. Bu kod 6000 ga teng bo'lsin. Protsessor 6000 xotira adresiga o'tadi va shu adresdan boshlab uzilishlarni qayta ishlashga o'tadi. Dasturni 6100 adresda tugallanadi deb hisoblaymiz. Ushbu adresga kelib protsessor uzilgan dastur joyiga qaytadi. Buning uzilishlar tashkil qilingan vaqtdagi stekdan 5001 adres qiymatini oladi, keyin protsessor 5001 adresdan buyruqni o'qib asosiy dasturni bajarishga o'tadi.



2.8-chizma. To'xtalishlarni qayta ishlashning algoritmi

Avariya holatlari ham xuddi shunday uslubda amalga oshiriladi. Dasturiy uzilishlar ham uzilishlar vektor jadvali asosida xizmat ko'rsatiladi, lekin uzilish raqami buyruqlar tarkibida ko'rsatiladi. Bunday uzilishlar dasturchi uchun xotiraning istalgan joyida uzilishlarni tashkil qilish imkoniyatini yaratadi. Uzilishlarni qayta ishlash dasturi bajarilayotgan davrida yangi so'rovlar bo'lishi mumkin. Bunday holatda oldingi uzilgan dastur qayta ishlanadi. Bunday holatlar ko'p martali ichma-ich uzilishlar deyiladi. Stek mexanizmi ichma-ich uzilishlarga xizmat ko'rsatishga imkon yaratadi, ya'ni stekdan oxirgi joylashgan kodga birinchi bo'lib xizmat ko'rsatiladi. Murakkab holatlarda uzilishlar vektori jadvalida uzilishlar qayta ishlash dastur bosh adresi emas, balki uzilishlar deskriptori joylashadi. Lekin deskriptor qayta ishlash natijasi uzilishlar qayta ishlash

dastur bosh adresi bo'ladi. CSEG va DSEG dan tashqari hamma zamonaviy mikroprotsessorlar maxsus ajratilgan kichik hajmli ma'lumot bo'shlig'iga ega, ular dasturli kirish to'plamiga ega RSEG (Register Segment). CSEG va DSEG ga qaraganda RSEG registrlari markaziy protsessor va MP ning ichida bevosita ALQ ga yaqin joylashgan bo'lib, ma'lumotlarga fizik kirishni ta'minlaydi. Ularda oraliq hisob natijalari saqlangan bo'ladi. RSEG registr joyi DSEG ma'lumot maydonidan ajratib qo'yilishi, qisman kesishishi mumkin va DSEGning adres qismi bo'lishi mumkin.

RSEG ning ichki mantiqiy tashkillashtirilishi turli xil bo'lib, mikroprotsessor klassifikatsiyasida asosiy rolni o'ynaydi. MP registrlari funksional turdosh: ular ma'lumotlar yoki adresli ma'lumotlarni saqlash uchun xizmat qiladilar, boshqalari – markaziy protsessorni boshqarish uchun xizmat qiladilar. Shunga asoslanib ushbu registrlarni: ma'lumot registrlari, ko'rsatkich va maxsus registrlarga ajratish mumkin. Ma'lumot registrlari arifmetik va mantiqiy operatsiyalarni bajarishda operandlar manbai va ma'lumotlarni qabul qilish, adres registrlari yoki ko'rsatkich registrlari asosiy xotirada joylashgan buyruq va ma'lumot adreslarini hisoblash uchun foydalaniladi. Maxsus registrlar-markaziy protsessor holati va tarkibiy qismlarini boshqarishda foydalaniladi. Umumiy ishlatiladigan registrlar bloki (UIRB)-ma'lumot adreslar va ma'lumotlarni saqlash uchun foydalaniladi. Keltirilgan registrlardan foydalanishga qarab mikroprotsessor arxitekturalari afzalliklarini ajratishimiz mumkin. Ma'lumot registrlari ichida ko'pincha bitta registr akkumulyator A (Accumulator) ajratish mumkin. U arifmetik va mantiqiy operatsiyalarni qayta ishlash bilan bog'liq, ya'ni operatsiyalar bajarilishi natijasida har doim bir operand akkumulyatorda saqlanadi va natija akkumulyatorda bo'ladi. Ushbu arxitektura akkumulyatorli deyiladi. Kamchiligi esa – kichik tezlik, ya'ni har doim operatsiya bajarishdan oldin operandni kiritish kerak.

Ishchi registrlar R0, R1...v.b. – arifmetik va mantiqiy operatsiyalar natijasi bitta registrda emas turli registrlarda saqlanishi mumkin, bu esa o'z navbatida imkoniyatlarni yaratadi. Dasturlash jarayonida registrlar adreslari ko'rsatiladi. Bunday tipli arxitektura registrli arxitektura deyiladi.Bularga intel firmasiga mansub x86 mikroprotsessorlari misol bo'ladi. Bir qancha real vaqtda ishlaydigan

mikroprotsessorlarda ishchi registrlar to‘plami nazarda tutilgan. Ma’lum ishchi registrga murojaat ketma-ketlikda bajariladi. Ushbu qurilmalarga intel firmasiga mansub.

Maxsus registrlarga MP ichki qismida joylashgan va MP tarkibiy qismlari, turli ishlarni boshqarish funksiyasini bajaradi. Ko‘p uchraydigan registrlarga adres registrlari(yoki ko‘rsatkich)-RS va SP- va PSWdastur holat so‘zi kiradi. Adres yoki ko‘rsatkich registrlar, MP ma’lum buyruqlarida foydalaniladigan operand adreslash usullarida foydalaniladi. Ushbu adreslash MP modeliga bog‘liq. Ular albatta ikki ishchi funksiyani bajaradi:

- bajarilayotgan buyruqning xotiradagi adresini aniqlaydi(buyruq hisoblagichi);
- joriy stek adresini aniqlaydi(stek ko‘rsatkichi).

Turli protsessorlarda har bir funksiya uchun bir yoki ikki ichki registrlar ajratilishi mumkin. Keltirilgan registrlar bir biridan tarkibini o‘zgartirishi bilan farqlanadi. Tarkibidagi dasturni o‘zgartirilishi kompyuter ishining buzilishiga yoki xotirani o‘zgarib ketishiga olib keladi. RS tarkibi quyidagi tartibda o‘zgartiriladi.

Tizim ishining boshlanishida o‘zgartirilmaydigan qiymat yozib qo‘yiladi. Bu nolga teng bo‘lgan boshlang‘ich ishga tushirilgan dastur adresi. Xotiradan buyruqlarni o‘qish natijasida buyruq formatiga qarab dastur hisoblagichi tarkibi osha boradi.

Dastur holatlar so‘zi:PSW(Program Status Word) operatsiya bajarilish natijalari alomatlarini saqlaydi intel kompaniyasiga tegishli MP x86 ko‘rib chiqamiz (2.9-chizma).

– ZF-(zezo) nomli natijaning bayroq belgisi agar nol natija olinsa, 1 yoziladi, aks holda (ZF) = 0.

– CF-(Carry) siljitish bayrog‘i operatsiya bajarish natijasida katta baytda siljitsa, yoki qo‘shish yoki ayirish operatsiya jarayonida 1 qarzga olinganda o‘rnatiladi.

– SF -(Sing) natija belgisining bayrog‘i birga teng, agar natija manfiy bo‘lsa, ya’ni natijaning katta belgili bitni dublashtiradi.

– PF -(Parity) juftlik bayrog‘i (PF=1) teng agar natija bitining 2 modulli yig‘indisi nolga teng bo‘lsa (birlik bitlar soni juft).

– AF -qo‘shimcha siljitish bayrog‘i (Auxi liary) agar kichik tetradaning katta bitidan (bit D3) katta tetradaning kichik bitiga (bit

D4) siljitilganda o‘rnatiladi operatsiyalarda joylashtirilgan BSD sonlari ustida qo‘llaniladi

– OF -to‘lib ketganlik bayrog‘i (Overflow) agar operatsiya natijasi bir yoki ikki baytli diapazondan oshib ketganida o‘rnatiladi va boshqa bir necha hollarda o‘rnatiladi.

- TF(Trap Flag)- ketma-ket qadam rejimi bayrog‘i.
- IF(Interrupt Flag)-uzilishlarga ruxsat bayrog‘i.
- DF(Direction Flag)- qatorli operatsiyalarda ko‘rsatish bayrog‘i.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				OF	DF	IF	TF	SF	ZF		AF		PF		CF

2.9-chizma. x86 prossessori registrining holati

Xotirani segmentli tashkil qilish bilan bir qatorda, x86 mikroprotsessorlarida qo‘shimcha ravishda xotirani sahifali tashkil qilish ham mumkin. Xotirani sahifali tashkil qilish mexanizmi CRO regisrining RG bayrog‘ini, o‘rnatish (tashlash) yo‘li bilan dasturiy ulanishi (uzilishi) mumkin.

Butun chiziqli adres fazosi soni 1024 ga etadigan bo‘limlarga ajratiladi. Har bir bo‘lim o‘z navbatida o‘lchami 4 k bayt qilib belgilangan 1024 tagacha sahifani ichiga olishi mumkin, sahifalarning boshlang‘ich adreslari fizik adres fazosida qat’iy belgilangan. Sahifalarning chegaralari 4 kilobaytli bloklar chegaralari bilan mos tushadi.

2.6.Adreslash usullari

MPT uchun dastur yozish –juda murakkab jarayondir. Samarali dasturlash uchun protsessor buyruqlar tizimidan xabardor bo‘lish kerak. Kompakt va tezkor dastur va qism dasturlar Assembler tilida yaratiladi. Assembler tili mashina tilining raqamli kodlari, buyruqlar tizimining mashina kodlarini ta’minlaydi. Protsessor buyruqlar tizimidan xabardor bo‘lmay yozish mumkin bo‘lgan dasturlar juda ko‘p, ulardan ko‘proq ishlatiladigani Si dasturlash tilidir.

Assembler dasturlash tili – personal kompyutyerdan tortib mikrokontrollerlarda dasturiy ta’minotning asosiy zarur qismidagi

dasturlar uchun effektiv natija beradi. Buyruq kodlari qanday operandlar bilan yoki qaysi buyruq bajarilishi to‘g‘risida protsessorga ma’lumot beradi. Buyruqlar bir baytdan bir necha baytgacha uzunlikka ega bo‘ladi. Buyruq kodlari protsessorda qayta shifrlanadi va mikrooperatsiyalar to‘plamiga aylantiriladi.

Bir turdagi buyruqlar kirish operandlarini, boshqalari esa chiqish operandlarini talab qiladi. Kirish operandlari manba-operand, chiqish operandlari qabul qilish-operand deyiladi. Hamma operand kodlari qayerdadir joylashishi kerak. Ular ichki registrlarda,x otira tizimida joylashadilar. Ularning joylashuvi buyruq kodlari orqali ko‘rsatiladi. Ma’lumotlarni qayerdan olib qayerga joylashning turli usullari mavjud bo‘lib ular adreslash usullari deyiladi [12].

Adreslash usuli deb-operand adresini kodlash yoki buyruq kodlaridagi operatsiya natijasiga aytiladi.

2.10-chizma. Buyruq kodlari

OPK-operatsiya kodi.

OPA1- birinchi operand adresi.

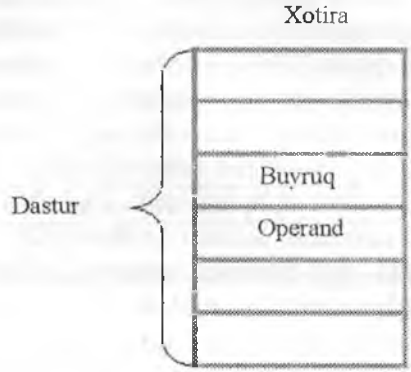
AR-natija adresi.

Zamonaviy mikroprotsessor modellarida keng foydalaniladigan adreslash usullari;

- registrli adreslash;
- to‘g‘ri adreslash;
- bevosita adreslash;
- mavhum registrli adreslash;
- mavhum avtoinkrement/avtodekrementli adreslash;
- baza bo‘yicha indeksli adreslash;
- segmentli adreslash.

Bevosita adreslash. Bevosita-operand ko‘rsatkichining eng oddiy usuli. Komandalar adres maydonida operand adresi emas, balki operand joylashadi. Operand avtomatik ravishda komanda bilan xotiradan chaqiriladi va darhol bevosita adreslash uchun tayyor bo‘ladi. Bevosita adreslashda qo‘shimcha xotiraga murojaat talab

etmaydi. Lekin uning kamchiligi bor. Birinchidan faqat konstantalar bilan ishlash mumkin, ikkinchidan qiymatlar maydoni chegaralangan. Shunga qaramay bu texnologiya ko'p arxitekturalarda butun sonlar ustida ishlash uchun foydalaniladi (2.11-chizma).



2.11-chizma. Bevosita adreslash

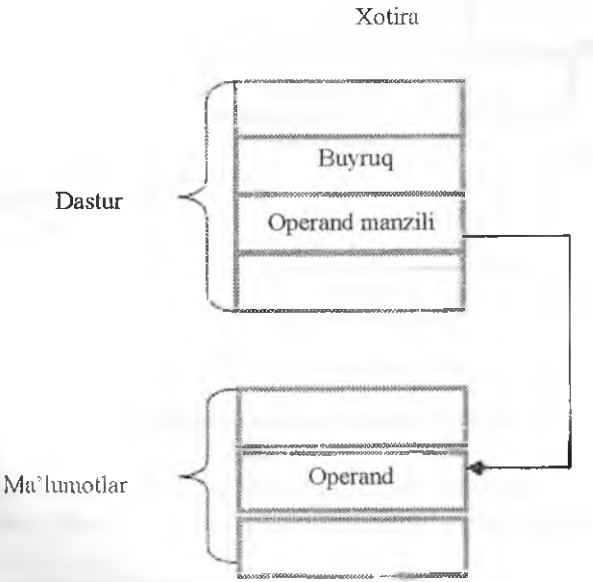
To'g'ri adreslash. To'g'ri - adres maydonida (ADR) operand adresi joylashadi. Komanda har doim bitta xotira adresiga murojaat eta oladi.

Masalan 1000000 xotira adresini tozalash kerak bo'lsa, u holda tozalash buyrug'idan keyin joylashadi. (2.12-chizma).

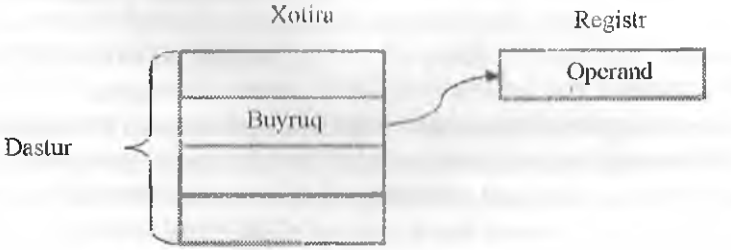
Registrli adreslash. Operand protsessorning ichki registrida joylashadi. Masalan, buyruq quyidagicha bo'lishi mumkin, nol registrdan birinchi registrga ma'lumot uzatish. Ikkita registrning raqami buyruq kodlari orqali aniqlanadi. Kod operatsiyalarida registr adreslari shifrovka qilingan. Buyruqda adres maydoni ko'rsatilmaydi (2.13-chizma).

Mavhum registrli adreslash. Protsessor ichki registrida operand emas balki uning adresi joylashadi. Operand fizik adresi mavhum adres DP (Data Pointer)da joylashadi. DP (Data Pointer) sifatida UIRB (RON) yoki registr maxsus adres registr joylashadi. (2.14-chizma).

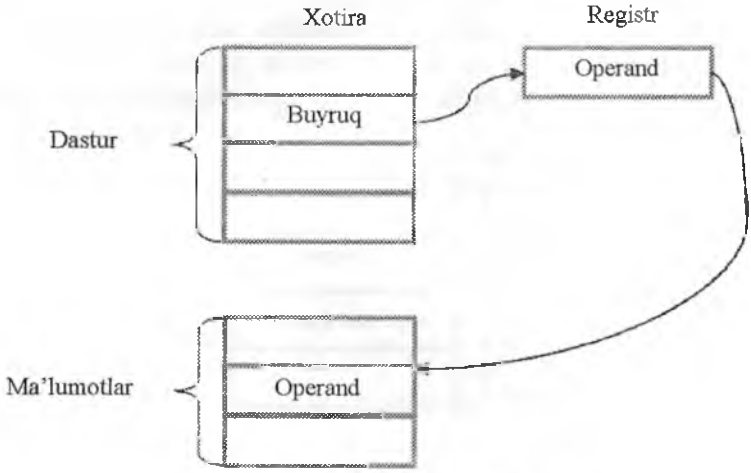
```
ADD          A, MEM
...
MEM : XDATA C3F0h
```



2.12-chizma. To'g'ridan to'g'ri adreslash



2.13-chizma. Manzillashni ro'yxatdan o'tkazish



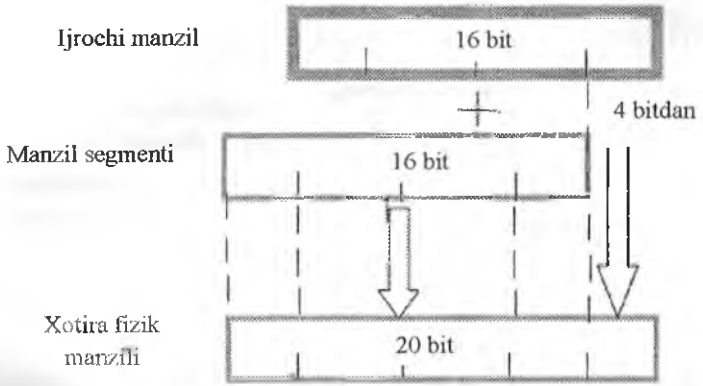
2.14-chizma. Mavhum manzillash

Masalan, Assembler 51 tilida MEM adresi bo'yicha xotiraning ikki baytini tozalash quyidagicha bo'ladi:

```
MOV     DP, #MEM
        CLR     a
        MOVX    @DP, a
        INC     DP
        MOVX    @DP, a
        ...
MEM: XDATA C3F0h
```

Xotirani segmentli adreslash. 8086 mikroprotsessori 20 razryadli adreslar shinasiga ega va u xotiraning 2^20 yoki taxminan milliondan bir yacheykasiga murojaat qilinishiga imkon beradi. 16 bitli ma'lumotlar shinasi ma'lumotni baytli yoki so'zli ko'rinishida uzata oladi. Xotira odatda chiziqli bir o'lchamli baytlar massivi ko'rinishida tuzilgan va 2 qo'shni baytlar bir so'zdek ko'rinishi bo'lishi mumkin. Xotiraning butun megabaytli sohasi 64 Kbitli 4 segmentga bo'lingan (2.15.-chizma). MP har bir daqiqada dastur segmentiga (CS),

ma'lumotlar segmentiga(DS), qo'shimcha ma'lumotlar segmentiga (ES) va stek segmentiga (SS) murojaat qilish imkoni bor. Bu segmentlarning boshlang'ich adresi CS, SS, DS va ES registrlarida saqlanadi.



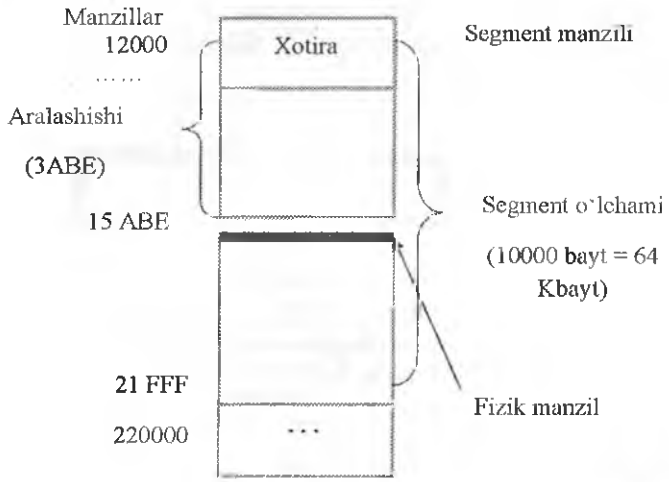
2.15-chizma. Manzil segmentidan xotiraning fizik manzilga o'zgarishi

Bu registrlar 16 bitli, adresli soha esa 20 bitli bo'lganligi sababli MP boshlang'ich segmentli adresni 20 bitli summatorda chapga 4 bit suradi (16ga ko'paytirishga teng) va (IP,SP,DI,SI) registrlardan biridagi ma'lumotga qo'shadi. Hosil qilingan son fizik adres deb ataladi. Masalan, xotiradan navbatdagi dastur kodi baytini olishda MP quyidagi formula orqali fizik adresni hosil qiladi:

Fizik adres = (IP) + (CS)*16, bu yerda (IP)-siljitish, bajaruvchi adres (CS)-segmentli adres, (SS)*16-boshlang'ich segmentli adres deb ataladi. 2.16.-chizmada xotirani tashkil etish keltirilgan.

Protsessor 4 ta guruh buyruqlar to'plamini o'z ichiga oladi:

- ma'lumotlarni uzatish buyruqlari;
- arifmetik buyruqlar;
- mantiqiy buyruqlar;
- o'tish buyruqlari.



2.16-chizma. Segmentdagi fizik manzil (hamma kodlar – o'n oltilikda)

2.7. Mikroprotsessor buyruqlar tizimi

O'tish buyruqlari hech qanday operandalar ustida operatsiyalarni talab qilmaydi. Operandlar manbaa registrlaridan (Source) qabul qiluvchi registrlarga (Destination) uzatiladi.

Arifmetik buyruqlar qo'shish, ayirish, ko'paytirish, bo'lish, inkrementlash, dekrementlash operatsiyalarini bajaradi. Ushbu buyruqlarga bir yoki ikki kirish operandi zarur bo'ladi.

Mantiqiy buyruqlar mantiqiy VA, YOKI, inversiyalash, turli xil surish operatsiyalarini amalga oshiradilar.

O'tish buyruqlari dastur ketma-ketligi holatini o'zgartiradi. Ularning yordamida tarmoqlangan dasturlar va qism dasturlarga o'tish tashkil qilinadi. O'tish buyruqlari shartli va shartsiz bo'ladi. Aynan shu buyruqlar murakkab ma'lumotlarni qayta ishlash operatsiyalarini amalga oshiradilar. Har bir bajarilgan buyruq asosida ma'lumotlar natijasi alomati tekshiriladi (PSW). Turli protsessorlarda komandalar farqlanadi, lekin ularning bajarilish funksiyasi o'xshash. Masalan 8086 protsessorida 133 buyruq bor.

Uzatish buyruqlari.

1. MOV DST, SRC; (SRC) ni (DST) ga uzatish. Shu yerda va keyinchalik registr ichidagisi, masalan AL registri (AL) yoki (al) ko'rinishida belgilanadi izohni uzatish esa ‹-- belgi bilan belgilanadi.

mov al ch; ‹--(ch)

mov cx, dx ;

mov bn, [mems]; mems simvollik adresli xotira yacheykasi ichidagisi

VN registrga uzatish.

mov al, [bx]; VX registrida joylashgan adresli xotira yacheykasini

akkumulyatorga uzatish.

mov bx, OFFSET src; joriy segmentda SRC xotira yacheykasi adresining siljishini BX ga joylashtirish.

Buyruqlar jadvali

2.1-jadval

Buyruq bajarilishidan oldin mov al, [table+bx]	Registr VX	Registr AL	Manzil	Kod
	0010	xx	0800(table)	8c
			08xx	Xx
			0810	58
Buyruq bajarilishidan so'ng mov al, [table+bx]	Registr VX	Registr AL	Manzil	Kod
	0010	xx	0800(table)	8c
			08xx	Xx
			0810	58

2. PUSHRP; juftlik registridan ma'lumotni stekning yuqorisiga joylashtirish (masalan, push bx).

3. POPRP; stek yuqorisidan 2 bayt olib uni RP juftlikka joylashtirish (masalan, pop ax).

4. XChG DST, SRC; (DST) va (SRC)lar ichidagilarini joylarini almashtirish. Ikki operand bir vaqtda xotira yacheykasidagi ma'lumot bo'la olmaydi.

5. XLAT SRC; jadval boshidan (AL) soniga teng ma'lumotlar bayti SRC boshlang'ich adresli jadvaldan olib, uni AL ga joylashtirish. SRC adres BX registrida joylashgan bo'lishi kerak.

6. IN ACCUM, PORT; AL yoki AX baytini yoki PORT adresli portdan so'zni akkumulyatorga joylashtirish. Agar port adresi <= FF bo'lsa, port adresi bevosita ko'rsatilishi mumkin, agar port adresi >FF bo'lsa, port adresi registri ifodasi orqali bevosita ko'rsatiladi (POH maxsus funksiyasi).

OUT PORT, ACCUM; AL akkumulyatoridan yoki AX baytidan yoki so'zni PORT simvolik adresni TQ ga uzatish.

7. OUTOUT PORT, ACCUM; AL akkumulyatoridan yoki AX baytidan yoki so'zni PORT simvolik adresni TK ga uzatish.

Arifmetik buyruqlar .

1. ADD DST, SRC; SRC va DST qiymatlarini qo'shish.

add al, [mem_bute]; mem_bute-bir bayti xotira yacheykasi

add [mem_word], dx; mem_word-ikki baytli xotira yacheykasi

add ch,10001010b;

2. INC DST; (DST) ni 1taga oshirish (inkrement (DST)).

inc si; (SI) <-- (SI) + 1.

inc count; (count) <-- (count) + 1.

3. SUB DST, SRC; (SRC) ni (DST)dan ayirish va natijani DST ga joylashtirish.

4. DEC DST; (DST) ni 1 taga kamaytirish;

5.5. CMP DST, SRC; DST va SRC ni ichidagisini solishtirish. Bu buyruq (SRC) dan (DST)ni ayirishni bajaradi, lekin farqini DST ga joylashtirmaydi va operatsiya natijasini natijasi bo'yicha bayroqlarga ta'sir ko'rsatadi.

shart	bayroqlar			
	OF	SF	ZF	CF
DST > SRC	0/1	0	0	0
DST = SRC	0	0	1	0
DST < SRC	0/1	1	0	1

0/1 - operatsiyaning qiymatiga qarab bayroqni 0 yoki 1 ga tengligini bildiradi. OF va SF, bayroq belgili sonli operatsiyalarda ma'noga ega, CF bayrog'i esa 1 ga qo'yiladi, agar qo'shish yoki

ayirish operatsiyasi natijasida qoldiq qiymatini katta ikkilik razryadiga o'tkazganda va shu katta ikkilik razryadidagi bilan mos kelganda boshqa aniqlashda OF 1 qiymatini qabul qiladi, agar natija berilgan oraliq mos ravishdagi sonlar oralig'idan oshib ketsa. DST › SRC va ikkalasi ham bir baytli sonlar bo'lsa, unda:

DST:	1. (+127)	2. (+127)
SRC:	- (+2)	- (-2)
	-------	-------
	(+125) (OF)=0	(+129)? (OF)=1

Ikkinchi misolda natija oraliqdan oshib ketayapti: -128‹=x‹=+127 SF belgilangan bayroq "1" ga qo'yiladi, agar operatsiya natijasining katta biti 1ga teng bo'lsa, ya'ni masofiy natijada. Teskari bo'lganda tushirib ketiladi. ZF nol bayrog'i (!) nolli natijada "1" ga qo'yiladi aksida tushirib qoldiriladi. Uzatish bayrog'ga CF=1 teng, agar qo'shishida katta razryaddan uzatish yoki ayirishda kichik razryaddan qoldiq olish bo'lsa. Aks holda bayroq tushirib qoldiriladi. Birinchi misol uchun SF=ZF=CF=0, ikkinchisi uchun: SF=1, ZF=CF=0.

Mantiqiy va surish buyruqlari.

1. AND DST, SRC; razryad bo'yicha "I" (VA).

mov dh, 10101100b;

and dh, 0f0h;

shu ikki buyruqni bajarilishi natijasida DH ning ichidagisi 10100000b ga teng bo'lib qoladi.

2. OR DST, SRC; razryad bo'yicha mantiqiy element "ILI" (YOKI).

or bx,dx;esli (BX)=5F0Fh,(DX)=7777h bo'lsa operatsiyadan so'ng;(BX)=7F7Fh.

Mantiqiy surish jadvali

2.2-jadval

BX	0101 1111 0000 1111 = 5F0F
DX	0111 0111 0111 0111 = 7777
BX (natija)	0111 1111 0111 1111 = 7F7F

3. XOR DTS, SRC; razryad bo'yicha mantiqiy "sikl. ILI" xor al,55h;agar (AL)=5ah bulsa, operatsiyadan so'ng (AL)=0fh.

4. NOT DST; qabul qiluvchi hamma bitlarni inversiyasi.

5. TEST TEST DST, SRC; AND operatsiyasini bajaradi, ammo faqat bayroqlarga operatsiyalarni o'zgartirmasdan ta'sir ko'rsatadi.

6. ShR DST, CNT; mantiqiy o'nga surishgandan bo'sh qolayotgan bitlar nollar bilan to'ldiriladi, o'ng tomondagi chetki bit SF bayrog'iga chiqarib yuboriladi. DST operandi xotira yacheykasi bo'lishi mumkin.

mov bl,10110010b;(CF) = x

shr bl,1;(BL) = 01011001,(CF) = 0

Siljishgacha		1	0	1	1	0	0	1	0		(CF)=X
Siljishdan keyin	0–>	0	1	0	1	1	0	0	1	–>	(CF)=0

mov cl,4;

shr bl,cl;(BL) = 00000101,(CF) = 1.

7. ShL DST, CNT; matiqiy chapga surish.

8. RLC DST, CNT; qoldiqni o'tkazish orqali chapga siklik surish.

9. RRC DST, CNT; qoldiqni o'tkazish orqali o'ngga siklik surish.

10. ROR DST, CNT; chapga siklik surish.

11. ROL DST, CNT; o'ngga siklik surish.

Boshqarishni uzatish buyruqlari

1. CALL SUB R ; SUBR adresli qism dasturini chaqirish call delay;

mov ....

2. RET CALL ga bevosita qism dasturidan keyingi operatorga qaytarish, ya'ni yuqorida keltirilgan misoldagi MOV ga .

3. JMP NAME ; NAME simvolik adresli buyruqga shartsiz o'tish

jmp short name; name belgisiga o'tish, quyidagidan kam bo'lmagan holda:

-128 yoki +127 bayt.

jmp near name; name belgisiga o'tish, quyidagidan kam bo'lmagan holda:

65535 bay, bitta segmentda.

jmp name;oddiy jmp near name;

4. JA NAME yoki JNBE NAME : shartli o‘tish, agar masalan: CMR DST, SRC solishtirish natijasida qabul qiluvchi uzatuvchidan absolyut kattalik bo‘yicha katta bo‘lsa, name belgisiga o‘tish kerak.

5.5. JB NAME yoki JNAE NAME: shartli o‘tish agar masalan, CMR DST, SRC solishtirish natijasida qabul qiluvchi manbadan absolyut kattalik bo‘yicha kichik bo‘lsa, unda name belgisiga o‘tish kerak (4n va 5n, buyruqlari belgisiz sonli operatsiyalarni bajarish natijalari ustida bajariladi).

6. JZ NAME yoki JE NAME: o‘tish, agar nolinchi bayroqga ta’sir qiluvchi operatsiya natijasi - nol bo‘lsa ("nol" bo‘yicha o‘tish).

7. JNZ NAME: yoki JNE NAME; "nol emas" bo‘yicha o‘tish.

Sikllarni boshqarish buyruqlari .

1. LOOP NAME: bu buyruq (SX)ni 1 ga noaniq kamaytiradi va yaqin belgiga o‘tishni bajaradi, agar (SX) 0 ga teng bo‘lmasa.

2. LOOPZ NAME yoki LOOPE NAME: bundan tashqari ZF bayrog‘ining tekshirishni ham bajaradi. Shuning uchun sikl (CX)=0 yoki (ZF)=0 yoki u ham bu ham birgalikda bo‘lgandagi shart bilan tugaydi. Shuning bilan bu buyruq birinchi nol emas bo‘lgan natijani aniqlash uchun xizmat qiladi.

3. LOOP NZ, LOOP NE - "nol" bo‘yicha sikldan chiqish.

Qatorlar ustida ishlash buyruqlari

1. LOD.SB: lodsb buyrug‘i ma’lumotlar segmentidan ST regisr orqali adreslangan baytni yuklaydi va ST ni 1ga ko‘paytiradi, agar bundan oldin CLD buyrug‘i kiritilgan bo‘lsa (DF yo‘nalish bayrog‘ini tozalash) va St ni 1ga kamaytiradi, agar STD buyrug‘i ishlatilgan bo‘lsa (yo‘nalish bayrog‘i o‘rnatilishi) .

```
.....
DATA
string DB 'abcdefg'
.CODE
....
cld; AL da bu buyruqlarni bajarilgandan so‘ng
mov si,OFFSET [string+2]; ASCII ga 's' kodi yuklandi.
lodsb; SI tarkibidagilar 'd' ni ko‘rsatadi.
```

2. MOVSB: bu buyruq SI registridagi xotira yacheykasi adresidan bir baytni DI registrdagi xotira yacheykasi adresiga o‘tkazadi va (SI) va (DI) ni 1 ga ko‘paytiradi. SI ning qiymati DS

ma'lumotlar segmentida ham qo'shimcha ES segmentida joylashishi mumkin. DI ning qiymati faqat ES qo'shimcha segmentda joylashishi mumkin.

```
....
.DATA
 msg  DB 'Hammasi O.K.'
 LEN  = $ - msg;LEN 8 ga teng
.CODE
....
cld
lea si,msg;v SI manba adresi
mov ax,0b800h; Видео xotira segment hajmi
mov es,ax;qo'shimcha segmentga o'tkazish
lea di,es:(0a0h * 3); yuqoridan 4-satr
mov cx,LEN;LEN – chiqish simvollar soni
rp:   movsb; ekranning joriy pozitsiyasiga simvolni yuborish
inc di;atribut pozitsiyasidan sakrab o'tish
loop rp; tugaguncha davom ettirish (CX)
....; stroka 'Vsyo O.K.' displey yuqorisidan
;4- satrga chiqariladi.
```

3. REP: buyruqni qaytarish prefiksi. Masalan, oldingi dasturning tugashi quyidagicha yozilishi mumkin:

```
mov cx,LEN; qaramasdan, massivi 'Hammasi O.K.' bo'ladi.
rep   movsb; B800 ni xotira maydoniga ko'chiriladi:(A0 * 3), ga
;display ekraniga quyidagi yozuv chiqadi:Hammasi OK.
```

Nimaga?

4. CMPSB; (ST) adresni beruvchining qator baytini, (DT) adresli qabul qiluvchining qator baytini solishtirishni bajaradi: ya'ni ((ST))-((DI)) ayirishni bajaradi. CMP DST, SRC buyrug'i bilan yangilish-maslik kerak, qaysiki qabul qiluvchidan beruvchini ayirish bajariladi. CMPSB buyrug'i (CX)ni 1 ga noaniq kamaytiradi va (SI) va (DI) ni 1 ga ko'paytiradi, agar (DF)=0 bo'lsa.

5. REPZ yoki REPE: qaytarish prefiksi. Agar (CX)=0 yoki (ZF)=0 bo'lsa, buyruqni bajarish tamomlanadi.

```
DATA
src  DB 'To be, or not to be'
dst  DB 'To be,or not to be'
```

```
len  = $ - dst; len 19 ga teng
.CODE
....
cld; (DF) = 0
push ds; manzillarni joylashtirish
pop es; ds va es segmentlari
mov cx, len; dst qator uzunligini cx ga yuborish
lea di, dst; manzilini yuklash (tprkibtga joylashish) qatorlarni dst dan  DI ga
lea si,src
repe cmpsb;  baytlab solishtirish
je equal;agarda baytlar mos tushsa, unda belgilarga o'tish
not cx; agar yo'q bo'lsa -- mos tushmagan baytlarni hisoblash
add cx, len;
jmp notequal;
equal:  ....
....
notequal: ....
```

Ushbu masala oxirida mos kelmaydigan birinchi nomeri bayt (CX)=5 ga teng.

Miroprotsessorni boshqarish buyruqlari.

1. CLC: o'tkazish bayrog'ini tushirib yuborish (CF)=0. .
2. STC: o'tkazish bayrog'ini o'rnatish (SF)=1.
3. CMC: yo'nalish bayrog'ini inverslash.
4. CLD: yo'nalish bayrog'ini tozalash(DF)=0, bunday holatda qator (bayt zanjirlari ustida operatsiyalar) kichik adresdan katta adresgacha bajariladi.
5. STD: yo'nalish bayrog'ini o'rnatish (DF)=1, baytlar zanjirlari ustida ishlash katta adresdan kichigigacha bajariladi.
6. STT: urilish bayrogini o'rnatish (IF)=1, tashqi qurilmalardan uzilishni ruxsat berish.
7. CLT: uzilish bayrog'ini tozalash.
8. MOP: maxfiy operatsiya.

Uzilish buyruqlari.

1. INT INUM : bu komanda dasturli uzilishni chaqiradi, ya'ni to'rt baytdan saqlanayotgan adresni INUM *4 adresidan boshlab, bu yerda INUM=(0..:25) ga teng xotira yacheykasiga o'tish. Bu 4 baytli

son berilgan uzilishning qism dasturi qayta ishlovchining ko'rsatkichi bo'ladi va u boshqacha nomi uzilish vektori deb ataladi.

Dasturli uzilishlar yordamida vujudga keladigan operatsiyalar AN registridagi kod orqali aniqlanadi, masalan,

```
....
mov ah,14d; displeyga simvollarni chiqarish
mov al,31h; kursorni bitta o'ngga surish.
int 10h; ekranga '1' simvolni chiqarish(ASCII kod 31h).
....
.DATA
privet DB 'Xayrli tong!','$';
.CODE
....
lea dx,privet;DX registrning maxsus funksiyasi
mov ah,9;9 – displeyga chiqarish
int 21h;salomlashish chiqariladi ....
```

3. TARMOQ PROSESSORLARI

3.1. Tarmoq protsessorlarini rivojlanish bosqichlari

Kompyuter tarmoqlarida ma'lumotlar paketlarini qayta ishlash uchun mo'ljallangan apparat-dasturiy tizimlarning keng spektridan foydalaniladi. Ularga kommutatorlar, marshrutizatorlar, tarmoq adreslarini translyatsiya qilish protsessorlari, bostirib kirishlarni aniqlash tizimlari, brandmauerlar, ADSL-modemlar kiradi. Tarmoq tizimlari bozorga tezroq chiqish maqsadida narxi, fizik o'lchamlari va tayyorlash vaqtini cheklash sharoitlarida yuqori unumdorlik va keng funksional imkoniyatlar mezonlari bo'yicha loyihalashtiriladi.

Ishlab chiqiladigan tizimlar masshtablanadigan, yetarlicha universal bo'lgan va moslashuvchan bo'lishi kerak. Loyihalashtirish davomida bozor tendensiyalaridagi, qo'llaniladigan texnologiyalardagi va chiqarilayotgan tizimga qo'yiladigan texnik talablardagi tez yuz beradigan o'zgarishlarni hisobga olish kerak.

Bundan 40-yil avval kompyuter tarmoqlari paydo bo'lgan vaqtdan beri ular rivojlanishning bir necha bosqichlaridan o'tdilar. Birinchi avlod hisoblanadigan asosiy tarmoq qurilmalari (1975-yildan 1980-yillargacha) universal kompyuterlar bo'lgan edi. Marshrutizator IP-protokolni amalga oshiruvchi dasturga ega bo'lgan mini kompyuter bo'lgan edi. Ikkinchi avlodda (1990-95-yillar) paketlarni tasniflash va ba'zi boshqa funksiyalar uchun ixtisoslashtirilgan protsessorlar paydo bo'ldi, yuqori tezlikdagi kommutatorlar qo'llanila boshladi. Tarmoqda qayta ishlash funksiyalarining ko'pi universal protsessorlarda bajarilar edi. Tarmoq qurilmalarining uchinchi avlodi (2000 -yildan boshlab) universal protsessorga, har bir tarmoq interfeysi uchun ajratilgan protsessorlarga ega muammoli yo'naltirilgan JKIS (juda katta integral sxema) da amalga oshiriladigan to'liq taqsimlangan arxitekturaga ega. Bu avlod ma'lumotlarni ancha tez qayta ishlash bilan farqlanadi.

Uchinchi avlodda to'liq taqsimlangan arxitekturaga o'tilishi tarmoq tizimlarini ishlab chiqish va ularning xususiyatlarini

o‘rganish ishini murakkablashtirdi. Multiprotsessorli marshrutizatorlarda har bir tarmoq interfeysidagi har bir protsessor uchun alohida marshrut jadvallarini ta’minlash zarurati yangilanishda jadvallarning replikatsiyalarini sinxronlash uchun qo‘shimcha muammolarni yuzaga keltiradi. Bundan tashqari, marshrutizator marshrut jadvallarini yangilashni ma’lumotlar paketlarini qayta ishlash bilan birga olib borishi kerak.

Uchinchi avlod arxitekturalarida yangi muammolar yuzaga keldi: muammoli-yo‘naltirilgan JKIS asosida yaratilgan qurilmalarning baland narxi, yangi mahsulotlarni bozorga chiqarishning uzoq vaqt olishi (18-24 oy), paketli protsessorlarni testlash va verifikatsiya qilish murakkabligi, qo‘llaniladigan JKISning yetarlicha bo‘lmagan moslashuvchanligi.

JKISning moslashmaganligi loyihaga qo‘yiladigan talablardagi eng kam o‘zgarishlar holatida ham mikrosxemaning kattagina qaytadan loyihalashtirilishi uchun qo‘shimcha ravishda ko‘p vaqt ketishiga olib keladi. Bundan tashqari, yangi tarmoq tizimlarida foydalanish uchun JKISni modifikatsiya qilish va sozlash ishlab chiqish narxi va vaqtini ko‘paytiradi.

1990-yillarning oxirida kompyuter tarmoqlarining jadal rivojlanishi tarmoq tizimlaridagi tez o‘zgarishlarga va ularni yaratish uchun yangi yondashuvning qo‘llanishiga: tarmoq paketlarini qayta ishlash vazifalariga qaratilgan dasturlashtiriladigan protsessorlarni ishlab chiqishga olib keldi [18].

Arxitektura g‘oyasi birinchi avlod tarmoq tizimlarining uchinchi avlod tizimlarining yuqori tezligi bilan dasturlash kombinatsiyasidan iborat. Ko‘rsatilgan arxitekturaga ega qurilmalar *tarmoq protsessorlari* deb ataladi. Tarmoq protsessorlari yetarlicha unumdorlikka ega bo‘lishlari uchun tarmoq paketlarini qayta ishlash vazifalari batafsil tahlil qilingan bo‘lishi, ularning funksional dekompozitsiyasi bajarilgan, ularning vaqtli va sig‘im murakkabligi baholangan bo‘lishi kerak.

Eng ko‘p vaqt sig‘imiga ega vazifalar uchun tarmoq protsessorlari strukturasiga muammoli orientirlangan apparat yechimlarini kiritish kerak. Tarmoq protsessorlaridan foydalanish multiprotsessor qurilmalari orqali qayta ishlanadigan ko‘p tarmoq interfeyslari, paketlar oqimlaridan iborat tizimlarda iqtisodiy jihatdan foydalidir.

Yuqori unumdorlikka erishish uchun zamonaviy tarmoq protsessorlarida quyidagi tayanch arxitektura yechimlaridan foydalaniladi: yuqori taktli chastotaga ega bir oqimli protsessor, paketlar oqimlarini parallellash, paketlarni konveyerli qayta ishlash.

Hozirgi vaqtda birinchi arxitektura unumdorligining o'sish imkoniyatlari deyarli tugagan.

Parallel arxitekturaning xususiyati bo'lib, paketlarning kirish navbatini samarali boshqarish apparat mexanizmini tanlash va amalga oshirish zarurati hisoblanadi.

Konveyer arxitekturasida paketlar oqimi funksional bloklar liniyasi bo'yicha siljiydi, ularning har biri paketga talab qilinadigan qayta ishlashning o'z qismini bajaradi.

Zamonaviy tarmoq protsessorlarida kombinatsiyalangan konveyer-parallel yoki parallel-konveyer arxitekturalar qo'llaniladi.

Tarmoq protsessori ichida ikkita asosiy ma'lumotlar oqimi mavjud:

• yuqori tezlikda oddiy qayta ishlashni talab qiladigan paketlar;

• kamroq tezlikda murakkab qayta ishlashni talab qiladigan paketlar.

Protsessor ichida har bir oqim uchun maxsus apparat vositalar ajratilishi mumkin.

Tarmoq protsessorlaridan foydalanish mumkin bo'lgan o'zaro bog'lanishlarning uchta darajasi mavjud:

• yuqori tezlikda katta hajmdagi ma'lumotlarni uzatish uchun tayanch daraja. Bu yerda marshrutlash, kommutatsiyalash, foydalana olishni nazorat qilish funksiyalari amalga oshiriladi.

• chegaraviy tarmoqlararo daraja. Chegaralarda tayanch darajaga kirish va undan chiqish yuz beradi. Chegaraviy daraja funksiyalari yuqori murakkablik bilan farqlanadi, o'rtacha va yuqori tezlikda bajariladi, o'z ichiga marshrutlash, kommutatsiyalash, oqimni boshqarish, foydalana olishni nazorat qilish, xizmat ko'rsatish sifatini boshqarishni oladi.

• tarmoqga kirish darajasi. Bu daraja axborotni tarmoqga yetkazishning barcha nuqtalarini oladi. Foydalanuvchilar lokal tarmoqlar, keng polosali tarmoqlar (ma'lumotlar, video, tovush), telefon kanallari orqali Internetdan foydalana oladilar. Bu darajada nisbatan sekin ishlaydigan protokollar va texnologiyalar mavjud.

Protokollar va chiqariladigan tarmoq tizimlarining tahlili shuni ko'rsatmoqdaki, tarmoq protsessorlarining asosiy funksiyalari quyidagilar hisoblanadi:

• paketlar sarlavhalarini tahlil qilish, sarlavhalar maydonlarini shablonlar bilan qiyoslash, izlash va jadvallardan tanlab olish;

• kiruvchi paketlarni chiqish portlariga yo'naltirish;

• tarmoqqa kirishni nazorat qilish va paketlar navbatlarini boshqarish;

• paketlarning chiqish oqimini cheklash va chiqish trafigini boshqarish;

• uzatiladigan paketlarni modifikatsiya qilish.

Tarmoq protsessorlariga qo'yiladigan asosiy talablar:

• tarmoq bo'yicha ma'lumotlar uzatish tezligiga yaqin bo'lgan ma'lumotlarni qayta ishlashning yuqori tezligi ta'minlash;

• moslashuvchanligi va dasturlashtirilishi;

• bozorga chiqarish uchun tez ishlab chiqish sikli;

• foydalanuvchilarga xizmat ko'rsatish qulayligi.

Tarmoq protsessorlarining nazariyasi va amalga oshirilishi sohasidagi eng muhim tendensiyalar quyidagilar hisoblanadi:

• FPGA(Field Programmable Gate Array)ga rekonfiguratsiya qilinadigan arxitekturaga ega tarmoq protsessorlarini yaratish;

• ko'plab tarmoq protsessorlariga ega arxitekturalar uchun parametrlanadigan apparat platformalarini yaratish;

• paketlar oqimlarini filtrlash yo'li bilan ajratish va ajratilgan oqimlarni qayta ishlash yo'llarini optimallashtirish;

• tarmoq protsessorlari uchun operatsion tizimlarni ishlab chiqish;

• paketlarni qayta ishlashni tezlashtirish uchun asinxron arxitekturalarni qo'llash.

Tarmoq protsessorlarini loyihalashtirishdagi asosiy muammolar:

• berilgan tarmoq protokollari uchun paketlarni qayta ishlashning eng muhim vazifalarini aniqlash;

• protsessorlar arxitekturalarini optimallashtirish mezonlarini aniqlash;

• tez ishlashning oshirilishini ta'minlovchi funksional bloklarning tarkibi, strukturasi va o'zaro ishlash usullarini aniqlash;

• kiritish-chiqarish samarali interfeyslarini tanlash;

• dasturlar xotirasi va ma'lumotlar xotirasining optimal texnologiyalari va hajmlarini aniqlash;

• tarmoq protokollari funksiyalarini amalga oshirish usullarini tanlash (apparat amalga oshirilishi, dasturiy amalga oshirish, muammoli orientirlangan interfeysli JKISni yoki soprotsessorlarni qo'llash);

• dasturlashni avtomatlashtirilgan instrumental vositalarini ishlab chiqish (dasturlash tillari, kompilyatorlar, assemblerlar, aloqalar redaktorlari, yuklovchilar, funksiyalar kutubxonalari).

Qurilmalarning mustaqil sinfi sifatida tarmoq protsessorlari (TP) 1990-yillarning oxirlarida paydo bo'ldi. Ularning yaratilishiga kompyuter tarmoqlarida ma'lumotlar oqimlari hajmlarining ko'payishi va shu bilan bog'liq tarmoq qurilmalarning o'tkazuvchanlik qobiliyatiga qo'yiladigan talablarning oshishi sabab bo'ldi. O'tgan o'n-yil davomida TP, asosan, unumdorlikning oshishi va funksionallikning kengayishi yo'nalishida rivojlandi.

Tarmoq protsessorlariga qo'yiladigan asosiy talab, marshrutizatorni ulangan kanal tezligida paketlar oqimini qayta ishlash qobiliyati hisoblanadi. Shuni qayd etib o'tish kerakki, marshrutizatorlar ma'lumotlar uzatish tezligi sekundiga yuzlab gigabitdan oshib ketadigan magistral tarmoqlarda, hamda oddiy foydalanuvchilarning o'nlab Mb/s dan bir necha Gb/s gacha bo'lgan kirish tarmoqlarida qo'llanishi mumkin. Bunda turli tarmoqlarning turli turdagi tarmoq protsessorlaridan foydalaniladi.

Tarmoq protsessorlarining umumiy vazifadagi protsessorlardan asosiy farqlari sifatida quyidagilarni ko'rsatib o'tish kerak:

- ko'pchilik tarmoq protsessorlarining yo'riqnomalari to'plami RISC-arxitekturaga asoslangan;

- tarmoq protsessorlarining arxitekturalari bitli operatsiyalar, nazorat summalari va izlash operatsiyalari uchun qo'shimcha yo'riqnomalarga ega.

Tarmoq protsessorlari paketlarni qayta ishlash vazifalarini amalga oshiruvchi qo'shimcha funksional bloklardan iborat bo'lishi mumkin.

3.2. Tarmoq protsessorlarining umumiy strukturasi va vazifalari

Tarmoqqa turli xil qurilmalar ulanadi. Foydalanuvchilar uchun bu, avvalambor, shaxsiy kompyuterlar (stol usti va noutbuklar), lekin, shu bilan birga, o'yin konsollari, shaxsiy elektron kotiblar (cho'ntak kompyuterlari), uyali telefonlar soni ham o'sib bormoqda. Kompaniyalar uchun yakuniy tizimlar rolini serverlar va shaxsiy kompyuterlar o'ynaydi. Bundan tashqari, tarmoqlarda son-sanoqsiz turfa xil oraliq qurilmalar ishlaydi, ularning sirasiga marshrutizatorlar, kommutatorlar, brandmauerlar, proksi-serverlar, yuklamani muvozanatlash tizimlari kiradi. Shunisi qiziqarliki, bu oraliq tizimlarga eng jiddiy talablar qo'yiladi – aynan ular bir sekundda maksimal miqdordagi paketlarni yuborishni ta'minlashi lozim. Bundan tashqari, serverlarga ham jiddiy talablar qo'yiladi.

Tarmoq va paketning o'ziga qarab, tarmoqqa kelib tushayotgan paket chiquvchi liniya orqali jo'natilishidan yoki amaliy dastur orqali taqdim etilishidan oldin u yoki bu ishlov berishni talab etishi mumkin. Ishlov berish quyidagilarni o'z ichiga olishi mumkin: paketni qaerga yuborish, paketni qismlarga bo'lish yoki uni qismlardan yig'ish, xizmat ko'rsatish sifatini boshqarish (ayniqsa audio- va video-oqimlarga nisbatan), ma'lumotlarni himoyalash (kodlash va dekodlash), kompressiya va dekompressiya va sh .k.

Lokal tarmoqda ma'lumotlar jo'natish tezligi 40Gbit/s.ga, paket hajmi esa – 1 Kbaytga yaqinlashsa, tarmoq kompyuteri sekundiga qariyb 5 mln.ta paketni qayta ishlashi kerak bo'ladi. 64 baytli paketlar uchun bu miqdor taxminan sekundiga 80 mln. paketgacha o'sadi. Barcha zikr etib o'tilgan fuknsiyalarning berilgan vaqt ichida o'rtacha 12-200 ta bajarilishi, buning ustiga majburiy ravishda paketlardan nusxa olinishi, dasturiy jihatdan umuman amalga oshirib bo'lmaydi. Apparatli qo'llab-quvvatlash bu yerda juda zarurdir.

Paketlarga tezkor ishlov berishni apparatli hal etish yo'llaridan biri ixtisoslashtirilgan integral sxemalar (Application-Specific Integrated Cicuit, ASIC) dir. Bunday mikrosxema oldindan ko'zda tutilgan barcha amallarni bajara oluvchi apparat tomonidan amalga oshirilgan dasturga o'xshaydi. Ko'pgina zamonaviy marshruzitorlarning asosi ASIC sxemalari sanaladi. Shu bilan birga, ixtisoslashtirilgan integral

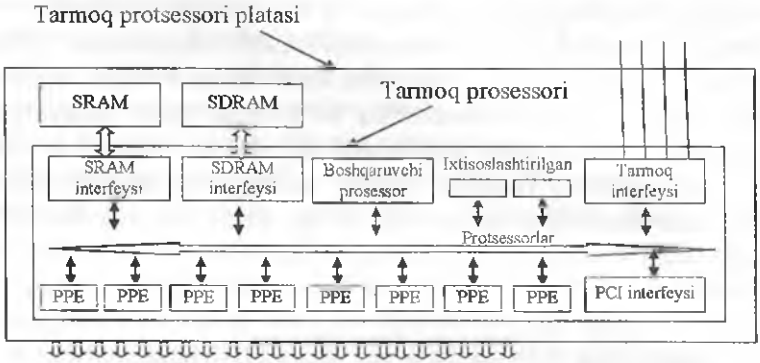
sxemalar bilan ham bir qancha muammolar bog‘liq. Avvalambor, ularni loyihalashtirish uzoq vaqtni oladi, ishlab chiqarish ham undan tez bo‘lmaydi. Bundan tashqari, bu – qat’iy dasturlangan qurilma, ya’ni yangi funktsionallik kiritish uchun yangi mikrosxemani ishlab chiqish va yaratish talab etiladi. Eng yomoni, xatolar haqiqiy dahshatning o‘zginasi. Chunki ularni tuzatishning yagona yo‘li yangi (tuzatilgan) mikrosxemani ishlab chiqish, tayyorlash va o‘rnatish hisoblanadi. Va nihoyat, bu yondashuv o‘ta serxarajat sanaladi.

Ikkinchi yondashuv dasturlanuvchi ventil matritsalarni ishlatishga asoslangan (Field Programmable Gate Array, FPGA). Bunday matritsa o‘zida qayta kommutatsiya yo‘li bilan talab etilayotgan sxema quriladigan ventillar to‘plamini aks ettiradi. Dasturlanuvchi ventil matritsalarning bozorga chiqish muddati, ixtisoslashtirilgan integral sxemalarnikiga qaraganda, ancha qisqa, bundan tashqari, ularning «dala sharoitlari»da maxsus dasturlagich yordamida qayta dasturlash mumkin. Ammo, shu bilan birga, ular ASIC ga qaraganda anchayin murakkab, qimmat va sekinroq, shu sababli dasturlanuvchi ventilli matritsalar bir nechta tor ixtisosli sohalarni hisobga olmaganda, unchalik keng ommalashmagan.

Va nihoyat, tarmoq protsessorlari – kiruvchi va chiquvchi paketlarni ularning uzatilish tezligida, ya’ni real vaqtda qayta ishlay oladigan qurilmalarga o‘tamiz. Odatda ular tarmoq protsessori kristalidan tashqarida, xotira va yordamchi mantiqni o‘z ichiga olgan, olib-qo‘yiladigan plata ko‘rinishida chiqariladi [20-21]. Plataga bir yoki bir nechta tarmoq liniyalari ulanadi. Liniyadan protsessor paketlarni qabul qiladi, ularni qayta ishlaydi, keyin, agar bu marshrutizator bo‘lsa – boshqa liniya orqali jo‘natadi, yakuniy qurilma (masalan, shaxsiy kompyuter) bo‘lsa – asosiy tizim shinasi (ya’ni PCI shinasi)ga yuboradi. Odatiy tarmoq protsessori va uning platasi 3.1-chizmada ko‘rsatilgan.

Odatda platada statik (SRAM) bilan bir qatorda dinamik tezkor xotira (SDRAM) ham mavjud bo‘ladi – bu xotira turlari turli maqsadlarda qo‘llaniladi. SRAM xotirasi SDRAM ga qaraganda tezroq ishlaydi, lekin qimmat bo‘lgani tufayli bu turdagi xotira odatda kam uchraydi. U marshrutizatsiya jadvallarini va boshqa asosiy ma’lumotlar tuzilmalarini saqlash uchun ishlatiladi. SDRAMda esa, bevosita qayta ishlanadigan paketlar yoziladi. Ushbu ikkala turdagi

xotira tarmoq protsessori kristallidan tashqarida joylashganligi tufayli, xotira hajmi masalasiga keng yondashish mumkin. Chunonchi, yagona tarmoq liniyasiga ega oddiy tizimlarda (bunday platalar, misol uchun, shaxsiy kompyuter yoki serverlarga qo'yilishi mumkin) xotira ko'p bo'lmasligi mumkin, ammo marshrutizatorga esa, ancha ko'proq xotira talab etiladi.



3.1-chizma. Odatiy tarmoq protsessorining kristali va platasi

Tarmoq protsessorlari katta hajmdagi kiruvchi va chiquvchi paketlarni tez qayta ishlash uchun optimallashtirilgan. Bu esa, tarmoq liniyalarining har biri orqali sekundiga millionlab paketlar o'tishini, marshrutizator esa, o'nlab bunday liniyalarni qo'llab-quvvatlashi lozimligini anglatadi. Bunday jiddiy ko'rsatkichlarni faqat ichki parallelizmi yuqori darajada bo'lgan protsessorlarda erishish mumkin. Bundan tashqari, protsessor tarkibida albatta bir nechta RRE-kotrollerlar (Protocol/Programmable/Packet Processing Engine – paketlar va protokollarni qayta ishlashning dasturlanadigan tizimi) mavjud bo'ladi, ularning har biri RISS-yadro va dastur hamda bir qancha o'zgaruvchilarni saqlash uchun unchalik katta bo'lmagan ichki xotiradan iborat bo'ladi.

RRE-kontrollerlarni tashkil etishning ikkita yondashuvi bor. Eng sodda holatda barcha RRE-kontrollerlar bir xil ishlanadi. Tarmoq protsessoriga yangi paket kelganida, u ushbu paytda bo'sh turgan RRE-kontrollerga ishlov berish uchun beriladi. Agar bo'sh RRE-

kontrollerlar bo‘lmasa, paket platadagi SDRAM xotirasiga navbatga qo‘yiladi va RRE-kontrollyerdan birining bo‘shashini kutadi. Bunday tashkil etishda 3.1-chizmada ko‘rsatilgan gorizontal aloqalar bo‘lmaydi, chunki turli RRE-kontrollerlarning o‘zaro aloqada bo‘lishi shart emas.

RRE-kontrollerlarni tashkil etishdagi boshqa yondashuv – konveyer bo‘lib, unda har bir RRE-kontroller qayta ishlashning bir bosqichini bajaradi, keyin olingan paketga ko‘rsatkichni konveyerdagi keyingi RRE-kontrollerga uzatadi. Bunday konveyer konveyerlarga o‘xshash MPlarda ishlaydi, ular II bobda ko‘rilgan edi. Ikkala talqindagi tashkil etishda RRE-kontrollerlar to‘liq dasturlanuvchidir.

Yanada takomillashgan tarmoq protsessorlarida RRE-kontrollerlar ko‘p oqimlikni qo‘llab-quvvatlaydi. Ya’ni ularning har biri bir nechta registrlar to‘plamiga va qaysi to‘plam ishlatilayotganligini ko‘rsatuvchi apparatli registrga ega. Bu esa, bir vaqtning o‘zida bir nechta dasturlami (ya’ni dasturlar oqimini) bajarish va shunchaki «registrlarning joriy ishchi to‘plami» o‘zgaruvchisini o‘zgartirish orqali ularning biridan ikkinchisiga o‘tish imkonini beradi. Dasturiy oqimlardan biri kutib qoladigan bo‘lsa (masalan, bir necha sikllarni talab etuvchi SDRAM ga murojaat amalga oshirilganda), RRE-kontroller bir lahzada ishni davom ettira oladigan oqimga o‘ta oladi. Bu hol, SDRAM bilan ma’lumotlar almashish yoki boshqa sekin ishlovchi tashqi amallarning tugashini tez-tez kutishga to‘g‘ri kelishiga qaramay, RRE-kontrollerlarning yuqori darajada yuklanganligiga erishishni ta’minlaydi.

RRE-kontrollerlardan tashqari, barcha tarmoq protsessorlarida bevosita paketlarni qayta ishlashga oid bo‘lmagan harakatlar (masalan, marshrutizator jadvallarini yangilash)ni bajarish uchun boshqaruvchi protsessor mavjud. Odatda u umumiy maqsadlardagi RISC-protsessor bo‘lib, undagi ma’lumotlar va buyruqlar uchun xotira protsessor bilan bitta kristalda joylashgan bo‘ladi. Bundan tashqari, tarmoq protsessorida o‘ta muhim operatsiyalarni bajarishga mo‘ljallangan bir nechta ixtisoslashtirilgan protsessorlar bo‘lishi ham mumkin. Ular juda kichkina ixtisoslashtirilgan integral sxemalar (ASIC) bo‘lib, marshrutizatsiya jadvalida maqsadli manzilni izlash kabi bitta yengil harakatni amalga oshirishga qodir bo‘ladi. Tarmoq protsessorining barcha komponentlari o‘zaro multigigabayt tezlikda,

kristalda joylashgan bir yoki bir nechta parallel shinalar bo'ylab ta'sirlanadilar.

Paketlarga ishlov berish. Protsessorda konveyerli yoki parallel yondashuv bo'lishidan qat'i nazar, har bir kelgan paket bir nechta qayta ishlash bosqichidan o'tadi [22]. Ba'zi protsessorlarda bu bosqichlar kirish (ingress processing) va chiqish (egress processing) qayta ishlashga bo'linadi. Birinchi guruhga tashqaridan (tarmoq liniyasi yoki tizim shinasi orqali) kelgan paketlar bilan operatsiyalar kiradi. Ikkinchisiga esa – paketlarni jo'natishdan oldin qayta ishlash kiradi. Shu tariqa, har bir paket oldin kirish qayta ishlashga, keyin chiqish ishloviga duch keladi. Bu taqsimlanish anchayin shartli, chunki ba'zi operatsiyalarni bu bosqichlarning istalganida bajarsa bo'ladi (masalan, trafik to'g'risida ma'lumotlar yig'ish).

Biz bu bosqichlarni ular amalga oshirilishi mumkin bo'lgan tartibda ko'rib chiqamiz. Ammo shuni inobatga olingki, ularni bajarish barcha paketlar uchun ham shart emas, bundan tashqari, harakatlarning boshqa ketma-ketligi ham mumkin.

1. Nazorat summasini tekshirish. Agar kiruvchi paket Internet tarmog'idan kelayotgan bo'lsa, paket bexato qabul qilinganligiga ishonch hosil qilish uchun uning nazorat summasi (CRC-kod) sanab chiqiladi va paketdagi qiymat bilan qiyoslanadi.

Agar ikkala qiymat teng bo'lsa yoki Internet-paketda CRC maydoni mavjud bo'lmasa, IP-paket nazorat summasi hisoblab chiqiladi va paketdagi qiymat bilan qiyoslanadi. Bu ish jo'natuvchi tomonidan IP-paket uchun nazorat summasi hisoblab chiqilganidan so'ng jo'natuvchi xotirasidagi xato bit aybi bilan IP-paket zararlanmaganligiga amin bo'lish imkonini beradi. Agar barcha nazoratlardan o'tilsa, paketga ishlov berish davom ettiriladi, aks holda u tashlab yuboriladi.

2. Maydonlar qiymatlarini chiqarib olish. Tahlil yo'li bilan kerakli sarlavha holati aniqlanadi va paketdan kalit maydonlarning ushbu sarlavhasiga mos keluvchi qiymatlar chiqarib olinadi. Ethernet-kommutatorda faqat Ethernet-sarlavhalar ko'riladi, IP-marshrutizatorda – IP-sarlavhalar. Kalit maydonlar qiymatlari yo registrlarda (RRE-kontrollerlarning parallel tashkil etilishida), yoki SRAM da (konveyerli tashkil etishda) saqlanadi.

3. Paketlar tasniflanishi. Paketlar dasturiy qoidalar qatoriga muvofiq tasniflanadi. Eng sodda holatda ma'lumotlar paketlari boshqaruvchi paketlardan ayriladi, lekin, odatda, taqsimlash ancha nozik bo'ladi.

4. Yo'l tanlash. Aksariyat tarmoq protsessorlari ma'lumotlar paketining butun turfa xilligini yetkazish uchun optimallashtirilgan maxsus tezkor yo'lga ega bo'ladilar. Bunda boshqa paketlar o'z holicha, odatda boshqaruvchi protsessor tomonidan qayta ishlanadi. Tegishlicha, yo tezkor yo'l, yoki sekin yo'llardan biri tanlanishi kerak bo'ladi.

5. Maqsadli tarmoqni aniqlash. IP-paketlarda qabul qiluvchining 32-razryadli manzili mavjud bo'ladi, biroq qabul qiluvchini izlash uchun 232 yozuvli butun jadvalni ishlatish imkonsiz (va nomaqbul) dir. Shu sababli manzilning chap tomonida odatda tarmoq manzili mavjud bo'ladi, o'ng tarafi esa, ushbu tarmoqdagi alohida mashinaga ishora qiladi. Tarmoq manzili uzunligi qat'iy chegaralanmagan, shu sababli uni aniqlash oson ish emas, buning ustiga, bir qancha variantlarning borligi, ular ichida eng uzuni eng to'g'risi hisoblanishi bu vazifani yanada qiyinlashtiradi. Bu qadamda aksari hollarda ixtisoslashtirilgan integral sxema qo'llaniladi.

6. Marshrutni izlash. Maqsadli tarmoq manzili SRAM xotirasidagi jadvaldan aniqlangach, chiquvchi liniyalarning qaysi biridan paketni jo'natish kerakligi ma'lum bo'ladi. Ushbu qadamda ham ixtisoslashtirilgan integral sxema qo'llanilishi mumkin.

7. Taqsimlash va yig'ish. Ilovalar ko'p hollarda TSR-paketlarning foydali yuklamasi (ma'lumotlar)ni maksimal darajada ko'paytiradi, bu bilan tizimli chorlovlarning sonini qisqartirishga harakat qiladilar. Lekin TSRda ham, IPda ham, Ethernetda ham paket maksimal hajmiga nisbatan cheklov mavjud. Ushbu cheklovlar oqibati o'laroq, paketlar (tegishlicha, foydali yuklama)ni jo'natishdan oldin qismlarga taqsimlash va qabul qiluvchi tomonda qaytadan yig'ish talab etilishi mumkin. Bu funksiyalarni tarmoq protsessoriga yuklash mumkin.

8. Hisoblash. Ba'zida ma'lumotlar ustida u yoki bu murakkab hisoblarni amalga oshirish kerak bo'ladi, masalan, kompressiya va dekompressiya, kodlashtirish va dekodlashtirish. Bu harakatlarni tarmoq protsessoriga yuklash mumkin.

9. Sarlavhalarni boshqarish. Ba'zida sarlavhalarni qo'shish yoki olib tashlashga, shuningdek, u yoki bu maydon qiymatlarini o'zgartirishga to'g'ri keladi. Masalan, IP-sarlavhada paket o'zini-o'zi yo'q qilishidan oldin o'ta oladigan hop miqdori hisoblagichi mavjud. Har bir hopdan o'tilgandan keyin hisoblagich qiymatini 1 ga kamaytirish kerak, bu funksiyani tarmoq protsessori bemalol bajara oladi.

10. Navbatlarni boshqarish. Kiruvchi va chiquvchi paketlar ishlov berishni kutib tez-tez navbatda turishlariga to'g'ri keladi. Lekin multimediali ilovalar uchun jitterga yo'liqmaslik maqsadida paketlar o'rtasidagi ushlanib qolishlar muayyan qiymatdan oshmasligi talab qilinadi. Bundan tashqari, brandmauer yoki marshrutizatordan kiruvchi yuklamani bir nechta chiquvchi liniyalar o'rtasida muayyan qoidalarga ko'ra qayta taqsimlash talab etilishi mumkin. Bu vazifalarning barchasi tarmoq protsessori tomonidan hal etilishi mumkin.

11. Nazorat summalarini jamlash. Chiquvchi paketlarda nazorat summalari bo'lishi kerak. IP-paketlar nazorat summasi tarmoq protsessori tomonidan hisoblanishi mumkin, Ethernet-paketlar nazorat summasi umumiy holatda apparatli jamlanadi.

12. Hisobga olish. Ba'zi hollarda paketlar o'tayotganda trafikni hisobga olish zarur bo'ladi, ayniqsa tarmoqlardan biri tijorat xizmati sifatida trafik tranzitini taqdim etayotgan bo'lsa. Hisobga olish bilan tarmoq protsessori shug'ullanishi mumkin.

13. Statistikani yig'ish. Ko'pgina kompaniyalar trafik statistikasiga ega bo'lishni istaydilar va tarmoq protsessorlari bu ma'lumotlarni yig'ishi mumkin.

Unumdorlikni oshirish. Unumdorlik – bu tarmoq protsessorlarining eng asosiy xarakateristikasidir. Uni oshirish uchun nima qilish mumkin? Bu savolga javob berishdan oldin, uning nima ekanligini aniqlab olish kerak. O'lchovlardan biri bir sekundda jo'natiladigan paketlar soni bo'lsa, boshqa o'lchov – bir sekundda jo'natiladigan baytlar sonidir. Bu o'lchovlar turli yondashuvlarni talab qiladi va kichik paketlar bilan yaxshi ishlovchi sxema katta paketlarni eplay olmay qolishi mumkin. Jumladan, kichik paketlarni jo'natishda jadvalda maqsadli manzilni izlash jarayonini jadallashtirish orqali unumdorlikni oshirish mumkin. Lekin katta paketlarni

jo'natishda bu ish unumdorlikning sezilarli o'sishini ta'minlay olmaydi.

Unumdorlikni o'stirishning oddiy yo'li – bu tarmoq protsessorining takt chastotasini oshirish sanaladi. To'g'ri, unumdorlik chastotaga parallel ravishda o'smaydi, chunki xotiraga murojaat qilishga ketgan vaqt va boshqa omillar o'z ta'sirini ko'rsatadi. Bundan tashqari, katta chastota ko'proq issiqlik chiqarish zaruriyatini anglatadi.

Odatda buning yechimi RRE-kontrollerlar sonini oshirish sanaladi – bu yondashuv parallel arxitekturalar uchun ayniqsa samaralidir. Konveyer uzunligini oshirish ham yordam berishi mumkin, lekin bu faqat paketni qayta ishlash jarayonini yetarlicha sodda bosqichlarga bo'lishga erishilganda ro'y beradi.

Yana bir yondashuv alohida xarajatli va tez-tez kerak bo'lib turiladigan operatsiyalarni bajarish uchun mo'ljallangan qo'shimcha ixtisoslashtirilgan protsessorlar yoki ixtisoslashtirilgan integral sxemalar sonini oshirishdan iborat, agar bunday operatsiyalarni apparatli bajarish samaraliroq bo'lsa. Ko'plab nomzodlar orasidan jadvaldan izlash, nazorat summalarini hisoblash va kriptografik operatsiyalarni qayd etish mumkin.

Shuningdek, qo'shimcha shinalarni kiritish va mavjudlaridagi liniyalar sonini oshirish hisobiga tizimdagi paketlar o'tishi vaqtini qisqartirib, tezlikni oshirish ham mumkin. Va nihoyat, odatda unumdorlikning o'sishiga xotira mikrosxemalarini o'zgartirish (SDRAM o'rniga SRAM) bilan ham erishish mumkin. Lekin bu, tabiiyki, narxning o'zgarishiga olib keladi.

3.3. Tarmoq protsessorlarining zamonaviy arxitekturalari

3.3.1. Internet Exchange arxitekturasi (IXA)

IX arxitekturasi turli maqsadli tarmoq qurilmalarini – korxona darajasidagi lokal tarmoq (LAN) uchun marshrutizatorlardan tortib hududiy taqsimlangan tarmoqlar (WAN) va ko'p funksiyali marshrutizatorlargacha yaratish uchun yagona universal ishlab chiqaruvchi baza bo'ladi [9,19].

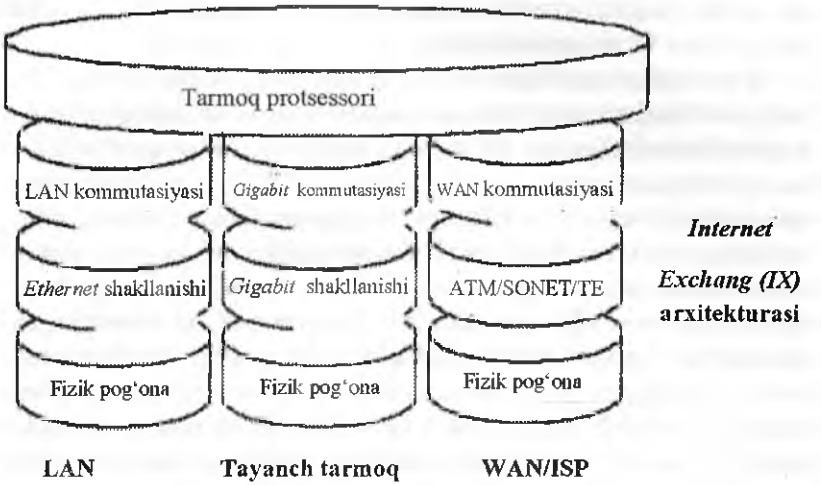
IXA (Internet Exchange Architecture) – telekommunikatsiya bozori uchun apparat va dasturiy ta'minot yaratish arxitekturasi. IXP12xx turidagi protsessorlar Intel kompaniyasi tomonidan IXA asosida ishlab chiqilgan tarmoq protsessorlari oilasida yetakchi hisoblanadi.

IXA keng ko'lamli qurilmalarda bevosita katta oqimli trafikni kommutatsiya/marshrutizatsiya qilish, protokollarni konvertatsiya qilish, QoS ni ta'minlovchi texnologiyalarni joriy etish, trafikni filtrlash, Firewall va VPN ni joriy etish, yuklamani boshqarish va bir qancha boshqa (masalan, IXP protsessori bazasidagi tizimlar ATM va Ethernet trafiklarini konvertatsiya qilish va orqaga qaytarish kabi masalalarni real vaqt masshtabida bajara olish holatida bo'ladi) masalalarini yechish uchun qo'llanishi mumkin.

Internet Exchange arxitekturasi (3.2-chizma) quyidagi komponentlarni o'z ichiga oladi:

- IXP seriyasidagi tarmoq protsessorlari;
- ATM, Gigabit va 10/100Mb Ethernet tarmoqlari uchun IXE seriyasidagi ilovalarning kommutatsion kontrollerlari. Bu yuqori samarali qurilmalar tizim ishlab chiquvchilarga o'z mahsulotlarini bozorga chiqarish vaqtini qisqartirish va uni yangi umumqabul qilingan standartlarga to'liq mos kelish imkoniyatini beradi;
- ATM yacheykalarini, T1/E1 va Sonet/SDH freymlarini, Gigabit i 10/100Mb Ethernet ma'lumotlar paketlarini shakllantirish uchun IXF seriyali shakllantiruvchi qurilmalar;
- T1/E1, HDSL, HDSL2, Sonet/SDH PHYs, 10/100Mb Ethernet, Gigabit (shuningdek, misli texnologiyalarga asoslangan) standartlarida ishlovchi tarmoq qurilmalarini bog'lash uchun LXT seriyali fizik sath komponentlari. Level OneTM kompaniyasi tomonidan ishlab chiqarilgan qurilmalarga nisbatan bu komponentlar o'ta yuqori integratsiya darajasi, kichik energiya sig'imi va yuqori samaradorligi bilan farqlanadi;
- IX platformasida loyihalashtirish muhiti.

IX arxitekturasi boshqa konstruktorlik usulidan tizimli yechimlarni yaratish uchun platforma bo'lib xizmat qilish imkoniyati bilan farqlanadi: samaradorlik keng diapazonda variatsiyalanadi, LAN i WAN tur protokollarini qayta ishlashga qodir va uning tarkibiga ishlab chiqishning yuqori sinfli instrumental vositalari to'plami kiradi.



3.2-chizma. IXA platformasida proektlash muhiti

IX arxitekturasi – tarmoq sanoatidagi barcha uchta talabga mos keluvchi arxitekturalardan biridir.

Ovoz va ma'lumot uzatish tarmoqlari birlashishi (konvergensiyasi), ochiq va xususiy tarmoqlarning kesishuvi natijasida ishlab chiqaruvchilardan ochiq standartlar asosida yaratilgan, bir vaqtda bir necha xizmatlarni qo'llashga qodir qurilmalarni yaratish talab etilmoqda. Xizmat ko'rsatuvchilarning intellektuallikka va tarmoq egiluvchanligiga bo'lgan talablari oshib borishi bilan qayta dasturlanuvchi protsessorlarning ahamiyati ortib bordi.

Hozirgacha tarmoq qurilmalarini ishlab chiqaruvchilar an'anaga binoan buyurtma (ASIC) asosida tayyorlanuvchi maxsus integral sxemalarni yaratganlar. Soha korxonasida u yoki bu marshrutizator seriyasini yaratish uchun shunday mikrosxemalarni yaratish uchun buyurtma berishganki, keyinchalik bu mikrosxemalarni faqat shunday seriyali qurilmalar uchun ishlatish mumkin bo'lgan.

Internet infrastrukturasining zamonaviy tizimli talablarini qondira oluvchi yechim har biri ma'lum bir protokol bilan ishlashga mo'ljallangan alohida komponentlarni yaratish va ularning har biri

universal, yuqori samarador tarmoq protsessorlari bilan birgalikda ishlay olishi bilan tushuntiriladi.

Tarmoqdan uzatilgan bitta ma'lumotlar paketi ustida bajariladigan o'rtacha komandalar soni oddiy xizmatlar (kommutatsiya va marshrutizatsiya) uchun 100 tadan 2 mingta operatsiyagacha oshdiki, bu operatsiyalar virtual xususiy tarmoqlar (VPN) da ishlash, viruslar va ruxsat etilmagan kirishlardan himoyadan iborat. Tarmoq xizmatlari borgan sari murakkablashib borayotganligi uchun ma'lumotlarga ishlov berish intensivligini oshirish zarur. Shu bilan bir vaqtda yangi xizmatlarni joriy qilishga talab oshib bormoqda, bu xizmatlar aloqa operatorlari qo'shimcha daromadlarining manbai hisoblanadi. Bu barcha talablarni qondirish uchun maxsus integral sxemalarning yaratilishi kamlik qiladi. Qoida bo'yicha IXP tarmoq protsessorlari asosidagi tayyor yechimlar bozoriga chiqishda maxsus integral sxemalar asosida analogik qurilmani yaratishga qaraganda 2 marta kam vaqt sarflanadi. Bu protsessorlar dasturlanuvchi hisoblanadi (ekspluatatsiya joyida ularni dasturiy modernizatsiya qilish imkoniyati mavjud), demak ular bozorda uzoq vaqt hukmronlik qiladi.

Internet Exchange Architecture ning asosiy g'oyalaridan biri topshiriqlarni (vazifalarni) bajarishda qayta dasturlanuvchi protsessorlardan foydalanish hisoblanadi.

Umuman IXA OEM ishlab chiqaruvchilarga va dasturiy ta'minotni mustaqil yetkazuvchilarga talab qilingan murakkab xizmatni ko'rsatishni ta'minlaydigan qurilmani yaratishga imkon beradigan strukturali elementlar (tarmoq protsessorlarini o'z ichiga oluvchi dasturiy va apparat ta'minoti)ning butun to'plamidan iborat.

Qayta dasturlanuvchi yarim o'tkazgichli elementlardan, shu jumladan, IXP protsessorlaridan foydalanish telekommunikatsiya qurilmalarini ishlab chiqaruvchilariga moliyaviy va vaqt resurslarini tejash imkonini beradi.

IXA turli protsessorlarni dasturlash uchun bir xil dasturiy vositalar (operatsion tizim, protokollar to'plami va h.k.)dan foydalanish imkonini beradi. Arxitekturani ishlab chiquvchilar bu yondoshuv dasturiy ta'minot yetkazib beruvchilar uchun ham, OEM-ishlab chiqaruvchilar uchun ham afzallikka ega ekanligini tasdiqlaydilar.

IXA doirasida boshqa elementlar ham ishlab chiqilgan:

* IXC – boshqaruv sathi protsessorlari (signalizatsiya va bog'lanish, marshrutizatsiya jadvallarini boshqarish);

* IXS – media-potoklarni qayta ishlovchi protsessorlar (ovoz trafikini kommutatsiyalanadigan tarmoqdan PK tarmoqqa o'tkazish);

* IXF – elementlar – kanal sathiga kirish kontrollerlarini yaratish.

Har bir toifa protsessorlari bir necha modifikatsiyalarga ega ma'lum oilaga tegishli. IXP protsessorlar oilasida beshdan ortiq modellar keltirilgan, uning asosiy farqi samaradorligi belgilanadi, o'z navbatida turli telekommunikatsiya qurilmalarini yaratishda afzalliklarga ega. 12xx protsessorlari IXP tarmoq protsessorlarining oddiy oilasiga tegishli.

IXP12xx dastlabki ishlab chiqaruvchi kompaniya tomonidan ishlab chiqarilganligiga qaramay keyingi protsessor versiyalari bilan hamkorlikda ishlay oladi.

Intel kompaniyasi IXA doirasida faqatgina protsessorlarnigina emas, balki aniq loyihalarni (SDK) joriy qilish uchun zarur dasturiy ta'minotni ishlab chiqish bilan ham shug'ullanadi. Shuningdek, bozorda qurilma imkoniyatlaridan maksimal foydalanishga imkon beruvchi operatsion tizim va muhitlar to'plami mavjud. Ularni doimiy ravishda mukammallashtirilib boriladi.

IXP1200 tarmoq protsessorlari.

IX arxitekturasining kalit elementli bo'lib kommutatsiyalovchi va shakllantiruvchi qurilmalarning butun spektri, shuningdek, fizik sath komponentlariga mos keluvchi IXP1200 tarmoq protsessori hisoblanadi. IXP1200 tarmoq protsessorining IX shinasi boshqaruvni ta'minlash maqsadida undan IXE kommutatsiyalovchi qurilmasini ajratib, tarmoqning murakkab funksiyalarini bajarishga imkon beradi (masalan, shifrlashni nazorat va xizmat ko'rsatish sifatini ta'minlash).

IXP1200 tarmoq protsessoriga barcha turdagi tarmoqlardan uzatilayotgan ma'lumotlar paketiga ishlov berish, marshrutlash funksiyasi yuklatilgan. IXP1200 tarmoq protsessori ma'lumotlar uzatish jarayonida hisoblashlarni, shuningdek, tarmoq tizimini boshqarishning ma'lum amallarini bajarish uchun yetarli quvvatga ega.

IXP1200 tarmoq protsessori o'zida ko'pgina tarmoq tizimlarida muhim rol o'ynovchi ikkita hisoblash komponentini birlashtirgan:

ichki mikroprotsessor va kabel translyatsiyasiga teng tezlikli ma'lumot uzatish kontrollerlari.

StrongARM ichki mikroprotsessori ARM® ning 32 razryadli arxitekturasi mos keladi, tarmoqni boshqarish amalini bajarish uchun xizmat qiladi, 6 ta dasturlanadigan mikrokontrollerlar kabel translyatsiyasi tezligida tarmoqdan o'tayotgan ma'lumotlarga ko'p potokli ishlov beradi. Bir vaqtda 7 ta turli tarmoq amalini bajarish mumkin, boshqa 18 tasi esa bir sikl davomida bajarish uchun navbatga qo'yiladi. Bu jarayon maxsus instruksiyalangan mikrokontrollerlar vositasida ta'minlanadi.

Har bir IXP1200 protsessori sekundiga 3 mln. paketni yo'naltirish imkoniyatiga ega, bir nechta protsessorlarni birlashtirish natijasida samaradorlikni 1,5 Tbit/s ga chiqarish mumkin.

IXP1200 Gigabit Ethernet, Sonet i ATM kabi ma'lumot uzatish protokollarini qo'llovchi marshrutizator, kontsentratorlarni yaratishda foydalanilishi mumkin.

IX arxitekturasidan foydalanishning afzalliklari.

Qisqa muddatda tayyor mahsulot ishlab chiqish.

Apparat va dasturiy vositalarni bir vaqtda yaratish imkoniyati IXP1200 tarmoq protsessori asosidagi tizimni loyihalash muddatini keskin kamaytiradi. Bundan tashqari, OEM-ishlab chiqaruvchilar ishlab chiqarish jarayonida IETF, IEEE va ITU tashkilotlari tomonidan Internet standartlariga uzluksiz kiritilayotgan o'zgarish va to'ldirishlarni hisobga olishlari mumkin.

Bog'liqlik barcha qiymatlarining pasayishi

IXP1200 tarmoq protsessoridan foydalanish ASIC integral sxemalari asosida ishlab chiqish an'anaviy usuliga qaraganda tarmoq qurilmalariga bog'liqlikni sezilarli kamaytiradi, ishlab chiquvchini yangi tarmoq xizmatlari va yangi avlod komponentlari paydo bo'lishi natijasida funksiyalari kengaygan tizimni to'liq qayta loyihalashdan ozod etadi. Buning o'rniga IXP1200 tarmoq protsessorini qayta dasturlash yetarli bo'ladi, oldingi platforma bilan bog'liqlik yo'qotilmaydi. Bunda funksiyaning kengayishi apparat modernizatsiyasi kabi dasturiy ta'minotni yangilash bilan ham amalga oshiriladi.

Yechimlarning keng diapazonli masshtabi

IX shinasidagi IXB3208 bog'lovchi qurilma bir necha IXP1200 protsessorlarni ulash imkonini beradi, buning natijasida tizimning

samaradorligi 10 Gbit/s ga oshishi mumkin. Samaradorlik parametrlari tarmoq yechimlarining keng diapazonli masshtabi mikrokontrollerlar uchun dasturiy ta'minot yaratishga sarflangan vositalarni to'liq oqlanishini kafolatlaydi. Bunday masalalarni yechishda tarmoq xavfsizligini ta'minlash uchun ishlab chiquvchilar o'tkazish qobiliyatiga bog'liq holdagi u yoki bu yechim samaradorligini nazorat qilib, xuddi shu protsessordan foydalanishlari mumkin.

Turli xususiyatli tizimlarni yaratish

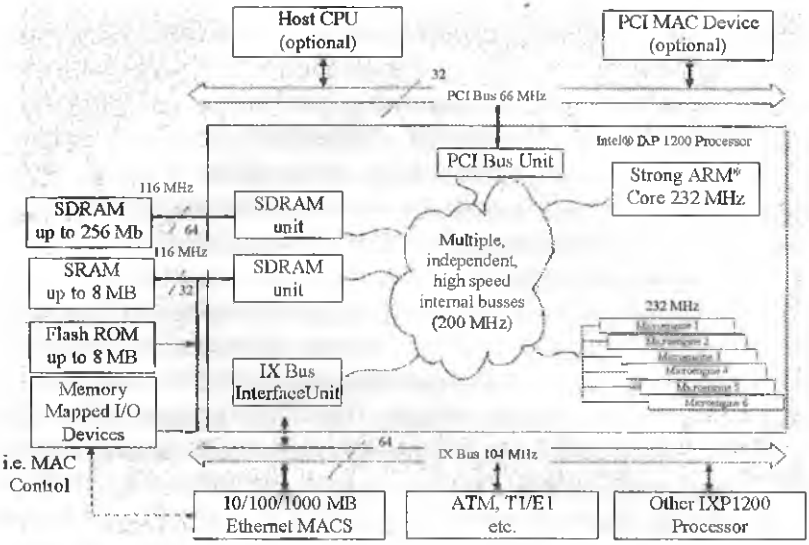
Mikrokontrollerlar va protsessorlar dasturlanishi mumkin, shuning uchun ishlab chiqaruvchilar maxsus algoritm va dasturlarni yaratish uchun ko'pgina imkoniyatlarga ega bo'ladilar, natijada tayyor qurilma barcha nazarda tutilgan funksiyalarga ega bo'ladi. IX arxitekturasi samaradorlikni tahlil qilish, dasturiy ta'minot yaratish, apparatni yig'ish uchun barcha umumiy instrumental vositalar to'plamiga ega. Yaratish muhiti IX arxitekturasi bazasidagi ixtiyoriy tizim komponentini kiritishni soddalashtiradi, hatto bir-biriga bog'liq bo'lmagan holda.

3.3.2. IXP tarmoq protsessorlari

IXP1200 protsessori. IXP1200 protsessori StrongARM protsessor yadrosini, 6 ta erkin 32 bitli RISC mikroprotsessorini (Microengine), SRAM, SDRAM qurilmalarini, PC1 va IX kontroller shinalarini o'z ichiga oladi (3.3-chizma).

Operatsion chastotasi 166–232 MGts. Protsessor samaradorligi 3 mln pak/s ga teng, 1,5 Gbit/s ni beradi. Samaradorlikni bir necha protsessorlarni parallel ishlatish bilan oshirish mumkin. Bunda 8 ta protsessorni bog'lab samaradorlikni 24 Mpps ga yetkazish mumkin.

Funksional bloklar. IXP1200 bir-biri bilan turli usullarda bog'langan bir necha funksional bloklardan iborat. Har bir modul mustaqil ishlashi, zaruriy hollarda boshqa modullarga so'roq yuborishi mumkin. Bu boshqa modullar o'z ishini tugatishini kutishga chek qo'yadi. Quyida StrongARM yadrosi, 6 ta mikroprotsessor, IX shinalari, SDRAM, SRAM, PCI modullari tavsiflangan.



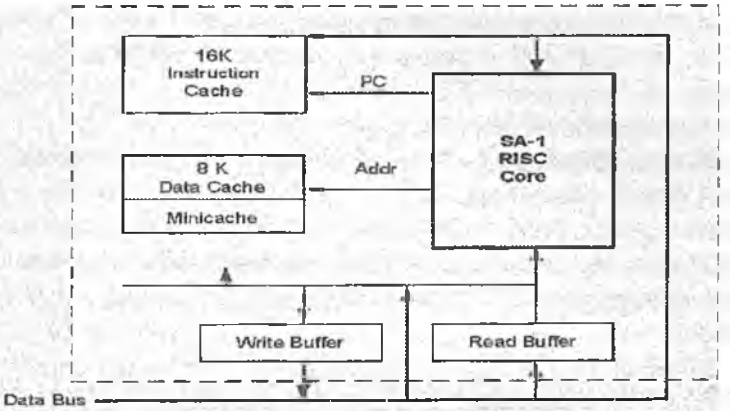
3.3-chizma. IXP1200 protsessori blok-sxemasi

StrongARM yadrosi – 32 bitli standart sanoat RISC protsessori, StrongARM protsessorlar oilasiga mos keladi, hozirgi vaqtda tarmoq va cho'ntak kompyuterlari, mobil telefonlarda ishlatiladi.

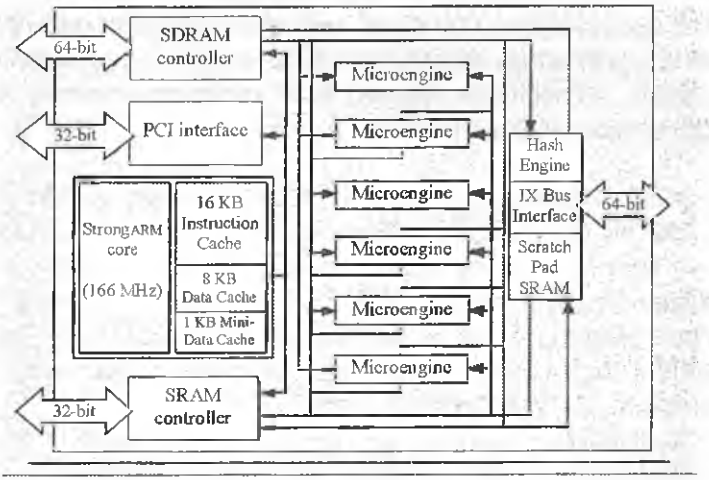
StrongARM yadrosi ikkita keshga ega: komandalar uchun 16 kilobaytli va ma'lumotlar uchun 8 kilobaytli (3.4-chizma). Bundan tashqari yadro bir marta hisoblanib, foydalaniladigan va o'chirib tashlanadigan ma'lumotlar uchun 512 baytli keshga ham ega.

StrongARM yadrosi ilovaga bog'liq holda turli usullarda qo'llanilishi mumkin. Agar tizim yetakchi protsessorga ega bo'lsa, unda IXP1200 bilan PCI shinasi orqali o'zaro aloqada bo'ladi, mikroyadro paketga ishlov berayotgan vaqtda StrongARM esa marshrutizatsiya protokollarini bajarishi mumkin. Yetakchi protsessor mavjud bo'lmagan sxemalarda StrongARM asosiy protsessor hisoblanadi.

Modomiki, mikroprotsessorlardan ma'lumot uzatishda foydalanilar ekan, StrongARM yadrosi operatsion tizimni real vaqt masshtabida (RTOS) ishga tushiradi va murakkab masalalar (adresni aniqlash, tarmoqni boshqarish, marshrutizatsiya jadvalini shakllantirish va nazorat qilish)ni bajaradi. StrongARM yadrosi nominal chastotada (166 – 232 MGts) ishlaydi.



3.4-chizma. StrongARM mikroprotsessori blok-sxemasi



3.5-chizma. IXP1200 mikroprotsessori blok-sxemasi

Microengines (qo‘shimcha mikroprotsessor yadrolari). Microengines – bu StrongARM ning yordamisiz ma’lumotlar uzatuvchi va texnik operatsiyalarni bajaruvchi va ma’lumot uzatish va bit, bayt, so‘z va uzun so‘zlar bilan ishlash uchun mo‘ljallangan 6 ta ko‘p potokli 32 razryadli RISC mikroprotsessorlari. Microengines mikrokod bilan simvol ko‘rinishida dasturlanadi (3.5-rasm).

Har bir Microengine 4 ta mustaqil komanda hisoblagichiga ega, bu har bir IXP1200 protsessorida mikrokodning to'la 24 holatini beradi.

Komandalar va ularning bajarilishi

Har bir Microengine komandalar uchun xususiy xotiraga (Program Control Store) ega, 128 ta umummaqsadli registr va 128 ta uzatish registri. Barcha Microengine bir xil, shuning uchun ular orasida funksiyalarni qayta taqsimlash mumkin. Bundan tashqari, ixtiyoriy Microengines uchun hech qanday oldindan biriktirilgan funksiya yo'q.

Microengines barcha komandalari bir takt siklida bajariladigan Strong-ARM singari IXP1200 ning chastota sinxronizatsiyasi asosida ishlaydi. Microengines axborotni 5 bosqichli konveyerli ishlov beruvchi mikroprotsessorlar kabi yaratilgan. 1-bosqichda instruksiya xotira qurilmasidan o'qiladi; 2-bosqichda komanda dekodlanadi va manba registr adresi shakllantiriladi; 3-bosqichda registrdan operand tanlanadi; 4-bosqichda operand ALU (arifmetik-mantiqiy qurilma) orqali o'tadi; 5-bosqichda natija tayinlangan registrga yoziladi.

Multithreading (axborotga ko'p potokli ishlov berish)

Har bir Microengines 4 ta mustaqil dastur hisoblagichlariga ega va 4 tagacha oqimga ishlov beradi. Bir oqim tashqi ma'lumot tushishini kutayotgan vaqtda unda boshqa jarayon ishga tushirilgan bo'lishi mumkin. Bunday holatda bitta Microengine 4 ta oqimga bir vaqtda xizmat ko'rsata oladi. Mos ravishda 6 ta Microengine 24 ta oqimga xizmat ko'rsata oladi.

Registrlar

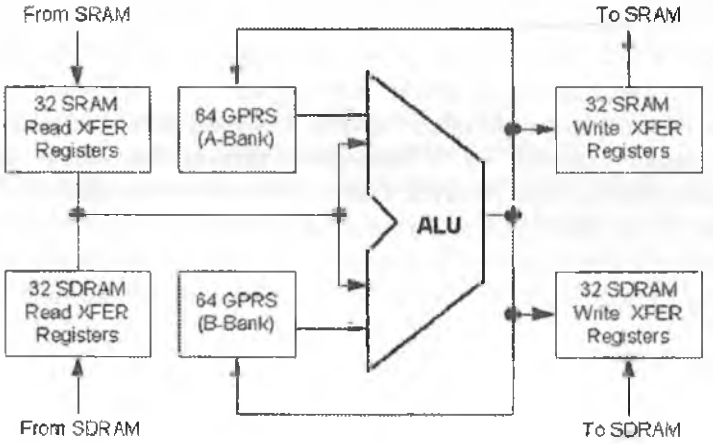
Microengine da 2 xil registrlar bor: universal (umummaqsadli) va uzatish. 128 ta universal registrlar o'z navbatida arifmetik-mantiqiy qurilmaga bir vaqtda ikkita operandni tanlash imkonini berish maqsadida ikkita bankka bo'linadi (A va V). Arifmetik-mantiqiy qurilma komandalari A va V bankning bittadan operandidan foydalanadi (3.6.-chizma).

Uzatish registrlari SDRAM va SRAM turlariga bo'linadi va o'qish va yozish qismida 32 tadan registr, jami 128 ta registrdan iborat. Har bir registrlar to'plamidan bir vaqtda foydalanish mumkin,

chunki ular funksional bloklarga mos ravishda alohida axborot kanallariga ega.

Registrlarga adreslashning 2 ta usuli bilan murojaat qilishi mumkin: context-relative va absolute. Nisbiy kontekst rejimida har bir oqimga xususiy registr biriktiriladi. Absolyut adreslashda oqimlarga taqsimlash rejimida xizmat ko'rsatiladi, bunda ular bitta fizik registrga murojaat qiladilar.

Registrlar simvolli shaklda nomlanadi. Dasturlash qurilmasi foydalanishdagi registrga murojaat qiladi, assembler mikrokodi esa mos keluvchi nomni beradi.



3.6-chizma. Registrlardan foydalanish

Registr toifasini (umummaqsadli registr, SDRAM, SRAM) va adreslash usulini ko'rsatish uchun bir necha simvol prefikslaridan foydalaniladi. Microengines axborot uzatish uchun asosiy ishni bajaradi, shuning uchun ular protsessorning boshqa modullariga kirish huquqiga ega.

IX bus moduli va IX shinasi

IX bus moduli IX shinalari bilan boshqariladi (3.7-chizma), ma'lumotlarni navbatga kelib tushish tartibida (FIFO) uzatadi. IX shinalari protsessorni 10/100 Mbit yoki Gbit Ethernet kontrolleri

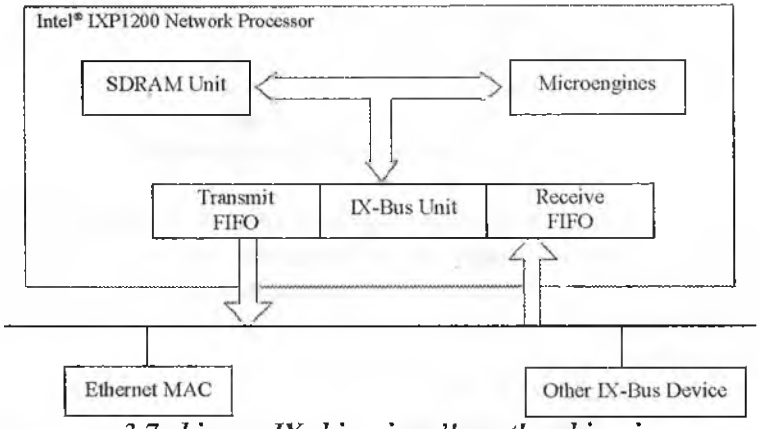
turidagi MAC (Media Access Control – muhitga kirishni boshqarish) qurilmasi va IXP1200 parallel protsessorlari bilan bog'laydi.

Ishchi chastota (66 – 104 MGts). Maksimal o'tkazish qobiliyati mikroprotsessorning ishchi chastotasiga bog'liq holda 4 – 6,26 Gbit/s ni tashkil etadi.

IX shinalari ikki tomonlama yo'naltirilgan 64 razryadli shina yoki ma'lumotni qarama-qarshi yo'nalishlarda uzatadigan 2 ta bir tomonlama yo'naltirilgan 32 razryadli mustaqil shinalar kabi konfiguratsiya qilingan bo'lishi mumkin.

FIFO navbatda turgan ma'lumotlar ixtiyoriy Microengine (uzatish registri) yoki SDRAM ga uzatilishi mumkin.

IX bus moduli nazorat va holat registrlaridan, 4 Kbayt yuqori operativ RAM (kesh) va 48 va 64-bitli hash-kalitlarni generatsiya qiluvchi hash modulidan iborat. Shuningdek, shinalari bilan parallel ishlovchi yon polosa shinalari mavjud. U Ready-shinasi deb ataladi, 8 ma'lumot razryadi va 5 boshqaruv razryadidan iborat va IX shinalari bilan sinxron ishlaydi. Qabul qilinuvchi va uzatiluvchi FIFO navbat 16-64 baytli jadval ko'rinishida yaratiladi. FIFO mikroprotsessorning barcha oqimlari uchun ochiq, dasturiy ta'minot esa barcha oqimlar tomonidan to'g'ri foydalanilishini ta'minlashi kerak.



3.7-chizma. IX shinasi ma'lumotlar shinasi

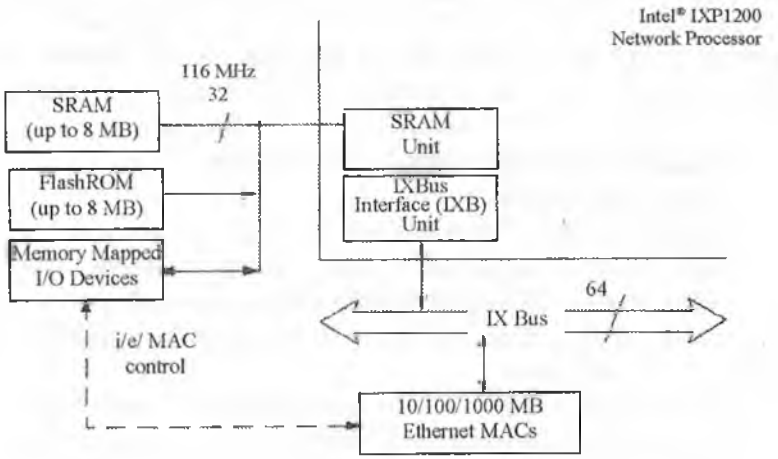
SDRAM moduli. IXP1200 ma'lumot saqlash, axborot uzatish va uni navbatga qo'yish uchun foydalaniladigan yuqori o'tkazish

qobiliyatiga ega xotiraga kirish uchun SDRAM moduliga ega. StrongARM bo'shlig'i 256 Mbayt SDRAM manzillash imkonini beradi.

SDRAM-shinasi 64 razryadga ega. Qachonki StrongARM yoki PCI modulidan bayt, so'z yoki uzun so'z operatsiyasi kelsa, to'rtlangan so'z (64 bit) SDRAM dan o'qiladi. Faqatgina kerakli baytlar o'zgartiriladi, to'liq to'rtlangan so'z esa SDRAM ga qayta yoziladi (bu 3 ta qadam: o'qish-korrektirovka qilish-yozish avtomatik tarzda amalga oshiriladi). Bitta mikrokomanda bir vaqtda 16 ta to'rtlangan so'z (128 bayt)ni uzatishni amalga oshirishi mumkin. Birgina Microengines dan qayta ishlangan to'rtlangan so'z tushadi. 8 baytdan kichik ma'lumotlar komanda doirasidagi bayt shablonidan foydalanib yozilishi mumkin, lekin bu o'qish-modifikatsiya-yozish sikliga kiradi.

SDRAM interfeysi 232 MGts da 928 Mb/s o'tkazish qobiliyatini ta'minlovchi asosiy chastotaning yarmida ishlaydi.

SRAM moduliga izlash jadvallarini, mikroprotsessor paketlarga ishlov berish va kerakli boshqa ma'lumotlarni saqlash uchun juda katta xotira qurilmasi kerak (9-rasm). SRAM moduli SRAM ni (8 Mb gacha), yuklash uchun BootROM (8 Mb gacha) va periferik qurilmalarga kirish uchun SlowPort 2 Mb li adreslar bo'shlig'ini boshqaradi.



3.8-chizma. SRAM moduli, tashqi interfeyslar

SRAM interfeysi 32 razryadli – bu SDRAM razryadlarining yarmi, SRAM katta hajmdagi ma'lumotlarni saqlash uchun emas, balki tezkor qidiruvga mo'ljallangan. SDRAM interfeysi kabi SRAM interfeysi ishchi chastotasi asosiy chastotaning yarmidir.

PCI Unit. PCI moduli periferik qurilmalar uchun PCI interfeysini tashkil qilib, standart 32 razryadli (Peripheral Component Interconnect, PCI) shina bilan ishlashni ta'minlaydi. PCI moduli 66 MGts gacha tezkorlikni va «PCI Local Bus Specification Revision 2.2» ni qo'llaydi. 33 MGts dan ko'proq tezkorlikda ishlashni faqat 2ta PCI qurilmasi ta'minlaydi, ulardan biri IXP1200. PCI-to-PCI ko'prigi yuqori chastotalarda katta sonli qurilmalarni qo'llash uchun ishlatilishi mumkin.

PCI moduli SDRAM moduli bilan bog'langan, shuning uchun PCI shinasidagi qurilmalar SDRAM ga to'liq murojaat qila oladilar. Ikkita DMA kontrolleri PCI moduliga kiritilgan, shuning uchun ikkalasi StrongARM yoki Microengine dan foydalanish mumkin. DMA kontrollerlari doimo SDRAM da turuvchi DMA tavsiflovchisi yordamida dasturlanadi. Tavsiflovchilar shunday bog'langan bo'lishi mumkinki, bunda SDRAMning bir necha qo'zg'almas ma'lumotlar bo'limidan iborat bloklar PCI ga bitta blok ko'rinishida uzatila olinishi kerak.

PCI va IX bus modullari orasida ma'lumotlar uzatilganda mikroprotsessorlardan foydalanilishi kerak. Ma'lumotlar IXbus moduli FIFO sidan SDRAM ga bevosita uzatila olinishi kerak, Microengine bo'lsa ma'lumotlarni SDRAM dan PCI interfeysiga uzatish uchun PCI DMA-kontrolleridan foydalanadi.

IXP1200 da paketga ishlov berish algoritmi

Paketni qabul qilish

Paket MAC dan IX bus moduliga tushadi (3.9-chizma).

Sarlavha Microengine qabul qilish oqimiga uzatiladi.

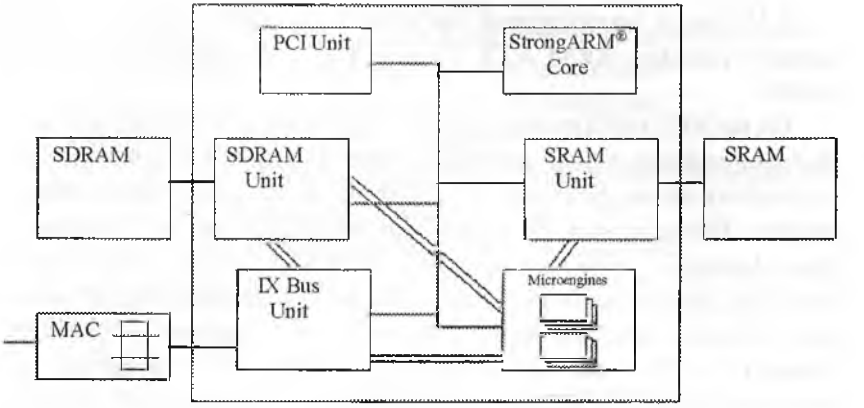
Paket tanasi SDRAM ga saqlash uchun yuboriladi.

Qabul qilish paketi SRAM ni ko'rib chiqadi va SDRAM dan adres axborotlarini oladi.

Microengine sarlavhani adres axborotiga mos o'zgartiradi.

O'zgartirilgan sarlavha SDRAM dagi paket tanasiga bog'lanadi.

Qabul qilish potoki paket diskriptorini SRAM ga uzatish uchun navbatga qo'yadi.



3.9-chizma. IXP1200 protsessorida paketga ishlov berish

Paketni uzatish. Uzatish jarayoni protokol diskriptorini hisoblaydi.

Paket diskriptoridagi axborotdan foydalanib uzatish jarayoni SDRAM dan paketni topadi va SDRAM moduliga paketni IX bus uzatish moduli FIFO siga o'tkazish komandasini beradi.

SDRAM moduli paketni IX bus moduliga uzatadi.

Uzatish jarayoni bo'sh paket diskriptorini qaytadan uzatish navbatiga qo'yadi.

IX bus moduli paketni MAC ga va keyinchalik IXP1200 protsessoriga uzatadi.

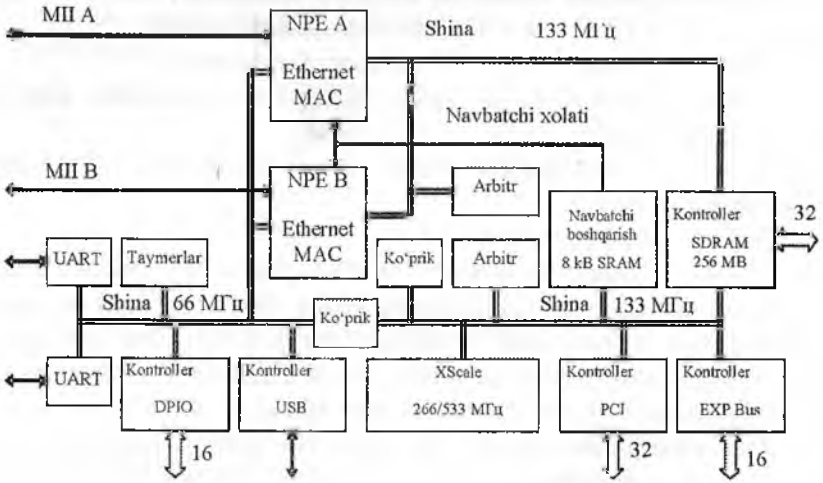
IXP420 tarmoq protsessori.

IXP420 tarmoq protsessori – uncha qimmat bo'lmagan, biroq unumdorligi yuqori uy shlyuzlarida, kichik ofislardagi marshrutizatorlarda, simsiz foydalanish qurilmalarida qo'llashga mo'ljallangan bir krisatalli protsessorlar qatorining kichik a'zosidir. Protsessorning, 3.10-chizmada ko'rsatilgan noyob taqsimlangan ishlov berish arxitekturasi (distributed processing architecture)uning asosida qurilmalar ishlab chiqilishini tezlashtiradi. Asosiy XScale ishchi yadroning ikkita mustaqil taqsimlangan ishlov berish apparat vositalari (network processor engines - NPE) bilan to'ldirilishi, paketlarga ular kelib tushadigan tezlik bilan ishlov berish uchun yetarli bo'lgan jami unumdorlikka erishish imkonini beradi.

0,18 mkm texnologiyasi bo'yicha bajarilgan XScale yadrosi dasturiy jihatdan ARM V5T Thumb RISC protsessori bilan mos keladi.

Ikkita NPE ma'lumotlar ustida katta hisoblash xarajatlarini talab etadigan operatsiyalarni, jumladan, nazorat summalari generatsiyasi va tekshirilishini, bayroqlar kiritilishini va chiqarib tashlanishini, paketlar filtratsiyasini, IP sarlavhalar tekshirilishini va o'zgartirilishini bajaradi. NPE arxitekturasi AMQ ni, ichki ma'lumotlar xotirasini, interfeyslarning keng to'plamini va tarmoq ilovalar uchun tipik bo'lgan hisoblashlarni tezlashtirishning apparat vositalarini ichiga oladi. Bu apparat vositalar mikrodasturiy boshqariladi va protsessor bilan birga yetkazib beriladigan funksiyalar bibliotekasidan foydalanishi mumkin.

Protsessorlardan tashqari, asbob SDRAM xotira kontrollerini, tashqi shinalar kontrollerini va taymerlar, hamda periferik qurilmalarning keng to'plamini ichiga oladi, bu uni tugallangan «kristaldagi qurilma» qiladi.



3.10-chizma. IXP420 protsessor arxitekturasi

Yadroning dasturiy ta'minotining asosi sifatida Wind River kompaniyasining Vx Works kabi keng tarqalgan operatsion tizimlaridan yoki standart Linux dan foydalanishi mumkin.

IXP420 ning farqlovchi xususiyatlari quyidagilar:

- dasturlashtiriladigan XS caleyadrosi;
- MII interfeysli Ethernet ikkita integrallashgan MAS-kontrolleri;
- ikkita yuqori tezlikli (920 KBod gacha) UART;
- 8 dan 256 Kbayt gacha SDRAM xotira kontrolleri;
- to'rttagacha qurilmani ulash imkoniyati bo'lgan PCI v.2.2 host-interfeysi;
- USB 1.1 interfeysining kontrolleri;
- 16 razryadli kengayish shinasi;
- 16 ta General Purpose I/O (GPIO) universal chiqish uyi;
- 533 MGts gacha bo'lgan chastota (temperaturalarning kengaytirilgan diapazonida 266 MGts gacha);
- namunaviy iste'mol qilinadigan quvvat 1 - 1,5 Vt.

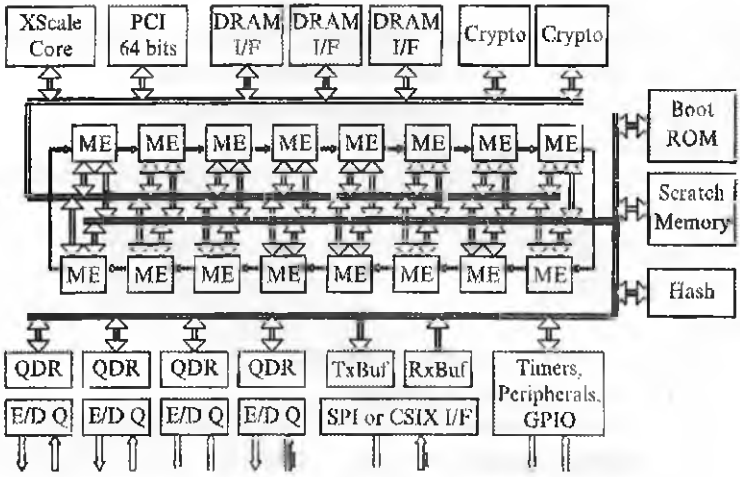
IXP2855 tarmoq protsessori

Intel kompaniyasi tarmoq protsessorlarining eng kuchlisi. 16 ta ko'p ipli (tolali) (multi-threaded) mikroprotsessorlar (micro-engines - ME), umumiy boshqarish uchun unumdorligi yuqori XScale yadroli va shifrlashning ikkita apparat tezlashtirgichi bo'lgan «qabul qil va jo'nat» (store-and-forward) arxitekturasini amalga oshiradi. Protsessorga standart apparaturani ishlab chiqish vositalari (advanced telecom computing architecture-compliant hardware development kit). bilan to'ldiriladigan dasturiy ta'minot ishlab chiqishning firma vositalari (exchange architecture software development kit)beriladi.

IXP2855 protsessorining 3.11-chizmada ko'rsatilgan arxitekturasida parallel ishlov berishning elastik arxitekturasini asosiy o'ziga xos xususiyat deb hisoblash mumkin. Unda 16 ta ME mustaqil ishlashi yoki shifrlash bloklari bilan birga umumiy konveyerga qo'shilishi mumkin.

Quyidagilar IXP2855 protsessorining farqlovchi xususiyatlari hisoblanadi:

-TCP, IPSec va SSL protokollarini qo'llash uchun optimallashtirilgan, to'liq dasturlashtirilgan elastik arxitektura;



3.11-chizma. IXP2855 protsessori arxitekturasi

-Polising, dispetcherlash, navbatlarni boshqarish va protokollararo o'zaro birgalikda ishlashni qo'llagan holda, paketlarni 10 Gbit/s gacha bo'lgan tezliklarda uzatish qobiliyati;

-sekundiga 25 mlrd. operatsiyadan iborat jami unumdorlikni ta'minlaydigan, 16 ta to'liq dasturlashtiriladigan ME;

-murakkab algoritmlarni puxtalash (me'yorga yetkazish), marshrut jadvallari bilan ishlash, umumiy boshqarish va menejment funksiyalari uchun mo'ljallangan, unumdorligi yuqori XScale yadrosi;

-navbatlar bilan sekundiga 64 mln. operatsiya o'tkazish unumdorligiga ega bo'lgan, navbatlarni boshqarishning apparat vositalari;

-tizim integratsiyasini soddalashtiradigan standart interfeyslar;

-DES, 3DES, AES, ShA-1 algoritmlarini amalga oshiradigan va 10 Gbit/s gacha bo'lgan tezliklarda IPSec shifrlashni bajaradigan, integrallashgan, shifrlashning apparat vositalari;

-dasturiy bloklar bibliotekasi va tayyor mahsulotni olish vaqtini tezlashtirish apparaturasini ishlab chiqishning keng tarqalgan vositalari.

IXR tarmoq protsessorlarini solishtirish.

Tarmoq protsessorlari turli xil tarmoq jihozlarida keng qo'llab kelinadi. Bu, birinchi navbatda, abonentda joylashtiriluvchi jihoz, tarmoqlarga ulanishni ta'minlovchi jihoz, tarmoqlar chegarasida o'rnatiluvchi jihoz va magistral tarmoqlar jihozidir.

Intel firmasi tarmoqning barcha segmenti uchun tarmoq protsessorlari ishlab chiqaradi. IXR4xx oilasi tarmoq protsessorlari abonentda joylashadigan jihozlarda qo'llaniladi (SRE). Avvalda ushbu jihozda ishlatilgan IXR22x tarmoq protsessorlari hozirda ishlab chiqarilmaydi. IXR12xx tarmoq protsessorlari tarmoqqa ulanish jihozida (Access) va tarmoqlar chegarasidagi jihozda (Edge) qo'llaniladi. Bular birinchi avlod tarmoq protsessorlaridir. IXR2400 tarmoq protsessorlarining ikkinchi avlodi bo'lib, ulanish va tarmoq jihozida ulash funksiyalarini bajaradi. IXR28xx tarmoq protsessorlari magistral tarmoqlar jihozi tarkibiga kiradi (Core).

IXR420, IXR421, IXR422, va IXR425 tarmoq protsessorlari ma'lumotlar va nutqni xavfsiz uzatish uchun mo'ljallangan. Ular XScale yadrosi asosida yaratilgan va o'zida yuksak funksionallik va standart arxitekturani jamlaydi.

O'rnatilgan shifrlash tizimiga ega IXR422 protsessori shlyuzlar, simsiz tarmoqlarga ulanish nuqtalarida, marshrutizatorlar va kommutatorlarda ishlatish uchun mo'ljallangan. IXR421 modeli ma'lumotlar va nutq uzatish qurilmasiga (VoIP) mo'ljallangan, IXR420 esa, uy tarmoqlari va kichik kompaniyalar lokal tarmoqlarida qo'llaniluvchi shlyuzlar va marshrutizatorlar kabi keng polosali ulanish qurilmalari uchun optimalashtirilgan. Windows operatsion tizimi CE.NET maxsus talqinini qo'llab-quvvatlash amalga oshirilganligi uchun IXR425 tarmoq protsessori IXR4xx oilasidagi turli maqsadlardagi qurilmalarda ishlatiluvchi birinchi tarmoq protsessori bo'ldi.

U quyidagilarda qo'llaniladi:

- turar joylardagi qudratli shlyuzlarda;
- axborotni muhofaza qilish qurilmalarida, kichik va o'rta korxonalar tarmoqlari kommutatorlari va marshrutizatorlarida;
- tarmoq printerlarida;
- multipleksorlar va xDSL adapterlarida;
- radio ulanish qurilmalarida;
- ishlab chiqarishni boshqarish tizimlarida.

IXP1200, IXP1240, IXP1250 tarmoq protsessorlari axborot paketlarini 1,6 Gbit/s gacha bo'lgan tezlikda qayta ishlash uchun mo'ljallangan. Ular LAN- va WAN-tarmoqlarga ulanish jihozlarida, yuqori tezlikdagi modemlarda, masofadan ulanish qurilmalarida, PCI-adapterlarda ishlatiladi.

Tarmoq protsessorlarining dasturlanishi tarmoq qurilmalariga protsessorning o'zini o'zgartirmay, yangi funksional imkoniyatlarga ega bo'lish imkonini beradi. Tarmoq protsessorlari keng diapazondagi tezliklarda ishlaydi va ochiq tizimlar o'zaro ta'siri etalon modeli (OTO'TEM)ning 2-dan 7-darajasigacha bo'lgan talablarni bajaradi.

Ikkinchi avlod tarmoq protsessorlari arxitekturasi o'z ichiga uchta asosiy elementni oladi:

- axborot paketlariga ishlov berishning yuqori tezligini ta'minlovchi butunlay dasturlanadigan elementar protsessorlar quyi tizimi;
- tarmoq protsessorining unumdorligi va energiya iste'moli o'rtasidagi eng yaxshi nisbatni ta'minlovchi XScale yadrosi;
- mavjud tarmoq protsessorlarini va ularning IXA (Internet Exchange Architecture) texnologiyasi bo'yicha bajarilgan keyingi avlodlarini qo'llovchi jihozlarda dasturlar bloklarini ko'p bora ishlatishni ta'minlovchi dasturiy ta'minotni ko'chirish uchun integrallashgan muhit.

IXP2400 axborotlar paketiga 4,8 Gbit/s gacha tezlikda ishlov berish uchun mo'ljallangan. U ko'p servisli kommutatorlar, marshrutizatorlar, simsiz aloqa tizimlariga ulanishning keng polosali qurilmalari kabi tarmoqqa ulanish va tarmoqlarni ulash jihozlarida qo'llaniladi.

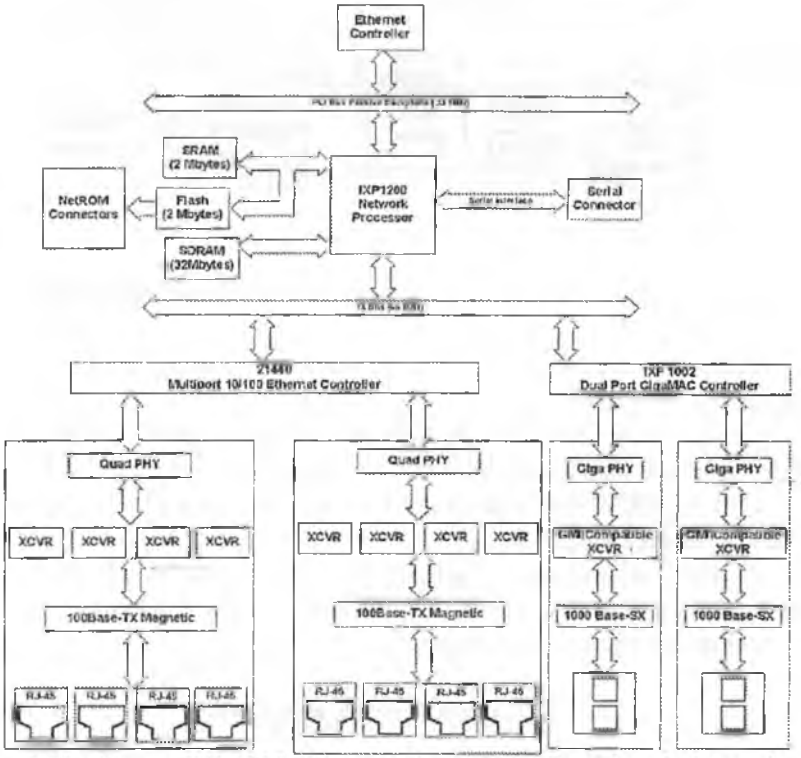
IXP2400 XScale yadrosiga va sakkizta dasturlanuvchi elementar protsessorga ega. Ular bir vaqtning o'zida sakkizta axborot oqimiga 600 MGts chastotada ishlov beradi. Sakkizta elementar protsessor va XScale yadrosi tufayli IXP2400 tarmoq protsessori ishlab chiquvchilarga unumdorlikni oshirish va IXP1200 tarmoq protsessorini ishlatuvchi jihozga yangi imkoniyatlar qo'shish imkonini beradi.

IXP2800 tarmoq protsessori ikkinchi avlod tarmoq protsessorlari ichida eng yuqori darajadagi unumdorlikni ta'minlaydi. Bu tarmoq protsessori axborot paketlariga 10 Gbit/s gacha bo'lgan tezlikda ishlov berish uchun mo'ljallangan. U tarmoqlarni ulash qurilmalarida

va kommutatorlar, marshrutizatorlar, ko'p servisli kommutatorlar kabi magistral tarmoqlar qurilmalarida ishlatiladi.

IXP2800 o'n oltita butunlay dasturlanuvchi elementar protsessorga va XScale yadrosiga ega. Ular bir vaqtning o'zida 16 ta axborot oqimiga 1,4 GGts yoki 1,0 GGts chastotada ishlov beradi.

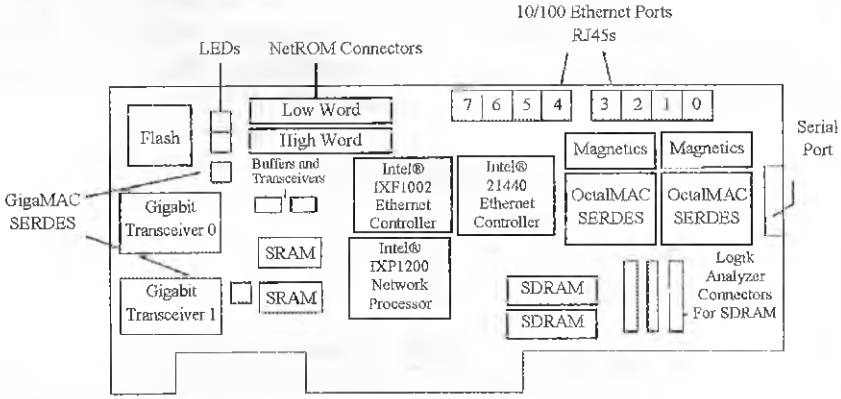
IXP2850 tarmoq protsessori IXP2800dagi kabi tuzilish va imkoniyatlarga ega, biroq axborotni shifrlash va deshifrlash imkoniyatini beradi. Buning uchun uning tarkibiga ikkita kriptohimoya bloki kiritilgan, ular 3DES/DES (Data Encryption Standard), AES (Advanced Encryption Standard) va ShA-1 (Secure Hash Algorithm) algoritmlarini qo'llaydi.



3.12-chizma. IXP1200 Ethernet Evaluation Board arxitekturasi

IXP1200 Ethernet Evaluation System

IXP1200 Ethernet Evaluation Board – bu dasturiy ta'minotni yaratish va tekshirishga yordam beruvchi kuchli vosita. Foydalanuvchi host kompyuteri bilan ma'lumotlar potoklari yo'lini aniqlash, chip va tizimning ishlashini, drayver va boshqalarning ishini tekshirish uchun chuqur tekshiruv o'tkazishga imkon beradi. Tizim asosiy plata arxitekturasi 3.12-chizmada, modul kompanovkasi esa 3.13-chizmada ko'rsatilgan.



3.13-chizma. IXP1200 Ethernet Evaluation Board elementlari

IXP1200 Ethernet Evaluation Board quyidagi maqsadlarda ishlatilishi mumkin:

- Namunaviy dasturiy ta'minotni ishga tushirish bilan bitta IXP1200 ning samaradorligini namoyish qilish;
- StrongARM va Microengines yadrolari dasturiy ta'minotini yaratish va testlash;
- Elektr va mexanik xarakteristikalarni, shuningdek IXP1200 asosidagi standart qurilma komponentlari tarkibini ko'rsatish;
- Dasturiy ta'minotni yaratish.

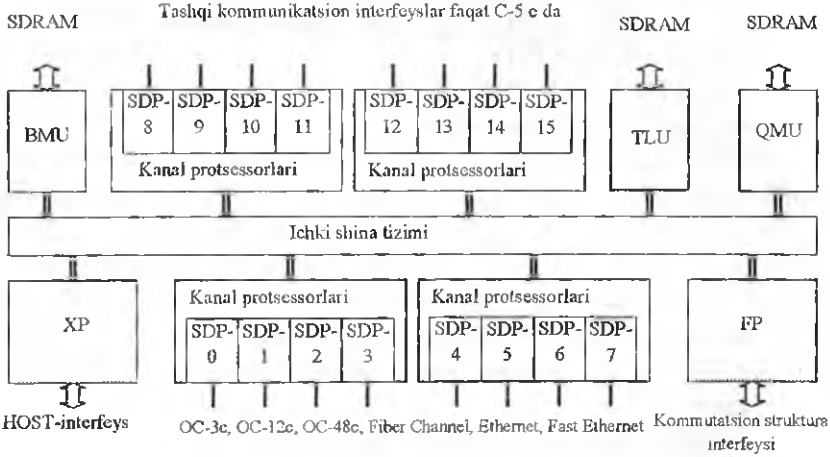
3.4. C-port tarmoq protsessori

Kommunikatsion protsessorlar Motorola kompaniyasi tomonidan dastlab o'tgan asrning 80-yillarida chiqarilgan va hozirgacha uning vorisi – Freescale kompaniyasi tomonidan chiqarilishi va

takomillashtirilishi davom ettirilmoqda. QUICC va PowerQUICC turkumdagi kommunikatsion mikrokontroller alohida maqolalar bag'ishlangan. Bu turkum hozirgi kunda ham, uncha qimmat bo'lmagan, universal dasturlashtiriladigan telekommunikatsion mikrokontrollerlar bozorida eng ommabop hisoblanadi.

Quyida Motorola kompaniyasining birmuncha yangi va kuchli tarmoq protsessorlari turkumi qarab chiqiladi. S-port tarmoq protsessorlari (network processors) turkumining arxitekturasi juda spetsifik va ko'plab yuqori tezlikni ma'lumotlar oqimlariga ishlov berishga yo'naltirilgan (3.14.-chizma).

Bu turkum o'z ichiga tashqi portlar soni va 8 port bo'lganda umumiy o'tkazish qobiliyati tegishlicha 3 Gbit/s va 16 port bo'lganda 5 Gbit/s bo'lishligi bilan farqlanadigan ikkita kichik turkum – S-3e va S-5e ni oladi.



3.14-chizma. C-port turkumidagi tarmoq protsessorlarining umumlashtirilgan strukturasi

Har ikki kichik turkum 4500 MIPS dan oshadigan ja'mi hisoblash quvvatiga ega, bu ko'plab tarmoq qo'llanishlar uchun yetarli va intellekt tarmoq servislarining qo'llab-quvvatlanishini, aynan paketlar klassifikatsiyasini, trafik boshqarilishini va QoS funktsiyalarini ta'minlaydi. S-port protsessorlari bir nechta funksional yo'naltirilgan bloklarni ichiga oladi, ularning har biri mustaqil ravishda amaliy

dasturlash interfeyslari orqali (applications programming interface - API) Si tilida dasturlashtiriladi. Alohida bloklarining imkoniyatlari u yoki bu tashqi dasturlashtiriladigan soprotsessorlarni ulash bilan sezilarli kengaytirilishi mumkin. Xususan, zarur bo'lganda S-port protsessoriga qo'shimcha tashqi kanal adapterlari (channel adapters) va trafikni boshqarish soprotsessori (traffic management coprocessor) ulash mumkin.

Kanal protsessorlari. Tipik holatda kirish trafikni C-port tarmoq protsessoriga soni 8 dan 16 gacha bo'lgan, oqimlari OC-3 (155 Mbit/s) dan Gigabit Ethernet (1 Gbit/s) gacha qabul qila oladigan kanal protsessorlari orqali keladi. Bunda, tashqi interfeyslarni dasturiy tanlash imkoniyati bor, ularning orasida T/E-freymerga yoki kanallar adapteriga mo'ljallangan maxsus interfeyslar, ham:

- Ethernet / Fast Ethernet (RMII);
- Gigabit Ethernet PHY (GMII yoki TVI);
- OC-3 / STM-1 PHY;
- OC-12 / STM-4 PHY;
- UTOPIA 3 / PoS PHY

kabi standart interfeyslar bor.

Har bir kanal protsessori Si tilida dasturlashtiriladigan RISC yadroni ichiga oladi va ketma-ket ma'lumotlarning to'rtta protsessoriga (serial data processors - SDP)xizmat ko'rsatadi. SDP arxitekturasi Ethernet interfeyslari uchun MAS darajasidagi apparaturani va SONET freymerlarini ko'zda tutadi. Shuningdek, har bir SDP mustaqil tarzda dasturlanadi va instruksiyalarning o'ta uzun so'zlardan iborat arxitekturasiga ega (very long instructions word - VLIW). U:

- Ethernet;
- Packet over SONET (GMII yoki TBI);
- Frame Relay;
- ATM;
- HDLC;
- Fiber Channel.

kabi namunaviy protokollarni qo'llanishga dasturlashtirilishi mumkin.

Dasturlashning mumkinligi SDP ga yangidan kiritiladigan protokol va formatlarni, masalan, belgilar bo'yicha ko'p protokolli

kommutatsiyani (multiple protocol label switching - MPLS), shuningdek, MAS darajadagi ixtiyoriy foydalanuvchi interfeyslarini qo'llash imkonini beradi.

Jadvalda izlash bloki. Jadvalda izlash bloki (table lookup unit - TLU) trafikni tasniflash bo'yicha vazifalarning keng doirasini hal etish va OS-48s / STM-16 klassidagi ilovalarni qo'llash uchun yetarli bo'lgan izlash algoritmlarini amalga oshirish uchun xizmat qiladigan egiluvchan yuqori tezlikli qurilmani o'zida ifodalaydi. Masalan, TLU 4-versiya IP-adreslarning sekundiga 46 mln. dan ortiq solishtirishlarni bajara oladi. Tipik solishtirish operatsiyalari quyidagicha tatbiq qilinadi:

- IP-tarmoqlardagi adreslarning eng uzun prefikslariga (4- va 6-versiyalar);
- ATM tarmoqlardagi VCI / VP Iidentifikatorlarga;
- Ethernet tarmoqlaridagi MAC-adreslarga, jumladan, virtual lokal tarmoqlar;
- MPLS tarmoqlaridagi belgilarga.

Bundan tashqari, TLU real vaqtda hodisalarning statistikasini avtomatik tarzda olib boradi. Bu funksiyalarni bajarish uchun, TLU ga ishchi chastotalari 133 MGts gacha bo'lgan SDRAM turidagi 64 razryadli xotira ulanadi. Agar zarurat yuzaga kelsa, xotira o'rniga ayni o'sha interfeys orqali qurilmaga, statistika to'plash va paketalarni tasniflashning kengaytirilgan dasturlanadigan imkoniyatlar bilan shu funksiyalarni bajaradigan tashqi protsessor ulanishi mumkin.

Navbatlarni boshqarish bloki. Navbatlarni boshqarish bloki (queue management unit - QMU)ikkita rejimda ishlashi mumkin. Avtonom rejimda u mustaqil ravishda 512 gacha navbatni ta'minlashi mumkin, bu ko'pchilik qo'llanishlar uchun yetarli. Yuqori unumdorlik talab etilsa yoki QoS ga oshirilgan talablar ko'rsatilsa, QMU ga trafikni boshqarish tashqi soprotsessori ulanishi mumkin.

Kommutatsion strukturaga ulanish protsessori. Kommutatsion strukturaga ulanish protsessori (fabric processor - FP) qurilma u yoki bu kommutatsion strukturaga ulanishini ta'minlaydi, bu tizim summar terrabitli o'tkazish qobiliyatiga erishish imkonini beradi.

FP dasturlash imkoniyatlari ichiga:

- kommutatsiyalaydigan sarlavhalar va belgilarning parametrlari sozlanishini;

- to‘xtalib qolishlarda paketlarning surilishi boshqarilishini;
- paketlarning segmentlashni va to‘plashni;
- navbatlar boshqarilishini (128 tagacha navbat) oladi.

FP protsessori 32 razryadli so‘zlar bilan dupleks rejimda 125 MGts gacha bo‘lgan chastotalarda ishlashi mumkin va tashqi portda 3,2 Gbit/s gacha o‘tkazish qobiliyatini ta’minlaydi. Qo‘llaniladigan interfeyslar qatoriga ixtiyoriy dasturlanadigan foydalanuvchi interfeyslaridan tashqari, quyidagi standart interfeyslar kiradi:

-UTOPIA level 2, 3;

-CSIX-L1 (32 razryad, 125 MGts);

-IBM PowerPRS.

Dasturlash imkoniyatlari. C-port turkumi uchun, holatlar avtomatlarini an’anaviy konfiguratsiyalash zarurligi o‘rniga Si tilida to‘la dasturlab bo‘lishlik yoki kodlarda yo maxsus tillarda dasturlashtirish xos. Har bir funksional blok, xususan, barcha kanal protsessorlari alohida dasturlashtiriladi, bu C-port bitta protsessorida 16-tagacha o‘z protokollariga ega bo‘lgan turli tashqi interfeyslarni amalga oshirish imkonini beradi.

Bloklar C-ware yagona dasturiy ta’minot tizimi API orqali dasturlashtiriladi. Ular dasturiga muayyan apparaturaning o‘ziga xosliklaridan mavhumlashish va fizik adresni sozlash, buferlar va navbatlarni boshqarish, ma’lumotlarni surish, adreslarni taqqoslash kabi funksiyalar bilan ishlash imkonini beradi. Dasturlashtirish ishlab chiqishni qo‘llab-quvvatlashning kuchli paketlari:

- ishlab chiqish tizimi (C-ware development system - CDS);
- dasturiy vositalar (C-ware software toolset - CST);
- amaliy funksiyalar bibliotekalari (C-ware application library - CAL) mavjudligi tufayli soddalashadi.

Motorola va Intel kompaniyalari yondashuvlarini taqqoslash.

Albatta, Motorola va Intel kompaniyalarining yondashuvlarini juda bo‘lmaganda yuqorida qarab chiqilgan mikroelektron asboblar doirasida obyektiv taqqoslash, ularning aynan bir xil pirovard mahsulotni ishlab chiqishda qo‘llanilishini beradi. Biroq, yetarlicha tushunarli bo‘lgan iqtisodiy mulohazalarga ko‘ra, tijorat kompaniyalari turli element asosdan foydalanib, aynan bir ishlanmani bir necha variantda bajarmaydilar. Bundan tashqari, amalda bu element asosni

tanlash u yoki bu yondashuvning mavhum afzalliklari bilan emas, balki yetarlicha aniq obyektiv cheklovlar va ishlab chiquvchilarning subyektiv tuyg'ulari bilan belgilanadi. Shuning uchun, quyida so'z bu sohada yetakchilik qilayotgan ikki kompaniyaning muayyan mikroelektron mahsulotlarini taqqoslash to'g'risida emas, balki, aynan ikki yetakchining bunday mahsulotlarni yaratishga bo'lgan arxitektura yondashuvlarini solishtirish to'g'risida boradi.

Motorola kompaniyasi Digital Equipment Corp. kompaniyasidan keyin birinchilardan biri bo'lib telekommunikatsion qurilmalar bozoriga chiqdi va telekommunikatsion mikroelektronika «pioneri» bo'ldi. O'shandan beri chorak asrdan ko'proq u Freescale sho'ba kompaniya bilan juftlikda har ikkala bozorda yetakchi bo'lib qolmoqda. Uning muhandislari izchil va sobitqadamlik bilan, ko'pchilik namunaviy va ommaviy qo'llanishlarni qamrab oladigan universal dasturlashtiriladigan telekommunikatsion va tarmoq protsessorlari to'la to'plami bosh konsepsiyasini rivojlantirmoqdalar.

Kompaniyaning barcha ishlanmalari orqali protsessorlarni, aytish kerakki, standart dasturlashtirish tillarida, birinchi navbatda Si tilida erkin dasturlab bo'lishlik va bir vaqtda barcha eski funksiyalarni va kommunikatsion protokollarni, qabul qilingan jahon standartlariga muvofiq amalga oshirish g'oyasi qizil ip bo'lib o'tadi. Bu g'oya foydalanuvchini ushbu standart, biroq anchagina ko'p mehnat talab qiladigan, uning ustiga ishlab chiquvchilarning malakasi yuqori bo'lishi talab etiladigan algoritmlarni dasturlashtirish zaruratidan xalos etadi.

Integratsiyalangan qator kommunikatsion mikrokontrollerlar ko'p yillik uzluksiz izlanishlar natijasi bo'ldi:

- uch portli integratsiyalangan ko'p protokolli protsessor (integrated multi-protocol processor - IMP);
- to'rt portli integratsiyalangan kommunikatsion kortroller (quadruple integrated communications controller - QUICC);
- ko'p portli Si tilida dasturlashtiriladigan port (C-port).

Eng ko'p foydalaniladigan QUICC o'rta turkumi turli tarmoq va telekommunikatsion ilovalar talablarining keng spektriga adekvat bo'lgan bir qancha kichik turkumga ega. Xususan, turkumni takomillashtirish jarayonida Motorola kompaniyasi erkin dasturlashtiriladigan birmuncha kuchliroq PowerPC RISC arxitekturaga o'tdi

va bu o'tish natijasida olingan PowerQUICC kichik turkumli rivojlanishning to'rt avlodini bosib o'tdi: PowerQUICC I, PowerQUICC II, PowerQUICC II-Pro va PowerQUICC III.

Intel kompaniyasiga mutlaqo boshqacha strategiya xos. Boshidan yuqori darajada integratsiyalashgan va «yuqori intellektual» mikrosxemalar, birinchi navbatda, mikroprotsessorlar bozorida yetakchi bo'lish uchun tashkil etilgan bu kompaniya maqsadiga sodiq qolmoqda va eng ilg'or texnologiyalarga yo'naltirilmoqda. Shaxsiy kompyuterlar uchun universal markaziy protsessorlar, ishchi stantsiyalar va Pentium turkumidagi protsessorlari bo'lgan turli serverlar sektorida yetakchilikni saqlagan holda, kompaniya qisman markaziy protsessorlar, avvalo, XScale texnologiyasi, arxitekturasi va usullarini ko'chirgan holda, qisman telekommunikatsiya va tarmoq texnologiyalari sohasida ishlaydigan boshqa kompaniyalarning tayyor yechimlarini va nou-xausini sotib olgan holda, telekommunikatsion protsessorlar sektorida ham unumdorlik bo'yicha yetakchi o'rinlarni egallashga intilmoqda.

3.5. Kompaniyalarning zamonaviy tarmoq protsessorlari

Hozirgi kunda tarmoq protsessorlarini turli kompaniyalar ishlab chiqmoqdalar. Quyida ayrim kompaniyalarning mahsulotlari haqida qisqacha ma'lumotlar keltirilgan.

Yuqori darajada yuklangan tizimlar uchun tarmoq protsessorlarining katta qismini EZchip kompaniyasi ishlab chiqarmoqda. Bu kompaniyaning hozirgi vaqtdagi asosiy mahsuloti 30 Gb/s ga teng bo'lgan o'tkazish qobiliyatini ta'minlovchi NP-3 tarmoq protsessorlari hisoblanadi.

NP-3 tarmoq protsessorlari paketlarni 4 darajali ierarxiyali tasniflashga ega kiruvchi va chiquvchi paketlar uchun trafik menejerlarini; marshrutlash, kommutatsiyalash va xavfsizlik siyosatlarini ta'minlash uchun foydalanilishi mumkin bo'lgan izlash bloklarini, shuningdek, mashrutizatorlarni ma'muriy boshqarish va qo'llab-quvvatlashni osonlashtiruvchi OAM protokoli (Operation, Administration and Maintenance)ning apparat amalga oshirilishini o'z ichiga oladi. Bu protsessor uchun marshrut jadvallari tashqi DRAM-xotirada saqlanadi va 1.5 Gbaytgacha yetishi mumkin.

Zamonaviy kompyuter tarmoqlarining tobora o‘sib boruvchi talablarini qondirish uchun EZchip kompaniyasi keyingi avlod tarmoq protsessori bo‘lgan NP-4 ni chiqardi, bu TP marshrutizatorning 100 Gb/s (dupleks rejimda 50 Gb/s) darajasidagi o‘tkazish qobiliyatini ta’minlaydi. Bu TP NP-3 arxitekturasiga asoslangan, ammo yanada ko‘proq o‘tkazish qobiliyatiga va yanada kengroq funksiyalar to‘plamiga ega.

NP-3 da amalga oshirilgan funktsiyalardan tashqari, NP-4 oqimli Видеони va IPTV ni qo‘llab-quvvatlaydi, kommutatsiyalashning tashqi matritsalariga ega ichiga qurilgan blok, trafikni shakllantirish funksiyalari, navbatlarni rejalashtirishning ko‘p sonli algoritmlariga ega.

NP-5 keyingi avlod tarmoq protsessori tarmoqlarning 200 Gb/s ishlash qobiliyatini ta’minlaydi. Bunda NP-4 protsessorlari bilan to‘liq teskari dasturiy moslashuvchanlik ta’minlanadi.

Foydalana olish qurilmalari uchun protsessorlarning ishlab chiqaruvchilari orasida eng yirigi PMC-Sierra kompaniyasi hisoblanadi, bu kompaniya 10 Gb/sgacha o‘tkazish qobiliyatini ta’minlaydigan WinPath3 turkumidagi tarmoq protsessorlarini taklif qiladi. Moslashuvchan arxitekturaga ega bo‘lganligi sababli bu qurilmalardan simli hamda simsiz tarmoqlarda ham foydalanish mumkin.

WinPath3 tarmoq protsesorlari umumiy vazifali ikkita uzib qo‘yiladigan hisoblash yadrolariga va paketlarni qayta ishlash uchun mo‘ljallangan 12 tagacha ixtisoslashtirilgan hisoblash yadrolariga ega, ular jami 64 tagacha apparat oqimlarga ishlov beradi. Paketlarning apparat vositalari yordamida amalga oshirilgan tasniflagichi 32 mingtagacha qoidalardan iborat bo‘lishi mumkin, ulardan 16 mingtasi qayta manzillash (forwarding) uchun javob berishi mumkin. Tasniflagich sekundiga 450 mln.gacha qidiruv so‘rovlariga ishlov berilishini ta’minlaydi. Mikroprotsessor 2,5 MB ichki xotiraga va tashqi operativ xotira bilan ishlash uchun uchta shinaga ega.

Tarmoqqa kirish qurilmalari uchun tarmoq protsessorlarining yana bitta yirik ishlab chiqaruvchisi bo‘lib, Axxia (Axxia communication processors) kommunikatsion protsessorlarining yangi turkumini chiqaruvchi LSI kompaniyasi hisoblanadi. Bu ishlab chiqaruvchi protsessorlari o‘zida umumiy vazifadagi hisoblash yadrolaridan Power arxitekturasi va paketlarni qayta ishlash turli vazifalarini:

shablon bilan qiyoslash, tasniflash va trafikni boshqarish uchun bir nechta ixtisoslashtirilgan yadrolardan foydalanadi. Hozirgi vaqtda LSI kompaniyasi tarmoq protsessorlarining bir-biridan strukturasi va unumdorligi bilan farqlanadigan to'rtta modelini yetkazib beradi: ARR3100, ARR3300, ARR650 va ARR300.

Aloqa tizimlari uchun mikroelektronikani eng mashhur bo'lgan ishlab chiqaruvchi Broadcom Ethernet-tarmoqlar uchun tarmoq protsessorlarining keng turlarini taklif qiladi. Bularga 800 MHz dan 1,2 GHz gacha taktli chastotaga ega bir, ikki va to'rt yadroli protsessorlar kiradi. Bu protsessorlar sekundiga 20 mln. paketlarni o'tkazish qobiliyatini ta'minlaydi. Ushbu protsessorlar iste'mol qiladigan quvvat hisoblash yadrolarining soniga bog'liq holda 4 dan 23 Vt gacha yetadi. Bu protsessorlar ixtisoslashtirilgan funksional bloklarga ega emas va paketlarni qayta ishlash bo'yicha butun yuklanish umumiy vazifadagi hisoblash yadrolari zimmasiga tushadi.

StrataXGS seriyasidagi kompyuterlarni alohida qayd etish kerak. Bu seriyaga yuqori darajada yuklangan tarmoqlarda va axborot-markazlarda ishlash uchun mo'ljallangan ixtisoslashtirilgan kompyuterlar kiradi. Ular 240 Gbit/s gacha bo'lgan o'tkazish qobiliyatini ta'minlaydi va quyidagi keng funksional imkoniyatlarni taqdim etadi:

- tunellash protokollarini qo'llab-quvvatlash;
- xizmat ko'rsatishning ierarxiyali modeli (QoS);
- trafikni shakllantirish;
- OAMni qo'llab-quvvatlash;
- vaqtni sinxronlash;
- trafik to'g'risida statistikani yig'ish uchun 100 mingdan ortiq hisoblagichlar.

3.6. Tarmoq protsessorlarining operatsion tizimlari

Ixtiyoriy apparat-dasturiy, shu jumladan real vaqt rejimida ishlovchi kompleks - operatsion tizim (OT) hisoblanadi. Operatsion tizim deb apparat-dasturiy kompleks (hisoblash tizimi) resurslarini boshqarishni ta'minlaydigan dasturlar va hisoblashlarda resurslardan foydalanadigan jarayonlar kompleksiga aytiladi. Ushbu kontekstda resurs hisoblash tizimi yoki apparat-dasturiy kompleksning ixtiyoriy logik yoki fizik (va ularning ikkalasi) komponenti hisoblanadi.

Asosiy resurs bo'lib protsessor, operativ xotira va periferik qurilma hisoblanadi.

Resurslarni boshqarish quyidagi masalalarni yechishga olib keladi: resurslarga kirishni soddalashtirish, ularni jarayonlararo taqsimlash.

Birinchi funksiyani joriy qilish hisoblash tizimi apparat xususiyatlarini «berkitish» imkonini beradi va shu bilan birga foydalanuvchi yoki dasturlovchiga boshqarish yengil bo'lgan virtual mashinada ishlashga imkon beradi. Resurslarni taqsimlash funksiyasi OT bajaradigan muhim masalalardan biri hisoblanadi, shunga qaramay barcha OT larda mavjud emas, faqatgina bir necha dasturlarni bir vaqtda ishlata oladigan OTlarda mavjud. Jarayon deb dasturda yoki uning logik tugatilgan qismida yozilgan amallar, shuningdek hisoblashda foydalaniladigan ma'lumotlar ketma-ketligiga aytiladi. Jarayon resursdan ajratilayotgan minimal ish birligi hisoblanadi.

Hozirgi kunda OT larning quyidagi belgilari bo'yicha klassifikatsiya qilinadigan har xil turlari mavjud:

- Tizim bir vaqtda xizmat ko'rsatayotgan foydalanuvchilar soni;
- OT nazorati ostida bir vaqtda bajarilayotgan jarayonlar soni;
- Foydalanuvchilarning tizimga kirish turi;
- Apparat-dasturiy kompleks turi.

Birinchi belgiga mos holda bir va ko'p foydalanuvchili OT larga bo'linadi. Ikkinchi belgi OTni bir va ko'p topshiriqli turlarga bo'ladi (keyinchalik faqat ko'p topshiriqli OT lar haqida so'z boradi).

Uchinchi belgiga muvofiq OT:

- Paketli ishlov berish tizimi. Bu holatda ishlayotgan dasturda tizimga ishlov berish uchun yuboriladigan paketlar shakllantiriladi. Bunda foydalanuvchilar OT bilan bevosita bog'lanmaydilar;
- Vaqtni taqsimlash tizimlari. Bir necha foydalanuvchi terminallarini bir vaqtda tizimga interaktiv kirishlarini ta'minlaydi. Tizim resurslari har bir foydalanuvchiga u yoki bu xizmat ko'rsatish tartibiga muvofiq navbat bo'yicha ajratiladi.
- Real vaqt tizimlari. Tashqi hodisalarga javob qaytarish kafolatlangan vaqtini ta'minlaydi.

To'rtinchi belgi OTni bir va ko'p protsessorli, tarmoq va taqsimlangan turlarga ajratadi. Ko'p foydalanuvchili va ko'p topshiriqli OT lar uchun xizmat ko'rsatish tartibi muhim ko'rsatkich hisoblanadi.

Shunga muvofiq ko‘p topshiriqli ishning siqib chiqarish va mos kelish rejimlari bor. Siqib chiqarish tashkil qilinganda topshiriqlarni bajarish uchun protsessor vaqtini faqat OT egallaydi (masalan, protsessor barcha topshiriqlar uchun navbat joriy qilgan bo‘ladi, lekin imtiyozli xizmat ko‘rsatish mavjud). Mos kelish tashkil qilinganda har bir topshiriq boshqaruvni olib, protsessorni boshqa topshiriq uchun berishni o‘zi hal qiladi.

Umumiy holda mos kelish rejimi effektiv va ishonchli, ammo dasturlarni yaratishda tavsiflash faktori bo‘lib dastur protsessor vaqtidan monopol tarzda foydalanmasligi kerakligi fakti hisoblanmoqda. Bundan ko‘rinadiki, hozirgi vaqtda OT ning ko‘p turlari mavjud, keyinchalik real vaqt operatsion tizimlari (RVOT) haqida so‘z yuritiladi.

RVOT ni ko‘rib chiqish uchun real vaqt tizimi tushunchasini aniqlashtirish zarur.

Real vaqt tizimi (RVT) – bu aniq ishlash faqatgina hisoblashning logik to‘g‘riligiga bog‘liq bo‘lmay, balki bu hisoblashlar bajariladigan vaqtga ham bog‘liq bo‘ladigan tizim.

Bunday tizimda bo‘ladigan hodisa uchun hodisa sodir bo‘lish vaqti va uning logik to‘g‘riligi muhim.

Tizim real vaqtda ishlaydi, agar tezkorligi nazorat va boshqaruv obyektlaridagi fizik jarayonlarning bajarilish tezligiga mos bo‘lsa (jarayon deganda konkret real vaqt tizimi tomonidan bajarilayotgan funksiya bilan bog‘liq jarayon tushuniladi). Boshqarish tizimi ma’lumotlarni yig‘ishi, ularni berilgan algoritm bo‘yicha qayta ishlashi va boshqaruv ta’sirlarini shunday vaqt oralig‘ida berishi kerakki, bunda qo‘yilgan topshiriqning muvafaqqiyatli bajarilishi ta’minlanishi kerak.

RVT ga asosiy talablar:

- bir necha topshiriqlarni parallel bajara olish;
- oldindan bilish;
- hodisa javobiga maksimal vaqtning muhimligi;
- xavfsizlik masalalari;
- uzoq vaqt rad etishsiz ishlash qobiliyati.

RVT umumiy xarakteristikalari:

- katta va murakkab tizimlar;
- taqsimlangan tizimlar;

- apparat bilan qat'iy ta'sirlashuv;
- topshiriqlarni bajarish vaqtga bog'liq;
- testlashning qiyinligi.

RVT ichki va tashqi hodisalarning har xil turlari (davriy va nodavriy) bilan ta'sirlashishi kerak. Tizimning RVT klassiga tegishli bo'lishi uning tezkorligi bilan bog'liq emas. Tizim reaksiya vaqtiga va boshqa vaqt parametrlariga talablar tizimning texnik topshirig'i yoki oddiygina uning ishlash mantig'i bilan aniqlanadi. RVT tezkorligi nazorat va boshqarish obyektidagi jarayonlarni amalga oshish vaqtidan katta bo'lishi kerak.

Qattiq va yumshoq real vaqt tizimlari mavjud.

Qattiq real vaqt tizimi deb qo'yilgan topshiriqni yechish imkoniyatini yo'qotishga olib keladigan berilgan vaqtdagi qandaydir rad etish hodisasiga reaksiyani ta'minlay olmaydigan tizimlarga aytiladi. Ko'plab nazariyachilar qattiq tizimlarda reaksiya vaqti sekund, soat, haftadan iborat bo'lishi mumkin deyishadi. Shunga qaramay, ko'pchilik amaliyotchilar qattiq tizimlarning reaktsiya vaqti minimal bo'lishi kerak deb hisoblaydilar. Qattiq real vaqt tizimlari asosan nazorat va boshqaruv tizimlari hisoblanadi. Bunday RVT ni joriy qilish qilish qiyin, chunki unga xavfsizlik masalalarining qat'iy talablari qo'yiladi.

Yumshoq real vaqt tizimlari haqida aniq tavsif yo'q, shuning uchun bunga qattiq kategoriyaga kirmaydigan barcha RVT larni kiritish mumkin. Yumshoq real vaqt tizimlari berilgan vaqtda barcha amallarni bajarishga ulgurmasligi mumkin, shuning uchun muvafaqqiyatli ishlash mezonlarini aniqlash muammosi yuzaga keladi.

Bundan tashqari RVT maxsus va universal tizimlarga bo'linadi. Maxsus RVT – konkret vaqt talablariga javob beradigan tizim. Bunday tizim yuqoridagi talablarni qondirish uchun maxsus loyihalashgan bo'lishi mumkin.

Universal RVT maxsus texnikani qo'llamagan holda ixtiyoriy (oldindan belgilanmagan) vaqt topshiriqlarini bajara oladi. Maxsus tizimlarga qaraganda bu tizimlarga qo'yilgan talablar yengilligiga qaramay, bunday tizimlarni yaratish juda qiyin masala hisoblanadi.

IXP1200 ni real vaqt tizimi sifatida ko'rsak, protsessor universal RVT deyish mumkin, chunki u ko'p yoki oz konkret topshiriqlarni

bajarish uchun loyihalashtirilgan, lekin u keng diapazonda foydalaniladi.

RVOT larning imkoniyatlarini chuqurroq ko'rib chiqish uchun reaksiya vaqtlari tartibi haqida taxminiy raqamlar va mos operatsion tizimlar keltirilgan (jadval). Bu jadval Intel 80486DX protsessori asosida qurilgan hisoblash kompleksi bazasida olingan eksperimental ma'lumotlar asosida shakllantirilgan. Shubhasiz, bu protsessor bugungi kunda eskirgan hisoblanadi, ammo turli RVOT tashqi hodisalarida reaksiya sathi xulosalarini keltirish mumkin.

RVOT tashqi hodisalarida reaksiya sathi

3.9-jadval

Reaksiya vaqti	Foydalanilgan OT
10 mks dan kam	Faqat RVOT, lekin hatto ular ham kuchsiz bo'lishi mumkin – bu sxemali va dasturli yechimlar orasidagi tanlash chegarasi
10 – 100 mks	Real vaqt operatsion tizimlari
100 mks – 1 ms	RVOT, RTAI, RT LINUX, Windows NT, CE uchun real vaqt kengaytmalari
1 ms	Linux va Windows NT da nimadir qilishga harakat qilish mumkin, lekin reaksiyaning kechikishi yomon vaziyatlarga olib keladigan tizimlar uchun emas.

Bunda RVOT ning vaqt chegaralari qat'iy ko'rinadi. Zamonaviy operatsion tizimlar orasida qattiq real vaqt tizimlari VxWorks, OS9, QNX, LynxOS, OSE va boshqalarni qurish uchun maxsus yaratilgan tovarlar klassi mavjud. Bu tizimlarda kerakli instrumentlar to'plami mavjud va ayrim holatlarda yagona tanlov bo'ladi, xarajatlariga qaramay shu tizimni tanlashga to'g'ri keladi. Shunga qaramay, ko'p hollarda real vaqtga talabni (reaksiya vaqtlarini oldindan to'liq bilish) kompromislar qo'yadi, masalan, faqatgina kerakli o'rtacha samaradorlikka erishish talab etiladi.

Ba'zida faqatgina bir hodisani qat'iy nazorat qilish yetarli, shunda qolganlari uchun reaksiyaning kechikishi bo'lmaydi. O'xshash hodisalarda tanlov imkoniyati kengayadi va LINUX, Windows NT,

Windows CE kabi keng tarqalgan operatsion tizimlarni real vaqt kengaytmalari (RTAI, RT LINUX, RTX) bilan to'ldirib foydalanish bilan ko'zlangan natijaga erishish mumkin.

RVOT ni proyektlashda taqdim etilgan OT ga talablar

Talab 1. OT ko'p tolali (multi-threaded) va uzilishli bo'lishi kerak

Yuqorida ko'rsatilganidek, RVOT u yoki bu amal bajarilishi maksimal vaqtini oldindan biladigan bo'lishi kerak, ilova talablariga mos kelishini tekshirish uchun oldindan bilish kerak bo'ladi.

Birinchi talab shundan iboratki, OT absolyut imtiyoz (uzilishli) tamoyili bo'yicha ko'p tolali bo'lishi kerak. Rejalashtiruvchi ixtiyoriy tolani uzib, uning resursini yanada ko'proq zarur joyga berish imkoniyatiga ega bo'lishi kerak. OT (va apparatura) ham uzilishlarni qayta ishlash sathida uzilishlarni ta'minlashi kerak.

Talab 2. Tolalar imtiyoz tushunchasi bo'lishi kerak

Muammo shundaki, qaysi topshiriqqa resurs talab qilinayotganligini bilish zarur. RVOT ideal holatda tola resursini drayverga beradi (Bu OT lar cheklangan vaqtni boshqarish OT deyiladi (deadline driven OS)).

Buni ta'minlash uchun OT toladagi har bir amalning tugash vaqtini bilishi kerak (haligacha bu printsipda qurilgan OT yo'q), shuning uchun OT ishlab chiqaruvchilar boshqa nuqtayi nazarni qabul qilishadi: topshiriqning imtiyoz sathi tushunchasi kiritiladi va vaqt bo'yicha cheklanishlar imtiyoz bo'yicha bajariladi. Shu bilan mushohadaga asoslanib, yechim xatolarni keltirib chiqarsa, bunda RVT ko'rsatkichlari pasayadi. Yuqorida keltirilgan cheklanishlarni qayta ishlashni yanada effektiv qilinsa, loyihalovchi jadval nazariyasi yoki imitatsion modellashtirishdan foydalanishi mumkin, hatto bu ham foydasiz bo'lishi mumkin. Bugungi kunda bundan boshqa yechim yo'q shuning uchun tolalarning imtiyozi tushunchasi juda muhim.

Talab 3. OT topshiriqlar sinxronizatsiyasi mexanizmini oldindan bilishni ta'minlashi kerak.

Topshiriqlar ma'lumotlar (resurslar)ni bo'lib beradi va ular bir-biri bilan aloqada bo'lishi kerak, o'z navbatida blokirovka va kommunikatsiya mexanizmlariga ega bo'lishlari kerak.

Hozirgi vaqtda 100 dan ortiq pulli RVOT lar mavjud. Ko'pgina bepul (yoki shartli bepul) RVT va tadqiqot yoki universitet proyektlari maqomidagi tizimlar bor. Bir necha real vaqt tizimlarining qisqa tavsifini ko'rib chiqamiz.

QNX operatsion tizimi.

QNX operatsion tizimi ishlab chiqaruvchisi Kanadadagi QNX Software System Ltd (1981) kompaniyasi hisoblanadi.

QNX operatsion tizimi foydalanuvchiga o'z talabi bo'yicha konfiguratsiya qilish imkonini beradigan, 16/32 bitli gibrid operatsion tizim hisoblanadi.Ko'pincha u real vaqt masshtabida ishlovchi tizimlarni yaratish uchun ishlatiladi. Tizimni to'liq o'rnatish uchun kerakli vaqt tarmoq vositalari bilan 10 – 15 minutni tashkil etadi, shundan so'ng ishni boshlash mumkin. Tizimning resurslarga talabi yo'qligi tizimning Watcom C/C^{++} (QNX uchun asosiy kompilyator) kompilyatori ko'rinishidagi kerakli va yetarli yaratish muhiti ekanligi bilan ko'rsatiladi.

QNX – mikroyadro va xabar almashish prinsipida qurilgan birinchi pulli OT. Tizim turli darajadagi (menedjerlar va drayverlar), har biri ma'lum xizmat ko'rinishini beradigan erkin (ammo xabar almashish orqali o'zaro ta'sirlashadigan) jarayonlar birligi ko'rinishida yaratilgan.

Bu g'oyalar bir necha muhim afzalliklarga erishishga imkon berdi:

- oldindan bilish, qat'iy real vaqt topshiriqlariga nisbatan qo'llaniladi; UNIX ning hech bir versiyasi yadro kodi juda kattaligi tufayli bunday sifatga erisha olmaydi. UNIX da uzilishlarni qayta ishlashdan kelgan ixtiyoriy tizim chaqiruvi oldindan aytib bo'lmaydigan (Windows NT kabi) kechikishga olib keladi;
- kengayuvchanlik va effektivlik, resurslardan oqilona foydalanish bilan erishiladi va ichki qurilgan (embedded) tizimlar uchun qo'llash mumkinligi bilan izohlanadi. Dev katalogida faqat zarur drayverga mos fayl topshiriqlari uchun kerakli ma'lumotlar bor. Drayver va menedjerlarni ishga tushirish va komandalar qatoridan oddiygina dinamik o'chirish (fayl tizimidan tashqari) mumkin. Faqatgina kerakli funksiyani ta'minlash uchun zarur bo'ladigan real modullarnigina sotib olish mumkin;

• bir vaqtda kengayuvchanlik va ishonchlilik, tizim barqarorligini yuzaga keltirmaslik uchun bu drayverni yadroga kompilyatsiya qilish kerak emas.

Tarmoqning bir uzelida ishlayotgan foydalanuvchi qolgan ixtiyoriy uzellar, portlar, fayl tizimlari va topshiriqlardagi resurslarga kira oladi. Foydalanuvchi tarmoq protokoliga murojaat qilishi shart emas. U xabarlarni uzatish uchun ham qo'llaniladigan paketni o'z ichiga oladi. Tarmoq administratori bu paketni taniydi va mikro-yadroga yo'naltiradi, o'z navbatida u ham paketlarni lokal xabarlar shinasiga uzatadi. QNX faqatgina QNX jarayonlari xabar paketlarini tanimaydi. Shuningdek, TCP/IP, 8MB va boshqa paket protokollarini uzatish uchun tarmoq administratoriga oson murojaat qilish mumkin. Turli tarmoq administratorlariga bitta kabel orqali murojaat qilish mumkin.

QNX operatsion tizimi butun PK tarmog'ini kirish shaffofligi absolyut bo'lgan yagona resurslar to'plamiga birlashtiradi. Tugunlar tarmoqqa qo'shilishi yoki chiqarib tashlanishi mumkin, bu tizim butunligiga ta'sir etmaydi. QNX da tarmoq ma'lumotlarni qayta ishlash shunday egiluvchanki, bunda muvofiq Intel kompyuterlarning ixtiyoriy turli to'plamini Arcnet, Ethernet, Token Ring yoki modem ham ulanishi mumkin bo'lgan serial porti orqali bog'lab bir tarmoqqa birlashtirish mumkin. Bundan tashqari kompyuter bir vaqtda bir necha tarmoqqa ulanishi mumkin, agar ulardan biri o'ta yuklansa yoki ishdan chiqsa, bunda QNX avtomatik tarzda boshqa mumkin bo'lgan, axborot yo'qotishi bo'lmagan tarmoqdan foydalanadi.

QNX ichki qurilgan real vaqt tizimlari bozorida tizimning o'rni bilan bog'liq bir necha cheklovlarga ega:

• SMP ni qo'llamaydi;
• Virtual xotiradan diskka yozish mavjud emas;
• Toladan noeffektiv va nostandart foydalanish;
• Fayllarni xotirada aks ettirishning noto'liq aks ettirilishi;
• UNIX-domain sockets ni qo'llamasligi;
• Xususiy tarmoq protokoli doirasidagi xavfsizlik vositasining zaifligi.

Keltirilgan kamchiliklarga qaramay, QNX ko'pgina foydalanuvchi dasturlarida ishlangan (masalan, ma'lumotlar ombori).

VxWorks operatsion tizimi.

Wind River Systems firmasining VxWorks real vaqt operatsion tizimi va Tornado instrumental muhiti qat'iy real vaqt tizimida ishlovchi kompyuterlar dasturiy ta'minotini yaratish uchun mo'ljallangan. VxWorks operatsion tizimi amaliy dasturiy ta'minot yaratish kross vositali tizimi hisoblanadi, yaratish ishlari instremental kompyuter (host) da Tornado muhitida olib boriladi, keyinchalik VxWorks nazorati ostida maqsad mashinasida (target) foydalaniladi.

VxWorks operatsion tizimi qat'iy real vaqt operatsion tizimlari kabi mikroyadro texnologiyasi bo'yicha qurilgan, ya'ni yadroning quyi uzluksiz sathida topshiriqlarni rejalashtirish va ularning kommunikatsiya sinxronizatsiyasini boshqarish asosiy funksiyalari bajariladi. Operatsion tizimning qolgan barcha yuqori darajali funksiyalari (xotira, kiritish/chiqarish, tarmoq vositalari va boshqalarni boshqarish) quyi daraja oddiy funktsiyalariga asos qilinadi, bu tezkorlik va operatsion tizim zaruriy konfiguratsiyalarini oson qurish imkoniyatini beradi.

Wind ko'p topshiriqli yadrosida imtiyozni hisobga oladigan va uzilishlar bo'yicha ishga tushadigan topshiriq rejalashtirish algoritmi qo'llanilgan. Wind yadrosida topshiriqlarni sinxronlash asosiy vositasi sifatida va umumiy resurslarga o'z-o'zini inkor etuvchi kirish sifatida semaforlar ishlatilgan. Turli amaliy topshiriqlarga nisbatan semaforlarning bir necha ko'rinishlari mavjud: ikkilik, butun sonli, o'z-o'zini inkor etuvchi va POSIX.

VxWorks bazaviy tarmoq vositalari: UNIX-networking, SNMP va STREAMS

VxWorks real vaqt talablarini hisobga olgan holda TCP/IP protokoli bilan yaratilgan birinchi real vaqt operatsion tizimi bo'lgan. VxWorks hozirgacha UNIX uchun standart bo'lgan barcha tarmoq vositalarini qo'llaydi: TCP/UDP/ICMP/IP/ARP, Sockets, SLIP/CSLIP/PPP, telnet/rlogin/rpc/rsh, ftp/tftp/bootp, NFS (klient va server).

SNMP-agentini MIB-I ni qo'llagan kabi MIB-II ni qo'llab joriy qilish intellektual tarmoq qurilmalari (hablar, ko'priklar, marshrutizatorlar, qaytargichlar) va tarmoqda ishlovchi boshqa qurilmalarda VxWorks ni qo'llash uchun mo'ljallangan.

Tornado instrumental muhiti ochiq arxitekturaga ega, real vaqt DT ishlab chiqish instrumental vositasini yaratuvchi boshqa

firmalarga o‘z dasturiy mahsulotlarini Tornado bilan integrallashtirish imkonini beradi. Foydalanuvchi o‘zining xususiy maxsus ishlab chiqish vositasini Tornadoga ulay oladi, shuningdek, Wind River Systems firmasi instrumental vositasi imkoniyatlarini kengaytiradi.

IXP12xx da qurilmasi yaratish jarayonini tezlashtirish uchun Intel o‘z bo‘linmasida yaratilgan (Level One kompaniyasi) IXDP1200 Advanced Development Platform apparat-dasturiy kompleksni tavsiya qiladi. Kompleksning apparat qismi Ethernet va Serial portli IXP1200 tarmoq protsessori platasini, CompactPCI slotli shassini o‘z ichiga oladi, qo‘shimcha ravishda WAN yoki LAN interfeysli kiritish/chiqarish platasi o‘rnatilgan bo‘lishi mumkin.

Dasturiy ta’minot tarkibiga IXP asosida dastur yaratish uchun zarur muhitni o‘z ichiga oluvchi Intel IXA SDK kirishi mumkin.

Yaratish muhiti Intel Internet Exchange Architecture SDK paketini o‘z ichiga oladi. Bu paket modulli dasturlash vositasiga ega: Intel Active Computing Element, ko‘chuvchi komponentlarni yaratish va ulardan dasturiy ta’minot qurish, ularni birlashtirib paketlarni qayta ishlash funksiyasini joriy qilishga imkoniyat beradi. Operatsion tizim – telekommunikatsiya qurilmasi dasturiy ta’minotining asosi. Operatsion tizim topshiriqlarni dispetcherlashni (har bir vaqt momentida ularning qay biri bajarilishi kerak) ta’minlaydi va topshiriqlararo ma’lumotlar almashinuvini boshqaradi. IXP1200 uchun dasturni ishlatish muhiti ichki o‘rnatilgan (embedded) real vaqt operatsion tizimi (VxWorks yoki embedded Linux) hisoblanadi.

Intel Embedded Linux Integrated Development Environment ishlab chiqish muhiti komanda qatoridan foydalanmay Linux uchun foydalanuvchining grafik interfeysini ishlatishga imkon beradi.

Linux operatsion tizimi.

Linux – shaxsiy kompyuter va ishchi stantsiyalar uchun zamonaviy POSIX bilan hamkorlikdagi Unix ga o‘xshash operatsion tizim, ya’ni ko‘p foydalanuvchili tarmoq operatsion tizimi.

Linux OT ochiq tizim standartlari va Internet tarmog‘i protokollarini qo‘llaydi. Tizimning barcha komponentlari, jumladan joriy matnlar cheklanmagan sonli foydalanuvchilar uchun erkin ko‘chirish va o‘rnatish litsenziyasi bilan tarqatiladi.

Linux ning OT sifatida xarakterli xususiyatlari:

- ko‘p topshiriqlilik (zaruriy shart hisoblanadi);

- ko‘p foydalanuvchili rejim;
- protsessorning himoyalangan rejimi (386 protected mode);
- protsessor xotirasi himoyasi; dasturning ishdan chiqishi tizimga bog‘liqlikni vujudga keltirmasligi kerak;
- TCP/IP tarmog‘i, shu jumladan ftp, telnet, NFS va boshqalarni qo‘llash.

Linux ning ommalashib borishi ishlab chiqaruvchilarning bu operatsion tizimga diqqat bilan e’tibor qaratishlariga olib keldi. Hozirgi vaqtda bu OT barqaror ishlashga tayyor, uning boshlang‘ich matnlari va arxitekturasining ochiqligi esa dasturchilarni ko‘pgina apparat platformalariga (SGI, IBM, Intel, Motorola va boshqalar) o‘zlarining qo‘shimchalarini kiritishlariga imkon beradi.

4. MIKROKONTROLLERLI TIZIMLARI

4.1. Mikrokontrollerli tizimlar strukturasi

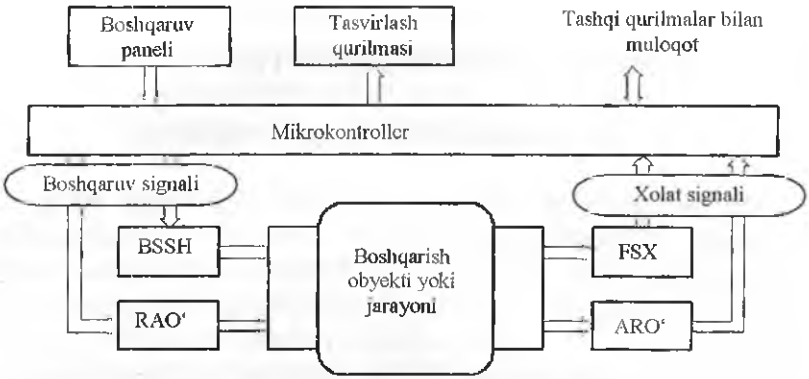
Mikrokontrollerlar – hisoblash asboblari, qurilmalar va har xil vazifalarda qo'llaniladigan tizimlarda ishlatiladigan mikroprotsessorlarni eng keng sinfini tashkil qiladi. Mikrokontrollerlar bu – texnik obyektlarni boshqarish qurilmalarni va texnologik jarayonlarni hosil qilish uchun mo'ljallangan maxsus mikroprotsessordir [14-17]. Tuzilishi jihatidan mikrokontrollerlar, kristalda hisoblash tizimining hamma tarkib qismlari: mikroprotsessor, xotira, hamda qo'shimcha finksiyalarni amalga oshirish uchun periferiya qurilmalari joylashtirilgan, katta integral sxemani (KIS) tashkil qiladi.

Mikrokontrollerning hamma elementlari bitta kristalda joylashgani uchun, ularni bir kristalli (bir korpusli) mirkoEXM yoki bir kistalli mikrokotrollerlar deb ham atashadi. Mikrokontrollerlarni qo'llashdan maqsad – komponentlar sonini qisqartirish, o'lchamini kamaytirish va qurilmani (tizimni) narxini tushirishdir.

Odatda, mikrokotrollerlar RISC-arxitekturasiga (RISC – Reduced Instruction Set Computer), kam hajmli xotiraga, fizik va logik bo'lingan dastur xotirasi va komandalar tizimini boshqarish uchun mo'ljallangan ma'lumot xotirasiga ega. Shunday qilib, mikrokontrollerlar boshqarish masalasini yechish, nazorat, tartibga solish va ma'lumotlarga dastlabki ishlov berish uchun mo'ljallangan.

Mikrokontrollerli boshqaruv tizimiga mikrokontroller va u bilan boshqaruv obyektini ulash (biriktirish) qurilmasi kiradi (4.1-chizma) [14].

Mikrokontroller obyekt bo'yicha holat signallarini davriy so'rab turadi va joylashtirilgan algoritmga muvofiq boshqaruv signallarini ketma-ketligini ishlab chiqaradi. Holat signallari boshqaruv signallarini joriy parametrlarini xarakterlaydi. Ular datchikning (D) chiqish signallarini analog-raqamli o'zgartirgich (ARO') yoki signallar holatini shakllantirgich (ShSh) yordamida o'zgartirish yo'li bilan shakllanadi.



4.1-chizma. Mikrokontroller asosidagi boshqaruv tizimining odatiy tuzilishi: BSSh – boshqaruv signallarini shakllantirgichlar; BQ – bajaruvchi qurilma; D – datchiklar; ShSh – signallar holatini shakllantirgichlar

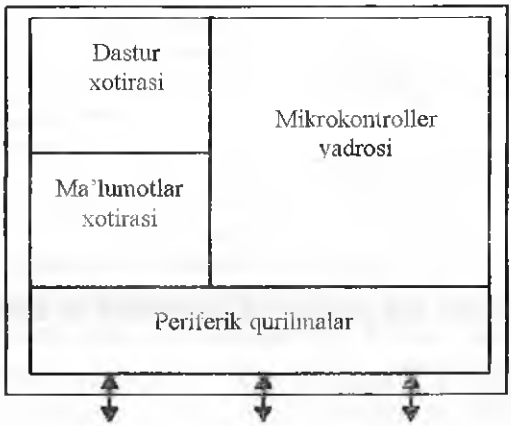
Mikrokontroller orqali tanlangan Boshqaruv signallari, raqamli-analog o‘zgartirgich (RAO‘) yoki boshqaruv signallarini shakllantirgich (BSSh) yordamida o‘zgartiriladi. Bajaruvchi qurilmaga (BQ) keladigan RAO‘ va BSSh chiqish siganllari mos ravishda analog va diskret boshqaruv ta’siriga ega. Tizimida yana boshqaruv paneli, indikatsiya qurilmasi va tashqi qurilma bilan ma’lumot almashtirib turish uchun interfeys bo‘lishi mumkin. Ma’lum tizimning vazifasi va xarakteristikasiga qarab yuqorida ko‘rsatitilgan elementlarning ba’zi biri mavjud bo‘lmasligi mumkin.

4.2. Mikrokontrollerlar strukturasi va ishlash asoslari

Mikrokontroller bitta integral sxema ko‘rinishida amalga oshiriladigan hisoblash tizimidan iborat va o‘z ichiga yadro, dastur xotirasi, ma’lumotlar xotirasi, periferiya qurilmalari kabi asosiy bloklarni oladi (4.2-chizma).

Mikrokontroller yadrosi dastur tomonidan beriladigan boshqaruv jarayonini amalga oshiradi. Mikrokontrollerli yadro negizida integral sxemasini ishlab chiqaruvchi zavod tomonidan modulli xotira va

periferiya qurilmalarining nom ro'yxati bo'yicha turlicha bo'lgan, lekin komandalar tizimi va ma'lumotlar almashinuvi sikli bo'yicha o'zaro moslashadigan mahsulotlar ishlab chiqariladi. Ushbu belgi bo'yicha ko'plab moslashadigan mikrokontroller (MK) mikrokontroller turkumi deb ataladi.

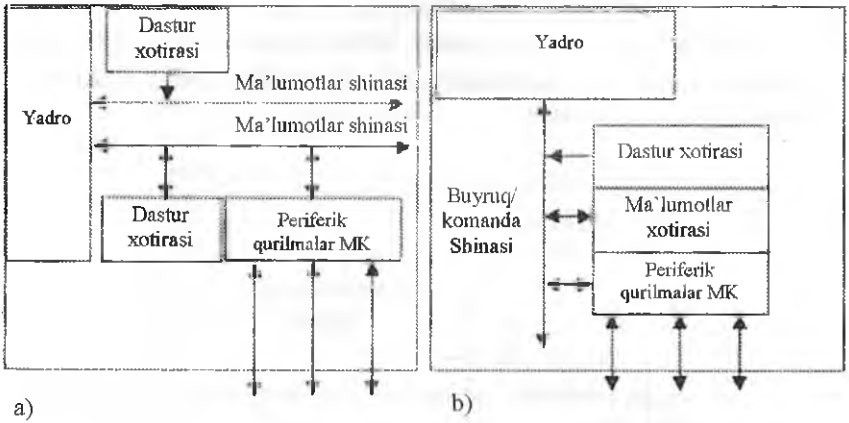


4.2-chizma. Mikrokontrollerning umumlashtirilgan tuzilmaviy sxemasi

Dastur xotirasi boshqaruvchi dasturni saqlash uchun mo'ljallangan. Boshqarish jarayoni uchun zarur bo'lgan ma'lumotlar ma'lumotlar xotirasida joylashadi.

Periferiy qurilmalari mikrokontrollerning tashqi obyektlari va qator boshqaruv funksiyalarini amalga oshiradigan apparat bilan birikishni ta'minlash uchun mo'ljallangan.

Mikrokontrollerlar, boshqa klasslarning hisoblash mashinalari kabi, Garvard yoki Prinston arxitektura asosida amalga oshiriladi (4.3-chizma). Garvard arxitektura asosida bajariladigan mikrokontrollerlarda dasturlar va ma'lumotlar foydalana olishning turli metodlaridagi mantiqiy bog'liq bo'lmagan xotira bloklarida joylashadi. Prinston arxitektura asosida bajariladigan mikrokontrollerlarda dasturlar va ma'lumotlar xotiraning umumiy blokida joylashishi mumkin, murojaat uchun foydalana olishning yagona metodidan foydalaniladi.



4.3-chizma. MK garvard (a) va prinston (b) arxitekturalari

Namunaviy va kristalga eng ko'p integratsiyalanadigan periferiya qurilmalarining mikrokontrolleriga quyidagi bloklar kiradi:

- mantiqiy signallar ko'rinishida keltirilgan ma'lumotlar almashinuvini amalga oshiradigan kiritish-chiqarish parallel raqamli portlar;
- vaqtli integrallar shakllantirilishini amalga oshiradigan va mantiqiy hodisalarni hisoblashni bajaradigan taymer-hisoblagichlar;
- vaqt bo'yicha bog'liq bo'lgan hodisalarni apparatli qayta ishlash uzellari;
- uzluksiz signallarni chiqarish va kiritishni amalga oshiradigan raqamli-analog va analog-raqamli o'zgartirgichlar;
- taqsimlanadigan tizimlarda ma'lumotlar almashinuvini amalga oshiradigan kiritish-chiqarishning ketma-ket portlari;
- uziluvchi hodisalarga xizmat ko'rsatish bloklari;
- ishlash ishonchliligini oshirish vositalari.

MK har bir periferiya uzeli maxsus funksiyalar registri deb ataladigan uzelning dasturiy qulay bo'lgan konfiguratsion regisrida boshqariladigan kodlarni yozish yordamida rostlash imkoniyatiga ega bo'ladi. Rostlash qurilmaning (masalan, taymer va parallel porti

razryadlaridagi ma'lumotlarni uzatish yo'nalishlarining razryadliligini va boshqalarni talab etadigan) ishlash rejimini tanlashni amalga oshirish imkonini beradi.

MKda joylashgan periferiya bloklarining tarkibi qurilmaning maqsadli vazifasiga bog'liq bo'ladi va ushbu turkumning mikrokontrollerida amalga oshiriladigan namunaviy vazifalari asosida ishlab chiqaruvchilari tomonidan aniqlanadi.

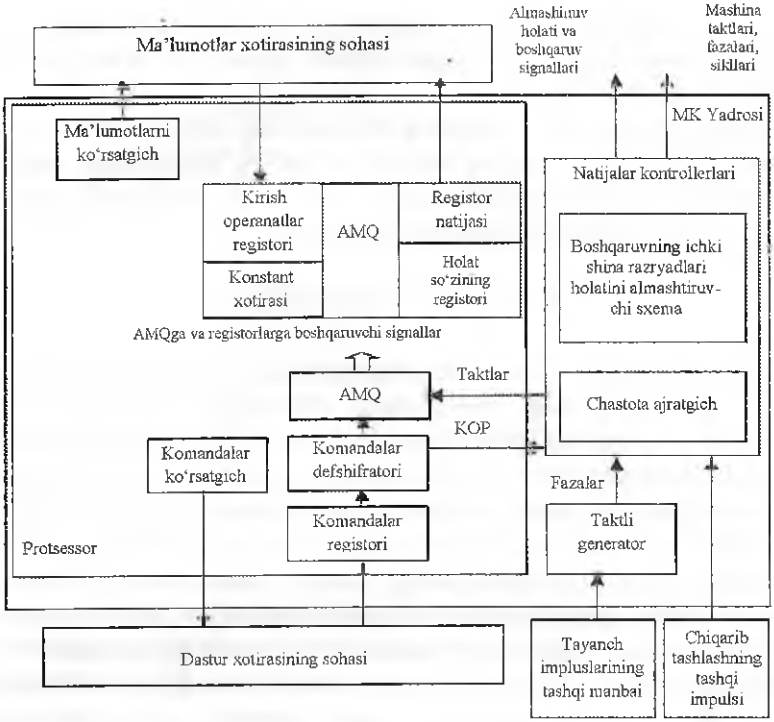
4.3. Mikrokontroller yadrosi

Mikrokontroller yadrosi tarkibiga protsessor, taktli generator va shina kontrolleri kiradi (4.4-chizma). Protsessor ikkilik kodi va komandalar ketma-ketligidan iborat bo'lgan dasturga muvofiq ushbu jarayonni boshqarish ko'rinishida keltirilgan axborotni qayta ishlash jarayonini bevosita amalga oshiradi. Taktli generator MK uzellarida jarayonlar o'tishini sinxronizatsiyalovchi tayanch signallar ketma-ketligini tayanch impulslarning tashqi ketma-ketligi asosida shakllantirishni amalga oshiradi. Shinalar kontrolleri MKda komandlar bajarilishining turli bosqichlarini ichki shina bo'yicha taktlovchi va MK periferiya qurilmalari bilan ma'lumotlar almashinuvini tashkil etish uchun zarur bo'lgan ko'p fazali impulsli ketma-ketlikning tuzilishini amalga oshiradi.

Komandalar berilgan adreslar (katakchalar raqami) bo'yicha komandalar xotirasida joylashadi va o'z ichiga bajariladigan operatsiyalarni tavsiflaydigan boshqariluvchi kodlarni va berilgan operandlar (amallar bajariladigan ma'lumotlar)ni oladi.

Har bir MK komandalar ro'yxati va ularning formatlari xarakterlanadigan komandalarning muayyan tizimiga ega bo'ladi. Komandalar ro'yxati o'z ichiga ushbu MK protsessorida bajarilishi nazarda tutilgan amallar to'plamini oladi. Har qanday MK komandalar ro'yxatida amallarning to'rtta guruhiga ajratish mumkin:

- ma'lumotlar uzatish amallari (MK katakchalar, shuningdek MKning boshqa dasturiy qulay elementlar o'rtasida);
- arifmetik amallar («VA», «YoKI», «YoKI»ni istisno qiladigan inversiya, turli siljishlar);



4.4-chizma. Mikrokontroller yadrosining tuzilmaviy sxemasi

- boshqaruvni uzatish amallari (berilgan adres bo'yicha shubhasiz o'tish, operandlar tengsizligi yoki tanglik sharti bo'yicha o'tish, quyi dasturga o'tish va undan qaytarish va h.).

Komandalar formati amallar dasturining navbatdagi qadamida bajariladigan tipni, kirish va chiqish operandlarini, shuningdek dasturning quyidagi qadamda bajarilishi kerak bo'lgan komandalar adresini aniqlash imkonini beradi.

Bajariladigan komandalar tipi amallar kodi (KOP) beriladi.

Operandalarni berish uchun ularni lokalizatsiyalash metodlari (adreslash usullari) qo'llaniladi:

- noaniq bo'lgan: operand undan foydalana olishning bir xilligi bilan bog'liqligi (masalan, uning joylashish imkoniyatiga bog'liqligi) ko'rsatilmaydi;

- bevosita: kirish operand komandalarda (masalan, konstant topshirig‘i maqsadida) joylashtiriladi;

- to‘g‘ri: kirish operand komandalarda (masalan, konstant topshirig‘i maqsadida) joylashtiriladi;

- to‘g‘ri: komandada operand joylashgan ma’lumotlar xotirasidagi adres ko‘rsatiladi;

- bevosita: komandada operand joylashgan ma’lumotlar xotirasidagi katakchalar adresini o‘z ichiga olgan ma’lumotlar xotirasidagi katakchalar adresi ko‘rsatiladi (masalan, dastur uchastkasining bir necha marta takroranganda ketma-ket joylashgan ma’lumotlardan foydalana olishni tashkil etishda izlanayotgan ma’lumotlar adresini o‘zgartirgan holda komandalar operandining qiymatini o‘zgartirish);

- nisbiy: komandada (masalan, noaniq beriladigan) ayrim kattalikka tuzilgan ma’lumotlar xotirasida katakchalar adresi ko‘rsatiladi, izlanayotgan operand joylashgan ma’lumotlar xotirasidagi katakchalar adresini beradi (masalan, ma’lumotlar jadvalining elementiga murojaat qilganda jadval boshiga nisbatan siljishi bo‘yicha izlanayotgan operandni aniqlash qulaydir).

Quyidagi bajariladigan komandalar adresi komandalarning ushbu vaqtida bajariladigan adresdan keyin keladigan dastur xotirasining adresi kabi noaniq beriladi, bu ko‘pgina dasturlarda komandalar ketma-ketligining liniyali uchastkalarga ega bo‘lish bilan tushuntiriladi. Sikllar, quyi dasturlar, shartlar bo‘yicha vetalenieni va h tashkil etishda uning aniq topshirig‘i uchun komandalar qo‘llaniladi, KOP boshqaruvni uzatishning muayyan amalini kodlaydi.

Ko‘plab MK komandalar tizimiga bir, ikki, uch adresli va adressiz komandalarni (bitta operandalar komandasida adreslanadigan miqdori bo‘yicha) kiritadi.

MKda komandalarni bajarish protsedurasi quyidagiga olib keladi.

Chiqarib tashlash impulsi amalining tugashi bo‘yicha MK yadro registrini initsializatsiyalash amalga oshiriladi. Komandalar ko‘rsatkichiga dastlabki ishga tushirish adresi kiritiladi.

Komandalar ko‘rsatkichidagi adres bo‘yicha shina kontroller tomonidan shakllanadigan boshqariluvchi signallar ta’siri ostida dastur xotirasi sohasidan komandalar registriga dastur kontrolleri tomonidan bajariladigan navbatdagi komandalar yuklanadi.

Har qanday komandalar elementar harakatlar (mikroamallar) ketma-ketligini, operandlar amali uchun talab etiladigan miqdorni aniqlash, zarur operandlarni lokalizatsiyalashni aniqlash, ularni chiqarib tashlash, bajariluvchi bloklar uchun harakatlar kodini shakllantirish, amallar bajarilishi tugashini kutish, natijalarni lokalizatsiyalashni aniqlash, natijalarni kiritish, keyingi komandalar va qator boshqa komandalar adresini aniqlashdan iborat. Navbatdagi komandalarni bajarishda amalga oshiriladigan mikroamallarning muayyan ro'yxati uning KOPni aniqlaydi.

Talab etiladigan mikroamalga protsessor sxemasini rostlash uchun boshqariluvchi signallar ketma-ketligidan foydalaniladi. Komandalar dasturi xotirasidan hisoblab chiqilgan KOP deshifrlanadi va sinxronizatsiyalashning har bir takt bilan ishlab chiqiladigan, boshqariluvchi signallar to'plamining komandalarini ishlab chiqishning ushbu bosqichida zarur bo'ladigan shina kontrolleridan kelib tushadigan mikrodasturli avtomat (MPA)ga kelib tushadi.

Arifmetik va mantiqiy amallar protsessorida siljish, nollash va hokazolarni bajarish arifmetik- mantiqiy qurilma (AMQ) tomonidan ta'minlanadi. Xotira, konstant ma'lumotlarning ikki-o'ntalik tarzda keltirishda to'g'rilovchi kodni, bitlar ustida amallarni AMQda bajarishda maska kodini ishlab chiqishni, shuningdek konstant kodlarini berishni ta'minlaydi. Amallar AMQda bajariladigan ma'lumotlarni vaqtli saqlash uchun tegishli ko'rsatkich qo'llanilishi bilan ma'lumotlar xotirasining sohasidan kiritiladigan kirish amallarining registrlari, axborot mo'ljallangan.

Amallarni AMQda bajarilishining tugashi bo'yicha uning natijalari natijalar registriga kiritiladi, shuningdek protsessor holatining so'z registriga kiritiladigan va dastur bilan hisoblash va tahlil qilish uchun (masalan, arifmetik jihatdan to'lganligi sababli dasturning boshqa shoxiga o'tishni tashkil etish uchun) qulay bo'lgan amallar natijalarining belgilari (to'ldirish, siljish, belgi va h)ni shakllantiradi. Keyin umumiy holatda ma'lumotlar ko'rsatkichiga komandlar natijalarini joylashtirish zarur bo'lgan ma'lumotlar xotirasi katakchalarining adreslari ketma-ket kiritiladi (ushbu adreslar chiqish operandlarni adreslash maydonlaridan chiqarib tashlanadi) va operandalar natijalarning registrlaridan ma'lumotlar ko'rsatkichi tomonidan adreslanadigan ma'lumotlar xotirasi katakchalariga kiritiladi. (Shuni

ta'kidlash kerakki, qoidaga ko'ra, MK protsessorida operandlarni kiritishda avtooshirish va xotiradan operandlar chiqarib tashlanganda avtokamaytirish mexanizmlari bilan ta'minlangan ma'lumotlar xotirasida qo'shimcha ko'rsatkichdan foydalaniladi. Xotiradan foydalana olish metodi stekli deb ataladi, ma'lumotlar xotirasida ajratiladigan bunday manipulyatsiyalash uchun soha stek deb ataladi. Stekdan quyi dasturni tashkil etishda, xususan, uzilishni qayta ishlash quyi dasturdan foydalaniladi).

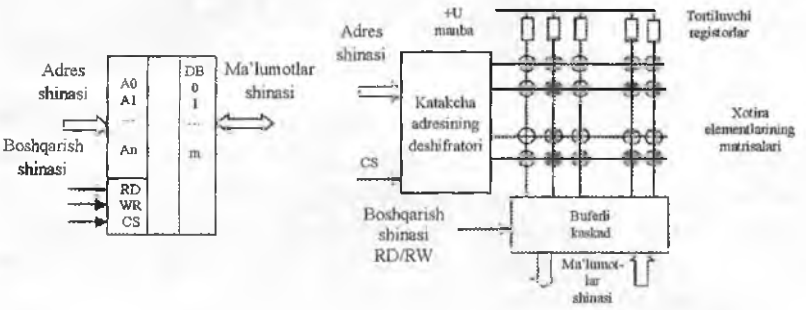
Chiqish natijalari joylashtirilgandan keyin komandalar ko'rsatkichining avtooshishi yuzaga keladi, yoxud bajariladigan komandalarda operandlarning mavjud berilgan maydoni unga kiritiladi. Ikkala holatda komandalar ko'rsatkichida navbatdagi bajarilishi kerak bo'lgan komandalarni o'z ichiga olgan xotira katakchalari adresi bo'ladi va tavsiflangan jarayon takrorlanadi.

Masalan, PLUS opX, opY, opZ (opZ=opX+opY) gipotetik komandalarini bajarish ma'lumotlar ko'rsatkichiga opX adresini yuklash, «ma'lumotlar xotirasini o'qish» signalini berish, ma'lumotlar xotirasining tayyorligini kutish, ma'lumotlar xotirasining ma'lumotlar shinasidan AMQning kirish operandining 1-son registriga yuklash, opY uchun shunga o'xshash harakatlarni bajarish (2-son registrga yozish), «qo'shish» kodini AMQga berish, AMQ tayyorligini kutish, opZadresni ma'lumotlar ko'rsatkichiga yuklash, ma'lumotlarni AMQning kirish operandining 1-son registridan ma'lumotlar xotirasining ma'lumotlar shinalariga berish, «ma'lumotlar xotirasini yozish» signalini berish, ma'lumotlar xotirasi tayyorligini kutish kabi harakatlarni yuzaga keltiradi.

4.4. Mikrokontroller qurilmalari

Xotira qurilmasi. Mikrokontroller kristallida dasturlar xotirasi va ma'lumotlar xotirasi kabi xotiraning ikkita bloki integratsiyalangan. MK avtonom rejimda ishlashining mo'ljallanishiga qarab, dastur xotirasi ta'minot kuchlanishi bo'lmaganda ichidagilarni saqlashi kerak. MKning ichki arxitekturasini soddalashtirish va taktli generatorning chastotalarning keng diapazonida ishlash imkoniyati uchun ma'lumotlar xotirasi statik arxitekturasiga ega bo'lishi (ya'ni regeneratsiyani talab etmasligi) kerak.

Xotira modulining umumlashtirilgan tuzilmasi 4.5.-chizmada keltirilgan. Xotira moduli m-razryadli qatorlarning N ko‘rinishida tashkil etilgan xotira elementlarining matritsalaridan, katakchalar adresining deshifratoridan va buferli kaskaddan iborat bo‘ladi.



4.5-chizma. Xotira modulining umumlashgan tuzilmasi

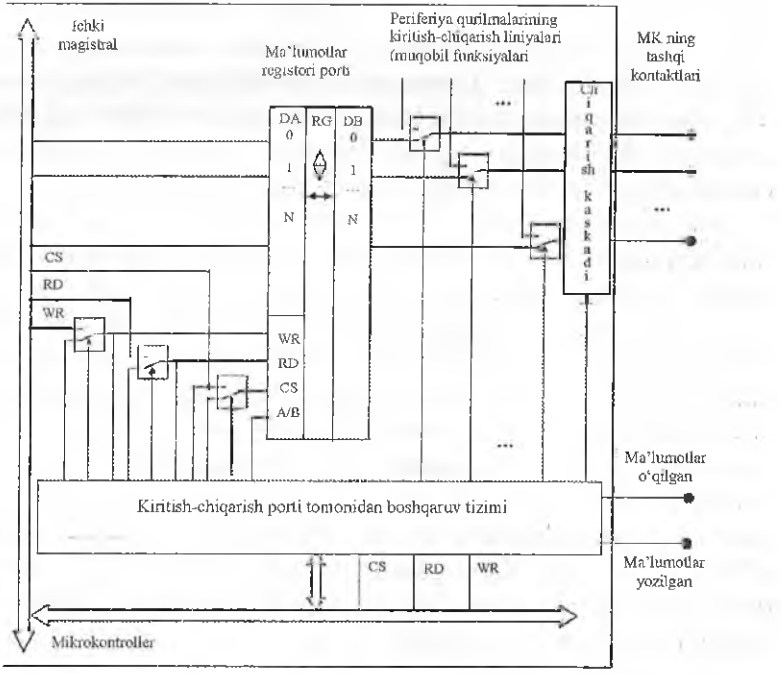
Xotiraning bunday modulining adres shinalarining razryadliligi n-($\log_2 N$) dan iborat, ma’lumotlar shinasining razryadliligi m dan iborat. Adres kodi ko‘rinishdagi katakcha tanlanishini kerak bo‘lgan raqam to‘g‘risidagi axborot o‘zining bitta chiqishidagi yuqori mantiqiy darajani generatsiyalashning xotira elementlarining matritsa qatorlaridan birini aktivlashtiradigan deshifratorga kelib tushadi. Bunda (kelib tushadigan boshqaruv signallariga bog‘liq holda) tanlangan qatorlarning barcha xotira elementlarining mantiqiy darajasi ma’lumotlar shinasiga buferli kuchaytirgich kaskadi orqali kelib tushadi (katakcha holatini o‘qish vaziyati), yoxud ma’lumotlar shinasiga tanlangan qatorning xotira elementlariga buferli kuchaytirgich kaskadlari orqali uzatiladi (katakcha holatini yozish vaziyati).

Kiritish-chiqarishning parallel portlari.

Kiritish-chiqarishning parallel portlari mikrokontroller almashinuvi va MK mikrosxemaning kiritish-chiqarish liniyalari orqali uzatiladigan mantiqiy signallar ko‘rinishda keltirilgan ma’lumotlarning tashqi obyekti uchun mo‘ljallangan. Umumiy holatda har bir port bilan ma’lumotlar registri (MKdan obyekti chiqariladigan axborotni saqlash uchun yoki MKga obyektga kiritiladigan axborotni saqlash uchun), boshqaruv tizimi (portning ishlash rejimlarini berish uchun)

va signallarni kuchaytirish va qo‘shish vazifasini hal etuvchi chiqish kaskadiga bog‘liq bo‘ladi. Port tuzilmasi 4.6-chizmada keltirilgan.

Kiritish-chiqarish portlari MK barcha modellarida amalga oshiriladi. Ma’lumotlar registri o‘z ichiga MKning ichki shinasi bilan axborot registriga kiritish yoki registr holatining ichki shinasiga chiqarish imkonini beradigan yozish va o‘qishni boshqarish liniyalari bilan ikki yo‘nalishli kiritish-chiqarishning N-razryadli registrini oladi. MK yadrosining protsessori va port registri o‘rtasidagi axborot almashinuvi momenti ichki shina orqali yadro shinasining kontrollerini belgilaydi. Almashinuv komandalarining tipiga muvofiq (protsessordan portga yoki portdan protsessorga uzatish) WR yoki RD signallari shakllantiriladi, keyin almashinuv CS boshqaruv signali bilan stroblanadi.



4.8-chizma. Kiritish chiqarishning parallel porti tuzilmasi

Umumiy holatda, almashinuvning juda murakkab vazifalarini hal etish imkoniyati uchun kiritish-chiqarish portining tuzilmasiga

signallar holatini aniqlash va o‘zgartirishning kombinatsiyalanadigan sxemasidan iborat bo‘lgan boshqaruv tizimi, shuningdek ushbu holatni saqlaydigan registrni yozish va o‘qish uchun dasturiy qulay bo‘lgan registr kiritiladi. Bunday vazifa ikkita: kiritish-chiqarish liniyalarining multiplekslash vazifasi va almashinuvning kengaytirilgan protokollarni qo‘llab-quvvatlash vazifasidir. Ularni batafsil ko‘rib chiqamiz.

Ma’lumotlar registridan tashqari, bir xil va xuddi shunday MK kontaktlari ayrim periferiya qurilmasi (masalan, taymer) obyekt bilan (kiritish-chiqarishning muqobil funksiyalari deb ataladigan) axborot almashinuvi uchun talab etilishi mumkin. Bunda qaysi qurilma (porti yoki boshqa periferiya moduli) bilan MK mikrosxemalarning muayyan har bir kontakti mahkamlanganligini aniqlash zarur bo‘ladi. Bunday rostlash signallar kommutatorining holatini belgilaydigan maxsus kodning boshqaruv tizimining registriga yoziladi. (4.8-chizma). Kommutator kontaktining holatiga muvofiq MK mikrosxemaning har bir tashqi kontakti port registrining kiritish-chiqarish liniyalariga, yoxud boshqa periferiya qurilmasining kiritish-chiqarish liniyalariga fizik jihatdan ulangan bo‘ladi.

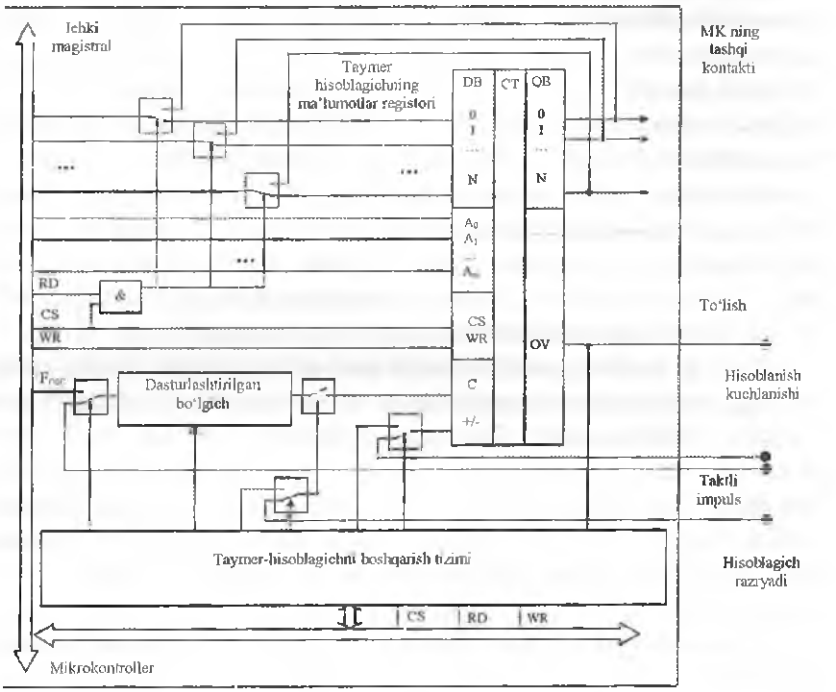
Ma’lumotlarni MKga obyektdan uzatishda obyekt tomonidan shakllanadigan «ma’lumotlar yozilgan» signali ma’lumotlar registrining CS kirishida boshqaruv tizimi bilan kommutatsiyalanadi, boshqaruv tizimi bilan WR signali shakllanadi. Bundan tashqari, «Ma’lumotlar yozilgan» signali boshqaruv tizimi registrida xotirada saqlanadi. Portning ma’lumotlar registrini o‘qishdan oldin amaliy dastur obyektdan yangi ma’lumotlar mavjudligini boshqaruv tizimi registridan so‘raydi va ushbu dalil aniqlangandan keyin shina kontrolleri RD va CS signallarini shakllantiradigan ma’lumotlar registrini o‘qishni amalga oshiradi. Boshqaruv tizimi RD signalini «Ma’lumotlar o‘qilgan» liniyasiga ma’lumotlarning quyidagi portsiyasini qabul qilishda MK tayyorligi to‘g‘risida obyektga xabar bergan holda kommutatsiyalaydi.

Taymer-hisoblagichlar

Taymer-hisoblagichlar vaqtli intervallarni shakllantirish va hodisalarni hisoblash uchun mo‘ljallangan, bu har qanday vaqt funksiyalari asosida ularni amalga oshirish (tegishli dasturiy ta’minotdan

foydalanilganda), shu jumladan real vaqtda boshqarish (ya'ni obyektning vaqt bo'yicha masshtabida) imkonini beradi.

Taymer-hisoblagichning umumlashtirilgan tuzilmasi 4.9.-chizmada keltirilgan.



4.9-chizma. Taymer-hisoblagich tuzilmasi

Taymer-hisoblagich axborotni parallel yuklash va o'qish imkoniyatiga ega sinxron reversiv hisoblagichga asoslanadi. Qoidaga ko'ra, MK hisoblagichining razryadliligi protsessor razryadliligidan oshganligi sababli MK yadrosi va hisoblagichning ma'lumotlar registri qismi o'rtasida axborotni adresli almashinuvidan foydalaniladi.

Ushbu tuzimlmadagi RD va CS signallarni bir vaqtda aktivatsiyalashda MKning ichki shinasida taymer-hisoblagichning chiqish shinasi kommutatsiyalanadi, aks holda, MKning ichki shinasi parallel

yuklamaning kirishiga kommutatsiyalanadi. Taymer-hisoblagichda ma'lumotlarni yuklash WR va CS signallarni bir vaqtda aktivatsiyalash sodir bo'ladi.

Boshqaruv tizimi hisob (boshqaruv tizimi registrining dasturiy qulay biti yoki tashqi signal) yo'nalishlarining topshiriq manbaini, shuningdek taktirlash manbai (yadroning tayanch chastotasi yoki tashqi impulslari) va taktli impulslarni bo'lish koeffitsientini aniqlash imkonini beradi. Nominal (maksimal bo'lib hisoblangan) intervallardan farqlanadigan vaqtli intervallarni berish uchun hisoblagichning dastlabki dasturiy yuklamasidan foydalaniladi.

0 kodiga erishilganda (kamayishdagi hisobda) yoki Nmax (ko'payishdagi hisobda) dasturiy vositalar uchun bo'lgani kabi (taymer-hisoblagichni boshqarish tizimi holatining registrini o'qishda), tashqi apparatura uchun, xususan, obyekt uchun qulay bo'lgan OV to'lish signali generatsiyalanadi.

Shuni ta'kidlash kerakki, vaqtli intervallarga rioya qilish vazifalarining yechimi dasturiy vositalar ayrim yo'riqnomalar bajariladigan t vaqtni, siklda uning N-karrali bajarishni tashkil etish oson emasligini bilgan holda, shu bilan birga T=Nt davomiylik bilan kechikishni ta'minlagan holda qo'llanilishi mumkin, biroq, bunday holatda boshqa dasturiy harakatlarni (masalan, axborotni yig'ish, operator bilan ma'lumotlar almashinuvi va h.) bajarish mumkin emas, bundan tashqari, T kvant qoniqarsiz darajada katta bo'lishi mumkin.

Taymer-hisoblagichlar MKning barcha modellarida amalga oshiriladi.

Hodisalarga ishlov berish bloklari.

Taymer-hisoblagichlar qo'llanilishi bilan erishiladigan vaqt funktsiyalarini amalga oshirish imkoniyatlarini kengaytirish uchun MKning ayrim modellarining periferiya qurilmalar tarkibiga taymer-hisoblagichlar bilan bog'liq bo'lgan vaqtli hodisalarga apparatli ishlov berish bloki kiritilgan. Turli firmalarning MKdagi bunday bloklarning nomi farqlanishi mumkin: EPA (Event Processors Array – hodisalar protsessorlar massivi), HSIO (High Speed Input/Output unit – tezkor kiritish-chiqarish bloki), RSA (Programmable Counters Array – dasturlanadigan hisoblagichlar massivi) va boshqalar.

Bunday bloklarning asosiy vazifalari bo'lib quyidagilar hisoblanadi:

- berilgan ko‘rinish hodisalar (hodisalarni tutib qolish) sodir bo‘lgan vaqtni aniqlash. Tutib qolishni dasturiy amalga oshirishning prinsipial qiyinchiligi taymer ma’lumotlarining ko‘rsatkichini aniq o‘qish imkoniyati mumkin emasligi bo‘lib hisoblanadi (chunki taymer razryadliligi ichki shina razryadliligidan oshadi va registr fragmentlarini ketma-ket o‘qish protsedurasi o‘zgarishlar bilan bajarilishi mumkin);

- vaqtning berilgan momentida berilgan ko‘rinishdagi hodisalarni generatsiyalash (vaqtga nisbatan hodisani bog‘lash). Bog‘lashni dasturiy amalga oshirishning printsipial qiyinchiligi vaqtning berilgan momentini aniqlash momenti (masalan, taymer ma’lumotlarining registrini to‘ldirish bo‘yicha) va berilgan mantiqiy hodisalarni generatsiyalash komandalarini bajarish momenti o‘rtasidagi vaqtli kechikishning mavjudligi bo‘lib hisoblanadi;

- umumiy vaqtli bazadagi keltirilgan ko‘rinishlarning ko‘p kanalliligini ta’minlab turish. Ko‘p kanallilikni dasturiy amalga oshirishning printsipial qiyinchiligi bir nechta dasturiy modullar tomonidan taymer ko‘rsatkichlarini kuzatishning murakkabliligi va samarasizligi bo‘lib hisoblanadi.

Ahamiyatli belgilangan qiyinchilik bo‘lib, dasturiy ishlov berish o‘lchanadigan va generatsiyalanadigan intervallarda ahamiyatli xatoliklar kiritiladigan hodisalar o‘rtasidagi vaqt intervallarida hisoblanadi.

Protsessorlarni o‘zaro almashinuvni ta’minlab turish vositalari

Protsessorlarni o‘zaro almashinuvni ta’minlab turish vositalari taqsimlangan tizimlarini qurish uchun qo‘llaniladi. Bunday tizimlarni qurish uchun ma’lumotlarni ketma-ket uzatishdan foydalaniladi, protsessor o‘rtasida almashinuvni ta’minlab turish vositalari kiritish-chiqarishning ketma-ket portlari deb ataladi. Ma’lumotlarni uzatishning ketma-ket metodini tanlash sababiga uzatish kanalining ishonchliligini ta’minlash zaruriyati, almashinuvning ko‘p bo‘lmagan jadalliligini hisobga olgan holda, kabel xo‘jaligiga sarflanadigan xarajatlarni kamaytirish zaruriyati, shuningdek yagona apparatli amalga oshirish asosida turli protokollar bo‘yicha ma’lumotlar uzatishni ta’minlash zaruriyati kiradi.

Taqsimlangan tizimlardan foydalanishning asosiy sabablariga quyidagilar kiradi:

- boshqaruv obyektining hududiy taqsimlanganligi (bunday holatda lokal boshqaruv vazifasini hal etuvchi va yagona maqsadlarni saqlash uchun ushbu boshqaruv xususiyati to‘g‘risidagi axborot almashinadigan bir nechta MKdan boshqaruv hisoblash tarmog‘i qurilishini hal etish samarali bo‘lib hisoblanadi);

- boshqaruv tizimining ierarxikliligi (boshqaruvning inson-mashina tizimlaridan foydalanish holatida operator qurilmalari to‘g‘risidagi teskari yo‘nalishda esa, datchiklar ko‘rsatkichlari, Mkdagi boshqaruv parametrlari to‘g‘risidagi axborotni operator stansiyasidan MKga (yoki MK tarmog‘iga) uzatish);

- MKning maqsadli funksiyalarini amalga oshirish vaqtida instrumental vositalar (asosan, instrumental EHM)ni qo‘llagan holda, MK negizida tizimni rostlash ehtiyoji (bunday holatda obyekt bilan o‘zaro ishlash jarayonini amalga oshirish uchun band bo‘lgan MK kiritish-chiqarishning shtatli vositalarini blokirovkalamagan holda, MKning resurslaridan foydalana olish imkoniyatini amalga oshirish zarur).

Protsessorlarni o‘zaro almashinuvni ta’minlab turish vositalarining nom ro‘yxati yetarli darajada kengdir. Universal vositalarga UART (U - universal asinxron qabul qilgich-uzatkich), SSI (S - ketma-ket perifieriya interfeysi), va uning turi bo‘lgan SPI (S - kontroller tarmog‘i) kiradi.

Rostlashni ta’minlab turish vositalariga TAR (T - testli foydalana olish uchun port) va OnSE (On C - kristali emulyatori) kiradi.

Blok o‘z ichiga ma’lumotlarni ketma-ket ko‘rinishdan parallel ko‘rinishga o‘zgartirish imkonini beradigan uzatkich va qabul qilgichning siljish registrlarini oladi. Ikkala registrlar R chiqarib tashlash va E ishlashga ruxsat berish kirishiga, uzatkich registri parallel yuklashga ruxsat berish kirishiga ega bo‘ladi. Siljish registrlarining S asinxronlash kirishlari mustaqil hisoblanadi va tashqi sinxrosignaldan bo‘lgani kabi, bo‘lishning dasturlanadigan koeffitsienti bilan Fosc chastotaning hisoblagichi/bo‘lgichidan foydalaniladigan ichki manbadan siljish takti qo‘llanilishi mumkin (qator holatlarda bunday hisoblagich sifatida umumiy vazifadagi taymyerda foydalanish mumkin). Umumiy holatda sinxrosignallar tashqi qurilmaga kelib tushadi.

Uzatkich va qabul qilgich MK dasturi bilan yozish va o‘qish uchun qulay bo‘lgan bufer bilan bog‘liq bo‘ladi. Buferga uning ichidagilarni yozishda uzatkichga qo‘shimcha tarzda ko‘paytiriladi, keyin chiqish liniyasi orqali bir bo‘yicha chiqariladi. To‘plashning tugashi (ovf bayrog‘i) boshqaruv tizimida ro‘yxatga olinadi va dasturiy tarzda aniqlanishi mumkin. Buferni o‘qishga urinishda unga qabul qilgich ichidagilar ko‘paytiriladi, keyin ichki shinada qulay bo‘ladi. (Chaqiruvlarni ishonchli o‘qish uchun qabul qilgichdan ovf bayrog‘ining mavjudligini nazorat qilish zarurdir). Chaqiruv boshlanishi triter bilan aniqlanadi.

UART interfeysida ikkita liniyadan foydalaniladi: RxD qabul qilgich vaTxD uzatkich chiqishi, bu bir dupleksli (bir vaqtdagi ikki tomonlama) asinxron aloqani tashkil etish imkoniga ega. Abonentlar o‘rtasidagi almashinuv faqat beriladigan va oldindan o‘rnatiladigan chastotada olib boriladi. Chaqiruvni uzatishning ishonchliligini oshirish uchun start-bit, 7 yoki 8 axborot razryadlari, nazorat summa bit iva stop-bitni o‘z ichiga oladi. Start-bit («1»dan «0»ga o‘tish) «E»qabul qilgich bitini aktivizatsiyalaydi, keyin chaqiruvni qayd etadi. Amalga oshiriladigan topologiya ko‘p abonentli tizimlarda (boshqaruvchi tarmoqlarida) - yulduzcha.

SSI da har bir stroblangan sinxrosignal bilan uzatiluvchi va qabul qiluvchi datchikdan foydalaniladi. SSI MOSI uzatkichning chiqishi, MOSI qabul qilgich kirishi kabi uchta liniyali SPI, va SCK sinxronlashning umumiy liniyasi qo‘llaniladi, bu yetakchi abonent bilan beriladigan chastota bilan to‘liq dupleksli aloqani tashkil etish imkonini beradi. Yetakchi bo‘lib SCKga beriladigan qurilma, sinxrosignal hisoblanadi. Har bir boshqariladigan qurilma alohida qo‘shimcha liniyalari yordamida tanlanadi. Topologiyasi – «yulduzcha».

I^2C ikkita: SDA (ma’lumotlar liniyasi) va SCK (sinxrosignal liniyalari)dan foydalaniladi, bu yarim dupleksli (teng sinxrosignallar) sinxron aloqani tashkil etish imkoniga ega. Ma’lumotlar uzatkichdan qabul qilgichga SDA bo‘yicha uzatiladi, bunda yetakchi qurilma SCL liniyani boshqaradi. Qurilmalarning almashinishi (uzatkich va qabul qilgich, yetakchi va boshqaruvchi) chaqiruvda uzatiladigan axborotni tahlil qilish asosida dasturiy tarzda amalga oshiriladi.

CAN interfeysida ikkita: CANH va CANLliniyalaridan foydalaniladi. Bu yarim dupleksli asinxron aloqani tashkil etish imkonini beradi. Barcha CAN-shinalari oldindan raqamlangan. Almashinuv seansida raqam katta bo'lgan (CSMACO-AMP – SeRRier - eltuvchini nazorat qilish va xabarlar ustuvori bo'yicha to'qnashishini aniqlash bilan ko'plab foydalana olish) yetakchi qurilma bo'lib hisoblanadi. Qo'llaniladigan tarmoq topologiyasi - «shina».

TAR interfeysida beshta: TMS, NRST, TCK, TDI va TDO liniyalaridan foydalaniladi, oxirgi uchtasi ketma-ket almashinadigan modulga taalluqli bo'ladi va sinxrosignallar liniyalari, ma'lumotlarni kiritish liniyalari va ma'lumotlarni chiqarish liniyalari bo'lib hisoblanadi. Ushbu chiqishlar orqali MKda rostlangan komandalarni sinxronlangan uzatish va MKdan uning ishlashi va uning holati (birinchi ikkitasi – modulni aktivizitsatsiyalashning kirish signali va uni initsializatsiyalashning kirish signali) to'g'risidagi ma'lumotlarni olish sodir bo'ladi. OnCE interfeysida TMS, TCK. TDI va TDO kabi liniyalar bilan vazifasiga ko'ra aynan o'xshash bo'lgan toq: DR, DSCK, DSI va DSO liniyalardan foydalaniladi. Ikkala interfeys sinxron yarim dupleksli aloqani amalga oshiradi va faqat rostlash maqsadlar uchun foydalaniladi, boshqariluvchi tarmoq ular asosida amalga oshirilmaydi.

Uzilishli hodisalarga xizmat ko'rsatish bloklari

Har qanday hisoblash tizimlarining ishlashi (shu jumladan MK asosida), qoidaga ko'ra, ayrim hodisalarga xizmat ko'rsatish nazarda tutiladi. Ularning yuzaga kelish sabablari va joyi turlicha bo'lishi mumkin, biroq ushbu hodisalar bitta umumiy xususiyatga ega bo'ladi, ularning yuzaga kelish vaqti dastur uchun oldindan ma'lum emas. Shu sababli hisoblash tizimida hodisalarni aniqlash dalili bo'yicha ularga xizmat ko'rsatishning ayrim hodisalarining vositalarini nazarda tutish zarur bo'lgan ushbu hodisalarni aniqlash vositalari nazarda tutiladi. (Chunki hodisalarga xizmat ko'rsatishda bajarilayotgan dasturni olib qo'yish («uzish») va xizmat ko'rsatiladigan dasturni bajarish zarur, bunday hodisalar uzilishli hodisalar deb ataladi).

Uzilishli hodisalarni aniqlashning oddiy usuli bo'lib dasturiy usul bilan yuzaga kelish dalilini vaqti-vaqti bilan tekshirish hisoblanadi, bunda ikkita ahamiyatli kamchiliklar bor:

- hisoblash tizimining unumdorligini kamaytirish (dasturning ayrim qismi yuzaga keladigan hodisalar belgisini ortiqcha tanlab olish ostida keltirilgan);

- hodisalar yuzaga kelish vaqti va uni aniqlash vaqti o'rtasidagi kechikish mavjudligi (qator holatlarda, masalan, real vaqt tizimlarida, prinsipial mumkin emas).

Ko'rsatilgan kamchiliklarni oldini olish berilgan uzilishli hodisalarni apparatli aniqlash holatda mumkin, bunda uzilishli hodisalarga xizmat ko'rsatish tizimi yordamida amalga oshiriladi. Uzilishli hodisalarni aniqlashdan tashqari, uzilishli tizimga ulardan ustuvorligini aniqlash uchun hodisalar arbitraji yuklangan.

Uzilishli hodisalar ham dasturiy va apparatli tabiatga ega bo'ladi:

- uzilishli hodisalar yuzaga kelishning dasturiy sabablariga mavjud bo'lmagan komandalarni bajarishga urinish yoki yo'l qo'yilmaydigan ma'lumotlar bilan komandalarni bajarish (masalan, nolga bo'lish holatida) kiradi;

- uzilishli hodisalar yuzaga kelishning apparatli sabablariga periferiya modullarining tayyorligi (taymerning to'lishi, chaqiruvni ketma-ket port bo'yicha qabul qilish va h.) va MKga boshqaruv obyektidan kelib tushadigan signallarni (masalan, bajariluvchi mexanizmning ishlab ketish datchigidan ikkilik signalni) aktivizatsiyalash.

Uzilishli hodisalar miqdori cheklangan (tashqi uzilishli hodisalar miqdori MKning kiritish-chiqarish liniyalarining sonidan oshmasligi kerak), shuning uchun ular apparatli vositalarni aniqlash bo'lishi mumkin. Aks holda, ularga xizmat ko'rsatish bo'yicha harakatlar MKning maqsadli funksiyalari bilan aniqlanadi va yetarlicha turli xil bo'lishi mumkin, qoidaga ko'ra, dasturiy vositalar bilan xizmat ko'rsatiladi. Xotiradan to'g'ri foydalana olish rejimida apparatli harakatlarning oldindan berilgan sxemalar bo'yicha bajariladigan ma'lumotlarni jo'natish bilan bog'liq bo'lgan vaziyatlar (masalan, analog-raqamli o'zgartirishning navbatdagi natijasini ma'lumotlar massivlarining berilgan katakchalariga joylashtirish) bundan mustasno.

Uzilishli hodisalarni aniqlash mohiyati quyidagi harakatlarga to'g'ri keladi:

- uzilishli hodisalar dalilini aniqlash;

- unga ta'sir etish zaruriyatini aniqlash;

- unga muayyan ta'sir zarurligini o'rnatish;

- hodisalar arbitrajini bajarish (avvalgi uzilishli hodisalarga xizmat ko'rsatish vaqtida ushbu hodisalarning muhimlik bosqichiga muvofiq hodisalarning yangi uzilishlari yuzaga kelganda boshlangan xizmat ko'rsatish davom ettirilishi, yoxud yangisiga xizmat ko'rsatish uchun qayta ulangan hodisani to'xtatish kerak);

- dasturiy kontekst tiklanishini bajarish (uzilgan dasturning holatini tiklash va unga teskari qayta ulash).

4.5. AVR- mikrokontrollerining strukturasi

Ishlab chiqariladigan mikrokontrollerlarning razryadiga qarab 4 dan 64 bitgacha bo'ladi. Har xil dasturlarda ishlata olinadigan va arzon narxi bilan ajralib turadigan 8 razryadli mikrokontrollerlar dunyoda eng ko'p tarqalganlar mikrokontrollerdir [15-16]. Atmel firmasining AVR oilasiga mansub mikrokontrllerlar shular sirasidandir.

AVR-mikrokotrollerlari bu – 8 razryadli RISC-mikrokontrollerlaridir. Ajralib turuvchi xususiyatlardan biri FLASh-dastur xotirasiga, ko'p periferik qurilmalariga, yuqori hisoblash unimdorligiga, hamda dasturiy ta'minotini ishlab chiqish uchun kerakli vositalarga egaligidir.

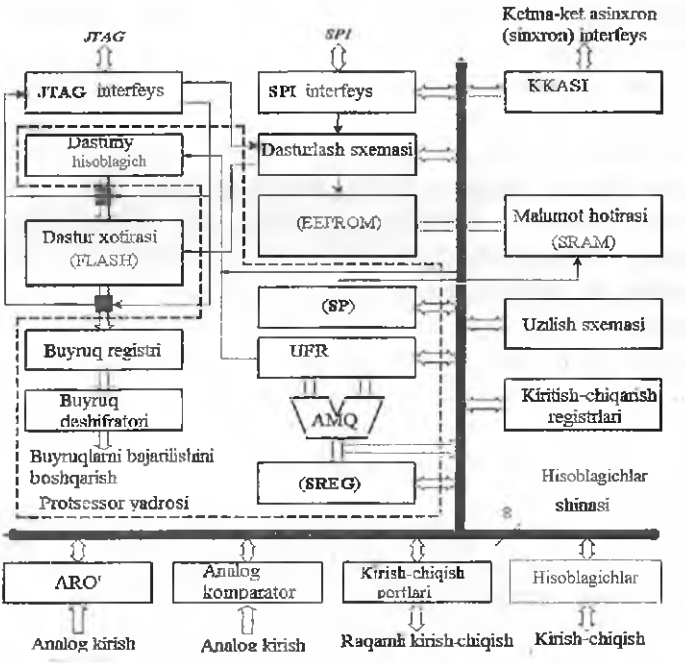
Bugungi kunda AVR oilasini tarkibiga 50 dan oshiq har xil qurilmalar kiradi. Ular bir necha guruhlarga bo'linadi [16].

Universal AVR-mikrokontrollerlari Tiny AVR va Mega AVR guruhlariga kiradi. Tiny AVR (ATtiny*XXX*) – kam sonli tarmoqlanib chiqish joyiga ega, arzon qurilmadir. Mega AVR (ATmega*XXX*) – katta hajmli xotira va tarmoqlanib chiqish joyiga ega hamda periferik qurilmalarning maksimal to'plamiga ega bo'lgan katta quvvatli AVR-mikrokontrolleridir.

Maxsus AVR-mikrokontrollerlari LCD AVR (ATmega169, ATmega329) guruhga kiruvchi - suyuq kristalli indikatorlar bilan ishlash uchun mikrokontrollerlar; USB AVR (AT43USB*XXX*, AT76C711) – USB interfeysli mikrokontrollerlar; DVD AVR (AT78C*XXX*) – CD/DVD-uzatmali kontrollerlar; RF AVR (AT86RF*XXX*) – simsiz aloqa tizimlarini qurish uchun mikrokontrollerlar; Secure AVR (AT90SC*XXXX*, AT97SC*XXXX*) –

smart-karta uchun mikrokontroller; FPGA AVR (AT94K*XX*AL) – bitta kristalda tuzilgan dasturlashtiriladigan mantiqiy integral sxemali AVR-mikrokontrollerlari.

AVR-mikrokontrollerlari kristalida quyidagi apparat vositalariga ega: 8 razryadli protsessor yadrosi, dastur xotirasi, ma'lumotlar operativ xotirasi, energiyaga bog'liq bo'lmagan ma'lumotlar xotirasi, kiritish-chiqarish registri, uzilishlar sxemasi, dasturlash sxemasi, hamda periferik qurilmasi (4.12-chizma).



4.12-chizma. AVR oilasiga mansub mikrokontrollerlarning strukturasi

AVR-mikrokontrollerlarining Protsessor yadrosi (Central Processing Unit – CPU) arifmetik-mantiqiy qurilmaga (AMQ), umumiy maqsadlardagi registriga (UMR), dastur hisoblagichiga, stek ko'rsatkichiga, holat registriga, komanda registriga, komanda deshifratoriga, bajariladigan komandani boshqarish sxemasiga ega.

AMQda hamma hisoblash operatsiyalari bajariladi. Operatsiyalar faqatgina UMR tarkibi ustida olib boriladi. Registr tarkibini tanlash, operatsiyani bajarish va natijalarni qayta UMRga yozish uchun bitta mashinaviy takt (bitta davriy takt chastotasi) ishlatiladi.

Umumiy maqsadlardagi registr – tez joylashadigan 8 razryadli xotira yacheykalaridan, foydalanila olinadigan ALQdan tashkil topadi. AVR-mikrokontrollerlarida UMR registrdan iborat.

Dastur hisoblagichi (Program Counter – PC) keyingi bajariladigan komandani adresini o‘z ichiga oladi.

Stek ko‘rsatkichi (Stack Pointer – SP) yuqori stekning adresini saqlash uchun qo‘llaniladi.

Holat registri (Status Register – SREG) protsessor holatini so‘zini o‘z ichiga oladi.

Komanda registri, komanda deshifratori va bajariladigan komandani boshqarish sxemasi dastur hisoblagichida adresi joylashgan komanda dasturini xotiradan tanlashni, uni dekodlashni (kodni ochishni), komandada ko‘rsatilgan asos (argumentga) kirish yo‘lini aniqlashni va albatta komandani bajarilishini ta’minlab beriladi. Komandani bajarilishini ta’minlash uchun konveyerlash mexanizmi ishlatiladi, ya’ni joriy komandani bajarilish vaqtida, keyingi dastur kodi xotiradan tanlaniladi va dekodlanadi.

AVR-mikrokontrollerlarining xotirasi Garvard turidagi sxema asosida tuzilgan – ya’ni, dastur xotirasini va ma’lumotlar xotirasini adreslari ajratilgan.

Dastur xotirasi dasturlanadigan FLASh tipidagi dasturlanadigan doimiy xotirlovchi qurilmasidan (DXQ) iborat, va u AVR-mikrokontrollerlarining ko‘p komandalari 16 razryadli so‘zlardan iborat bo‘lgani uchun, ketma-ketlik ko‘rinishidagi 16 razryadli yacheykalar ko‘rinishida bajarilgan. 10 000 dan kam bo‘magan sikllarni qayta yozilishi kafolatlanadi. Dastur xotirasi 2 dan 256 Kbaytgacha (1 dan 128 Kso‘z) kattalikka ega.

Ma’lumotlar operativ xotirasi statik operativ xotirlovchi qurilmadan (OXQ) (SRAM – Static Random-Access Memory) tashkil topadi va 8 razryadli yacheykalarni ketma-ketligidan tuzilgan. Ma’lumotlar operativ xotirasi ichki (16 Kbaytgacha) va tashqi (64 Kbaytgacha) bo‘lishi mumkin.

Energiyaga bog'liq bo'lmagan (nonvolatile) ma'lumotlar xotirasi 8 razryadli yacheykalarni ketma-ketligidan tuzilgan va elektrik o'chiriladigan dasturlanadigan DXQ (EEPROM – Electrically Erasable Programmable Read-only Memory) dan tashkil topadi. Energiyaga bog'liq bo'magan ma'lumotlar xotirasi 64 Kbaytgacha kattalikka ega.

Kiritish-chiqarish registri protsessor yadrosini va AVR-mikrokontrollerlarining periferik qurilmalarini boshqarish uchun mo'ljallangan.

Uzilishlar sxemasi ma'lum shartlarda dasturning bajarilishi jarayoni uchun asinxron uzilishlar imkoniyatini yaratadi.

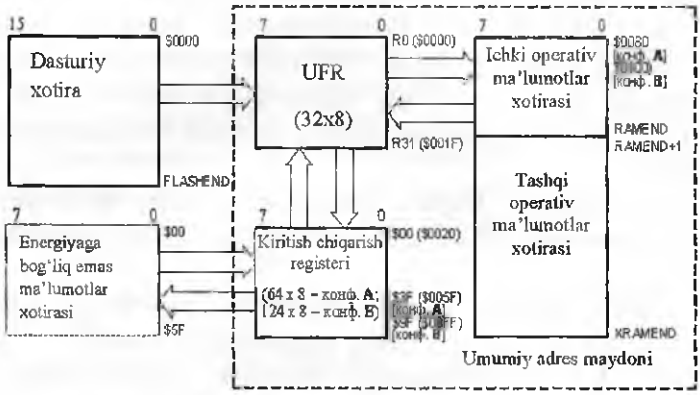
AVR-mikrokontrollerlarning periferik qurilmalariga kiritish-chiqarish portlari, taymer, schyotchiklar, kuzatuvchi taymer, analog komparator, ARO', universal asinxron (sinxron-asinxron) - qabul qilib-uzatgichi, ketma-ketli periferik SPI interfeysi, JTAG interfeysi va boshqalar tegishlidir.

AVR-mikrokontrollerlar o'rtasida ma'lumotlar ayirboshlash ichki 8 razryadli ma'lumotlar shinasi orqali amalga oshiriladi.

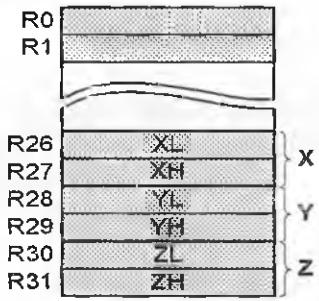
AVR-mikrokontrollerlarning dasturiy modeli. Mikroprotsessorning dastur modeli - dasturga ega manbalarning majmuidir. AVR oilasiga mansub mikrokontrollerlarning dastur modeliga - UMR, kiritish-chiqarish registri, dastur xotirasi, ma'lumotlar operativ xotirasi va energiyaga bog'liq bo'lmagan ma'lumotlar xotirasi kiradi.

Umumiy registrlar (UFR) (R0-R31), dasturda ma'lumotlarni, adreslarni, o'zgarmas (konstant) va boshqa ma'lumotlarni saqlash uchun ishlatiladi. Oltita katta registrlar juft qilib birlashtirilgan va uchta 16 razryadli X [R27:R26], Y [R29:R28] i Z [R31:R30] registrlaridan tashkil topadi (4.13-chizma).

Kiritish-chiqarish registri va ma'lumotlar operativ xotirasi bilan birgalikda bitta adreslar muhitini tashkil qiladi. Adreslar muhiti bu – adresi bilan ajralib turadigan ko'p xotira yacheykalariga ega; xotira katagini identifikatsiyalaydigan adresni son bilan ataladi. Yacheykalar adresi odatda o'noltilik sanoq tizimiga yoziladi, va adresi belgilanayotganda $ belgisi unga ishora beradi.



4.13-chizma. AVR-mikrokontrollerlarining dasturiy modeli



4.14-chizma. Umumiy registrlar

AVR-mikrokontrollerlar xotirasining umumiy adres muhitining ikkita shakli (konfiguratsiyasi) mavjud. A shaklida kichik 32 adres ($0000 – $001F) UMRga tegishli, keyingi 64 adresni ($0020 – $005F) kiritish-chiqarish registri egallaydi, ichki ma'lumotlar operativ xotirasi $0060 adresidan boshlanadi. V shaklida $0060 adresidan boshlab 160 ta qo'shimcha kiritish-chiqarish registri joylashtiriladi, ichki ma'lumotlar operativ xotirasi $0100 adresidan boshlanadi. A shakl, mikroprotsessorlarning eski modellarida va ishlab chiqarishdan olib tashlangan modellar bilan ishlay olish

rejimida ishlaydigan ba'zi bir yangi modellarda ishlatiladi, V shakli – yangi modellarida ishlatiladi.

Dastur xotirasida, o'zining dasturidan tashqari, mikroprotsessor tizimi ishlash jarayonida o'zgarmaydigan doimiy ma'lumotlar yozilishi mumkin. Manba yoqilganidan keyin dasturning bajarilishi, dastur xotirasida $0000 (birinchi yacheyka) adresida joylashgan komandadan boshlanadi.

Energiyaga bog'liq bo'lmagan ma'lumotlar xotirasi mikroprotsessor tizimi ishlash jarayonida o'zgarishi mumkin bo'lgan ma'lumotlarni saqlash uchun mo'ljallangan. Energiyaga bog'liq bo'lmagan ma'lumotlar xotirasi alohida adreslar muhitiga ega va dasturiy yo'l bilan o'qish va yozish (o'zgartirish kiritish) mumkin.

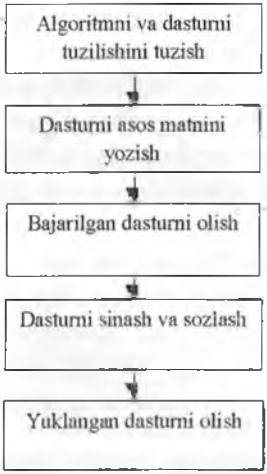
4.6. AVR-mikrokontrollerlarning komandalar tizimi

AVR-mikrokontrollerlarining komandalar tizimiga arifmetik va logik operatsiyalar komandalari (instruktsiyalari), tarmoqlanish komandalari, dasturning ketma-ketlikda bajarilishini boshqarish komandalari, ma'lumotlarni uzatish komandalari va bitlar bilan bajariladigan operatsiyalar komandalari kiradi. Umumiy komandalar tizimiga 130 dan ortiq instruksiyalar kiradi. Mikrokontrollerlarning eski modellarida ba'zi bir komandalarni qo'llab bo'lmaydi.

Mikrokontrollerlarni dasturlash. Bir kristalli mikrokontrollerlar asosida dasturiy ta'minotni tuzish jarayoni quyidagi bosqichlarni o'z ichiga oladi (4.15-chizma) [15-16]:

- algoritmni va dasturni tuzilishini (strukturasini) tuzish;
- dasturni asos matnini yozish;
- bajarilgan dasturni olish;
- dasturni sinash va sozlash;
- yuklangan dasturni olish.

Algoritmni va dasturni tuzilishini (strukturasini) tuzish jarayonida vazifani bajarish metodi tanlaniladi va uni amalga oshirish algoritmi tuziladi. Algoritm bu – qoidalar jamlamasi yoki ma'lum bir vazifani bajarilish operatsiyalarini ketma-ketlikda yozilishidir. Algoritmni grafik ko'rinishi - algoritm sxemasidir (flowchart).



4.15-chizma. Bir kristalli mikrokontrollerlar asosida dasturiy ta'minotni tuzish jarayoni

Dasturni asos matnini yozish jarayonida tuzilgan algoritm dasturning asos tili ko'rinishida yoziladi (assemblerda yoki yuqori darajali tilda). Assembler tili – har bir protsessor komandasiga yoki protsessor komandalarining jamlamasiga qisqartirilgan ramziy yozuv (mnemonika) mos keladigan dastur tilidir. Komandalarni ramziy belgilashni, hamda registr va xotira yacheykalarini adresini, o'zgaruvchilarni, o'zgarmaslarni (konstant) va boshqa dastur elementlarni ishlatish, dasturni tuzish jarayonini osonlashtiradi. Elementlarni ramziy belgilash odatda ularni mazmunini yoritib beradi. Assembler tili, dasturlanayotgan mikroprotsessorning (mikrokontrolleming) hamma resurslariga yo'l ochib beradi, va tezlik jihatdan va xotirada egallanadigan joy bo'yicha effektiv dastur tuzishga imkon beradi. Shuning uchun assembler tilida dasturlash mikroprotsessorning xususiyatini va arxitekturasini bilishni talab etadi. Assembler tillari har xil turdagi mikroprotsessorlar uchun ajraladi. Bir qator assemblerlarda bitta makrokomandalar (makros) singari qaytariladigan ketma-ketlikdagi komandalarni ro'yxatga olishga yo'l qo'yiladi. Bunday assemblerlarni makroassemblerlar deb atashadi.

Yuqori darajali tillar (S, Paskal, Beysik va boshqalar), assembler singari, mikrokontrollerning hamma resurslariga yo'l ochib beradi, lekin bundan tashqari, yaxshi tuzilishga ega dasturlarni tuzishga imkon beradi, dasturchidan xotirani bo'lish tashvishidan holi qiladi va standart operatsiyalarni bajarish uchun ko'p biblioteka funksiyalari to'plamiga ega.

Bajarilgan dasturni olish bosqichida dasturning asosiy matni maxsus vositalar (translyatorlar, kompilyatorlar, joylashtirgichlar va boshqalar) yordami bilan bajariladigan kodga aylanadi. Translyator (translator) bu – assembler tilidagi dasturni, protsessor «tushunadigan» mashinaviy kodga o'girish uchun mo'ljallangan dasturdir. Kompilyator (compiler) bu – yuqori darajali tilda dastur matnini, ekvivalent mashinaviy kodga o'giradigan dasturdir. Obyekt modullaridan tuzilayotgan dasturni qurish uchun joylashtirgich (linker) qo'llaniladi. Dasturni asos matnidan, bajarilayotgan dasturni olish jarayonida, ishlatilayotgan dastur tilida sintaksis qoidasi buzilgan statik xatolar yo'q qilinadi.

Dasturni sinash va sozlash bosqichida izlash, joylashish va unda logik xatolarni bartaraf qilish amalga oshiriladi. Sinash, dasturda xatolarni aniqlash uchun xizmat qiladi va ba'zi bir matnli ma'lumotlarni jamlamasidan foydalanilgan holda bajariladi. Matnli ma'lumotlar algoritmni hamma tarmoqlarini tekshiruv bilan ta'minlamog'i lozim. Dasturni sinoviddan keyin sozlanishi (debug) lozim. Buning vazifasi xatolarni yo'qotishdir, ya'ni dasturda xatolik berayotgan joylarni topishdan iborat.

Yuklangan dasturni olish bosqichida sinash va sozlash uchun ishlatilgan ortiqcha parchalardan (qismlardan) dasturni «bo'shatish» amalga oshiriladi. Bu parchalar dastur hajmini oshiradi va mikroprotsessor tizimini normal ishlashi uchun kerak emas. Keyin olingan yuklash dasturi mikrokontroller xotirasiga kiritiladi.

O'rnatiladigan mikroprotsessorlarni dasturiy ta'minotini (DT) tuzish, maxsus dasturiy va apparat vositalarini ishlatgan holda, shaxsiy kompyuterlarda bajariladi. DT bunday tuzish turi kross-ishlab chiqish deb ataladi. DT ishlab chiqishda va sozlashda qo'llaniladigan apparat va dasturiy vositalarni jamlamasini, bitta nom bilan – ishlab chiqishni qo'llab-quvvatlash vositalari deb atashadi. Dasturni asos matnini tuzish, translyatsiyalash va sozlash, AVR Studioda ishlab chiqilgan

integratsiyalashgan muhitida (Integrated Development Environment – IDE) bajaridadi.

AVR Studio muhitida ishlash. AVR Studio muhiti tarkibiga - asos matnlarini tahrirlagichi, assembler tilidan translyatsiyalagich, sozlagich va simulyator kiradi.

Translyator - belgilar, direktivalar, komandalar va sharhlarga ega, assembler tilidagi asos dasturlari bilan ishlaydi. Belgi - adreslarni ishoraviy belgilanishidir. Belgilar dasturda joyni ko'rsatish uchun ishlatiladi. Direktivalar translyator uchun ko'rsatma vazifasini bajaradi va dasturning bajarilayotgan kodiga kiritilmaydi. Direktivalar bitta yoki bir qancha parametrlarga ega bo'lishi mumkin. Komandalar dasturga bajarilayotgan operatsiyaning mnemonik belgilanish ko'rinishida yoziladi va bitta yoki bir qancha operandlarga ega bo'lishi mumkin. Translyator har xil sanoq tizimlaridagi operandalarga yo'naltira oladi: o'nlik (misol uchun, 15, 154), o'noltilik (prefiks 0x yoki $, misol uchun, 0x0f, $0f, 0x9a, $9a), sakkizlik (prefiks – nol, misol uchun, 017, 0232) va ikkilik (prefiks 0b, misol uchun, 0b00001111, 0b10011010). Dastur satri 120 belgilardan uzun bo'lmasligi shart va to'rttadan bitta formaga ega bo'lishi mumkin:

[belgi:] .direktiva [parametrlar] [;izoh]
[belgi:] komanda [operandalar] [;izoh]
[;izox]
[bo'sh satr]

Kvadrat qavslardagi o'rin shart emas. Nuqta-verguldan keyingi matn izoh hisoblanadi va translyator inkor qiladi. Dastur matniga izohlarning qo'shilishi, dasturlash uslubini yaxshiligidan dalolat beradi va uni kuzatishni osonlashtiradi. Assemblerda dasturlashda bunday qoidalarni qo'llash juda ham muhimdir, chunki assembler tilidagi dasturlar o'qishga noqulaydir.

Dastur translyatsiyalanayotgan mikrokontrollerni turini ko'rsatishda. device direktivasi imkon beradi, misol uchun:

.device AT mega 8535; mikrokontroller uchun dastur AT mega 8535

Translyator uchun assembler tilidagi dasturning matni bilan chiquvchi fayl <fayl_nomi>.asm dir. Translyator to'rtta yangi fayl yaratadi: listing

fayli (<fayl_nomi>.lst), obyektga oid fayl (<fayl_nomi>.obj), dastur xotirasini fayl proshivkasi (<fayl_nomi>.hex), energiyaga bog'liq bo'lmagan ma'lumotlar xotirasi (<fayl_nomi >.eep).

Listing fayli bu – translyatorning ishi haqidagi hisobotdir. 4.16 -chizmada dastur translyatsiyasining listing qismi keltirilgan. Unda 2, 5 va 19 sonlari tegishli R17, R18 va R19 registirlariga kiritiladi; ko'paytma va R17, R18 registrlarining tarkibini summasi hisoblanadi; R17, R18 registrlarining tarkibini summasidan R19 registrining tarkibi ayirib tashlanadi. Listing translyatsiyalanayotgan dasturning asos matnini o'z ichiga oladi. Mashinaviy kodlar va adreslar o'n oltilik sanoq tizimida keltiriladi. Misol uchun, listing satri ADD komandasi bilan quyidagi ma'lumotni o'z ichiga oladi:

0f12 – komandaning mashinaviy kodi;

000004 – berilgan komandani dastur xotirasida joylashgan adresi.

```
000000 e012  LDI R17, 2    ; R17 registriga 2 sonini yuklash
000001 e025  LDI R18, 5    ; R18 registriga 5 sonini yuklash
000002 e133  LDI R19, 19   ; R19 registriga 13 sonini yuklash
000003 9f12  MUL R17, R18  ; R17 ni R18 ga ko'paytirish, natija
R1:R0 da
000004 0f12  ADD R17, R18  ; R17 ni R18 ga qo'shish, natija R17 da
000005 1b31  SUB R19, R17  ; R17 dan R19 ni ayirish, natija R19 da
000006 cfff met: RJMP met; tugallanmaydigan sikl (sozlash uchun)
```

4.16-chizma. Translyatsiya listingiga misol

Obyektga oid fayl maxsus formatga ega va AVR Studio muhiti simulyator-sozlagichi yordamida dasturni sozlash uchun ishlatiladi[14]. Dastur xotirasini proshivka fayli, sozlangan dasturni mikrokontrollerni dastur xotirasiga kiritish uchun xizmat qiladi.

EEPROM-ma'lumotlar xotirasi proshivka fayli, ma'lumotni energiyaga bog'liq bo'lmagan ma'lumotlar xotirasiga yuklash uchun mo'ljallangan. Dastur xotirasini va energiyaga bog'liq bo'lmagan ma'lumotlar xotirasini yuklash operatsiyasi maxsus apparat vositalari (programmatorlar) yordamida bajariladi.

AVR Studio muhitida dasturni tuzish

1. Project menyusidagi New Project komandasidan foydalanilgan holda yangi loyihani yaratish. Dialog (So'zlashuv) oynasida paydo bo'lgan Project Name hoshiyasiga yaratilayotgan loyihani nomini kengayish qismisiz yozish (loyiha haqidagi ma'lumot faylda .aps kengayish nomi bilan saqlanadi). Location hoshiyasida loyiha fayllarining diskdagi joylashuvini ko'rsatish; Project type hoshiyasida AVR Assembler (dasturning asos matni assembler tilida tuziladi) punktini tanlash. Birlamchi dasturni faylini yaratish uchun Create initial File ni belgilash lozim. Dastur fayliga, loyiha nomidan farqlanadigan nomni, Initial File hoshiyasiga kiritish mumkin. Loyiha fayllarini saqlash uchun katalogni yaratishda Create Folder ni belgilash kerak. Har bir loyihani alohida-alohida kataloglarga joylashtirish tavsiya qilinadi. «Next» tugmasini bosish.

2. Select debug platform and device guruhida Debug Platform hoshiyasida yaratilayotgan loyihani sozlash usulini - AVR Simulyator (simulyator-sozlagich) da ko'rsatish, Device hoshiyasida dastur tuzilayotgan mikrokontroller turini (ATmega8535) tanlash. «Finish» tugmasini bosish. Ekranda loyihaning ierarxiya daraxti (Workspace oynasi, zakladka Project) va dasturning asos matnini tahriri oynasi paydo bo'ladi.

Agar loyiha yaratilish jarayonida Create initial file belgilanmagan bo'lsa, dasturning asos matnini faylini File menyusida joylashgan New File komandasi orqali yaratish mumkin.

3. Dastur matnini kiritish va tahrirlash. File menyusida joylashgan Save komandasidan foydalanilgan holda, faylni saqlash. Agar dasturning asos matni fayli bor bo'sa, uni Project menyusida joylashgan Add existing File komandasi orqali loyihaga qo'shib qo'yish mumkin.

4. Yaratilgan dasturni Project menyusidagi Build and run komandasi (yoki klaviaturadagi Ctrl+F7 klavishlarini kombinatsiyasi) orqali translyatsiyadan o'tkazish. Translyatsiyadan oldin AVR

Assembler dialog (so'zlashuv) oynasidagi List file belgilanganligiga ishonch hosil qiling (bu translyatsiya listingini yaratish uchun kerak bo'ladi). AVR Assembler dialog (so'zlashuv) oynasi Project menyusidagi AVR Assembler Setup komandasi orqali chaqirish amalga oshiriladi. Translyatsiya tugashi bilan Output oynasida Build zakladkasida translyatsiya natijalari haqida ma'lumot paydo bo'ladi. Listing faylini, loyihaning ierarxiya daraxtidan ochish mumkin (Workspace oynasi, zakladka Project). Undan keyin Project menyusidagi Save Project komandasi yordami bilan o'zgarishlarni loyiha fayliga saqlash.

4.7. Mikrokontrollerlarni dasturiy ta'minotini sozlash vositalari

O'rnatiladigan MP (hamda bir kristalli mikrokontrollerlar) asosidagi qurilmalarning DT ni sozlashni asosiy xususiyati, rivojlangan vositalarning tarkibida yo'qligi hisoblanadi. Ayniqsa o'rnatiladigan mikroprotsessor tizimlari uchun sozlash bosqichi ayni mas'ulyatlidir.

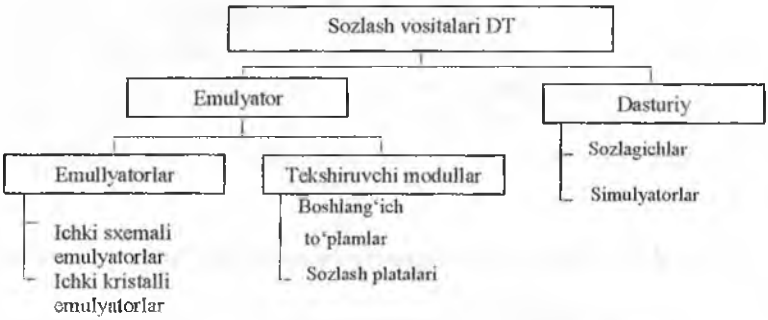
Mikroprotsessorlarning (mikrokontrollerlarning) datchiklar va bajaruvchi qurilmalar bilan o'zaro bog'liqligi periferik qurilmalarning registri (kiritish-chiqarish registri) orqali ma'lumotlarni uzatish yo'li bilan amalga oshadi. Bunday registrlarning alohida joylashgan razryadi periferik qurilmalarning ishlash tartibini, ma'lumotlarni uzatishni tugatishni va shunga o'xshash rejimlarni kiritadi. Bu razryadlarni holati dasturiy o'rnatilishi mumkin. DT sozlashda tez-tez registrlararo uzatish darajasiga o'tib turishga va alohida joylashgan razryadlarni to'g'ri o'rnatilganligini tekshirib turishga to'g'ri keladi. Bundan tashqari, sozlash bosqichida algoritmni optimallashtirish, kodlarni qiyin joylarini topish va ishlab chiqilgan DT ishonchliligini tekshirish mumkin.

Ko'rsatilgan vazifalarni yechish uchun DT sozlash uchun appartli va dasturiy vositalar qo'llaniladi (4.17-chizma).

Sozlashning apparat vositalariga, apparat emulyatori va tekshirish modullari kiradi.

Apparat emulyatori real vaqt rejimida mikroprotsessor tizimlarini dasturiy va apparatli sozlash uchun mo'ljallangan. Ular maxsus DT - sozlash-dasturlari bilan ta'minlangan «boshqaruvchi» kompyuter boshqaruvida ishlaydi. Apparat emulyatorlarining asosiy turlari:

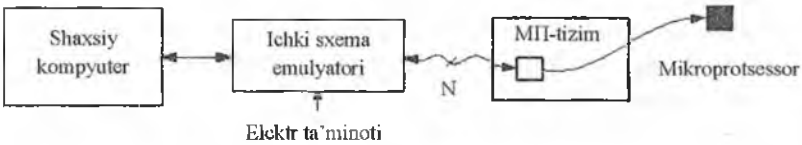
- ichki sxemali emulyatorlar yoki, sozlash tizimida mikroprotsessorlarni o'rnini bosadigan ulanadigan (qo'shiladigan) - emulyatorlar;



4.17-chizma. Dasturiy ta'minotni sozlash vositalarini klassifikatsiyasi

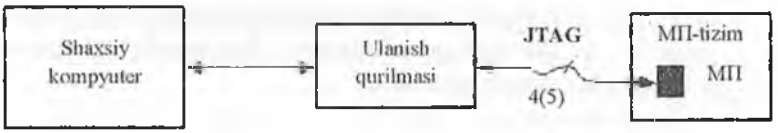
- mikroprotsessorni bitta ichki qurilmalaridan bo'lgan – ichki kristalli emulyatorlar.

Ichki sxemali emulyator (In-Circuit Emulator, ICE) bu – protsessorning apparatli o'xshatuvchiga (imitator) va o'xshatuvchini boshqarish sxemasiga ega qurilmadir. Emulyator yordami bilan sozlalayotganda mikroprotsessor sozlanayotgan tizimdan chiqarilib olinadi, uni o'rniga kontaktga oid kolodga ulanadi (4.18.-chizma). Egiluvchan kabel yordami bilan kontaktga oid kolodga emulyatorga ulanadi. Sozlash jarayonini boshqarish shaxsiy kompyuter orqali amalga oshiriladi. Ulanadigan(qo'shiladigan)-emulyatorlar quyidagi kamchiliklarga ega: qimmat baholigi, kerakli darajada ishonchli emasligi, yuqori energiya iste'moli, emulyator ulangan zanjirlarning elektrik xarakteristikalariga ta'siri.



4.18-chizma. Ichki sxemali emulyator yordami bilan sozlash

Ichki kristalli emulyator (On-Chip Emulator) mikroprotsessorni tizimdan olmasdan turib sozlashga imkon beradi. Bundan tashqari to‘g‘ridan-to‘g‘ri dasturni boshqarish mumkin. Ichki kristalli sozlashni eng ko‘p tarqalgan vositasi - JTAG (Joint Test Action Group) nomi bilan taniqli, IEEE 1149,1 ketma-ketlik interfeysidir. JTAG ketma-ketlik sozlash porti, maxsus bog‘lanish qurilmalari yordamida kompyuterga ulanadi. Bunda protsessorning sozlash vositalariga ulanishga imkon yaratiladi (4.19-chizma). Bunday sozlash usulini skanlovchi emulyatsiya deb ham atashadi. Bu uslubning ustunligi shundaki, protsessorni tizimdan chiqarmasdan turib har xil ishlarni bajarilish mumkin, tizimning elektrik xarakteristikalarini o‘zgartirmagan holda protsessorni maksimal darajada unumdor ishlashini ko‘tara olish imkoniyati yaratiladi [14].



4.19-chizma. Ichki kristalli emulyator yordamida sozlash

Tekshiruvchi modullar real vaqt mobaynida dasturiy ta’minotni tez sozlash uchun mo‘ljallangan. Tekshiruvchi modullar 2 xil turda bo‘ladi: boshlang‘ich to‘plamlar va sozlash platasi.

Boshlang‘ich to‘plamlar (Starter Kit) aniq bir mikroprotsessor bilan ishlashni o‘rgatish uchun mo‘ljallangan. Boshlang‘ich to‘plam mikroprotsessorni xarakteristikalarini o‘rganishga, uncha qiyin bo‘lmagan dasturlarni sozlashga, qiyin bo‘lmagan maketlashni bajarishga, aniq bir vazifani bajarish uchun mikroprotsessorni qo‘llash imkoniyatini tekshirishga imkon beradi. Boshlang‘ich to‘plam tarkibiga plata, DT va hujjatlar to‘plami kiradi. Plataga mikroprotsessor, dasturni yuklaydigan qurilma, ketma-ket yoki parallel portlar, ichki qurilmalar bilan aloqa uchun ajratgichlar va boshqa elementlar o‘rnatiladi. Plata kompyuterga parallel yoki ketma-ketli portlar orqali ulanadi. Boshlang‘ich to‘plamlar mikroprotsessor bilan ishlashni boshlang‘ich bosqichida juda ham qulaydirlar.

Sozlash platalari (Evaluation Board) tuzilgan algoritmni real sharoitda tekshirish uchun mo'ljallangan. Ular algoritmni sozlash va optimallashtirish imkonini beradi. Odatda platada mikroprotsessor, sinxronizatsiya sxemasi, xotira va periferiya kengayish interfeyslari, elektrmanba sxemasi va boshqa qurilmalar joylashadi. Plata kompyuterga parallel yoki ketma-ketli portlar orqali ulanadi, yoki to'g'ridan-to'g'ri PCI slotiga o'rnatiladi.

Sozlashning asosiy dasturiy vositalari simulyator va sozlagichlardir.

Simulyatorlar (simulator) yoki komandalar tizimini simulyatorlari, protsessorni komandalar darajasida ishini taqlid (imitatsiya) qiladigan dasturdir.

Odatda simulyatorlarni dasturni yoki uni alohida bir qismlarini apparat vositalarida sinashdan oldin tekshirishda qo'llaniladi.

Sozlagichlar (debugger), yaratilgan dasturiy ta'minotni ishlashini tahlil qilishda qo'llaniladigan dasturdir. Sozlagichlarni quyidagi imkoniyatlarini ko'rsatish mumkin.

1. Qadamma-qadam bajarilish. Dastur, komandalar ketma-ketligida, har bir qadamdan keyin boshqaruv sozlagichga qaytish yo'li bilan bajariladi.

2. Haydash. Dasturni bajarilishi ko'rsatilgan komandadan boshlanadi va dastur oxirigacha to'xtovsiz bajariladi.

3. Nazorat nuqtasidan haydash. Dasturni bajarilishi paytida to'xtash sodir bo'ladi va adreslar bilan ishlash komandasi bajarilgandan keyin, ro'yxatda ko'rsatilgan nazorat nuqtalari bilan boshqaruvni sozlagichga beriladi.

4. Registrlar tarkibini va xotira yacheykalarini ko'rish va o'zgartirish. Foydalanuvchi registr tarkibini va xotira yacheykalarini ekranga chiqarish va o'zgartirish imkoniyatiga ega bo'ladi.

O'rnatiladigan mikroprotsessorlarning DT sozlagichlari odatda ichki sxemali va ichki kristalli emulyatorlar bilan birga ishlatiladi, hamda simulyator rejimida ishlashi mumkin. Ba'zi bir sozlagichlar profillashga, ya'ni dasturni ma'lum bir joyini bajarilganini aniq vaqtini aniqlashga imkon beradi. Ba'zida profillash funksiyasini maxsus dastur – profillagich (profiler) amalga oshiradi.

AVR-mikrokontrollerlarining DT sozlash vositalari. AVR-mikrokontrollerlarining dasturiy ta'minotini sozlashni apparat vositalari

ICE50 ichki sxemali emulyatorida JTAG ICE ichki kristalli emulyatorda, hamda STK500 boshlang'ich to'plamida ko'rsatilgan.

AVR Studio muhitining sozlagichi, ICE50 ichki sxemali emulyatorida JTAG ICE ichki kristalli emulyatorda, STK500 boshlang'ich to'plami yoki simulyator bilan qo'llanishi mumkin. Ko'rsatilgan sozlash uslubi loyiha yaratilishi uchun mo'ljallangan. AVR Studio muhitining simulyatori apparat vositalarini qo'llamasdan dasturni dastlabki sozlash uchun mo'ljallangan. AVR Studio muhitida DT sozlash. AVR Studio dasturining sozlash komandalari Debug menyusida joylashgan.

AVR Studio muhitida sozlash rejimiga o'tish Debug menyusidagi Build and Run yoki Start Debugging komandalarini ishlatganda avtomat ravishda ishga tushadi. Sozlash rejimidan chiqish Debug menyusidagi Stop Debugging komandasi orqali amalga oshadi.

Dasturni qadamma-qadam bajarilishi Debug menyusidagi Step Into, Step Over komandalari orqali kiritiladi. Step Into, dasturni bitta komandasini bajarish imkonini beradi (dastur osti dasturni chaqirish komandasini ham).

Dastur osti dasturni bajarilishini tugatish uchun Step Out komandasi ishlatilishi mumkin. Step Over komandasi ham dasturni bitta komandasini bajaradi, lekin agar bu dastur osti dasturni chaqirish komandasi bo'sa, bu komanda bitta qadamni umumiy bajaradi. Keyingi bajariladigan komanda (adresi dasturning schyotchigida joylashgan komanda) dasturning asos matni oynasida ⇨ belgisi bilan belgilanadi. Dasturning bajarilishini uzish Reset komandasi orqali amalga oshiriladi.

Dasturni haydash (bajarishni boshlash va davom ettirish) Run komandasi bilan amalga oshiriladi. Dasturni bajarilishini to'xtatish uchun Break komandasi qo'llaniladi.

Nazorat nuqtalari maxsus markerlardan tashkil topadi va uch xil bo'lishi mumkin: to'xtash nuqtasi, trassirovka nuqtasi va kuzatish nuqtasi.

To'xtash nuqtasi Debug menyusidagi Toggle Breakpoint komandasi yoki dasturning asos matni tahriri menyusidan kiritiladi. Asos matn tahririda to'xtash nuqtasi, ● belgisi bilan belgilanadi. Kiritilgan to'xtash nuqtalarini Output oynasidagi Breakpoints zakladkasidan

ko‘rish mumkin. Dasturni xuddi o‘sha satrida yana toxtash nuqtasini o‘rnitish komandasini qaytadan chaqirish, to‘xtash nuqtasini olinishiga olib keladi. Hamma kiritilgan to‘xtash nuqtalarini o‘chirish Debug menyusidagi Remove Breakpoints komandasi yoki Output oynasidagi Breakpoints tuguni menyusidagi Remove all Breakpoints komandasi orqali amalga oshiriladi.

To‘xtash nuqtasining parametrlari dasturning asos matn tahriri menyusidagi Breakpoints Properties komandasi orqali chaqiriladigan Breakpoint Condition dialog oynasidan kiritiladi. Iterations ni belgilash komandani qayta bajarishlar sonini kiritishga imkon beradi. Watchpoint belgilanganda to‘xtash nuqtasigacha yetib kelinishi bilan, faqatgina registrlarni qiymati va ko‘rish oynalaridagi xotira yacheykalari yangilanishi ishlab chiqiladi. Iterations va Watchpoint larni bir vaqtda belgilash mumkin emas. Show message – to‘xtash nuqtasiga yetib kelganligi haqidagi xabarni ko‘rsatishi uchun belgilanadi. Xususiyatlarni kiritish va to‘xtash nuqtalarini o‘chirib tashlash uchun dialog oynasini chaqirish, Output oynasidagi Breakpoints menyusi tugunida amalga oshiriladi.

Trassirovka nuqtasi real vaqt rejimida dasturning bajarilishini boshqarish uchun mo‘ljallangan. Trassirovka o‘zgacha nomga ega - dastur trassasini kuzatishga imkon beradi. AVR Studio muhitida trassirovka funksiyasi faqat ichki sxemali emulyatorni qo‘llab, dasturni sozlashda ishlatilishi mumkin. Simulyator rejimi ishlayotganda trasirovka funksiyasini ishlatib bo‘lmaydi.

Kuzatish nuqtasi dasturning asos matni tahriri menyusidagi Add to Watch komandasi orqali kiritiladi. Kuzatish nuqtasi, tarkibini kuzatish kerak bo‘lgan registr yoki xotira yacheykalarining ramziy nomlarini tashkil qiladi. Add Watch komandasi bajarilayotganda ekranda to‘rtta ustunga bo‘lingan Watches oynasi paydo bo‘ladi: Name (kuzatish nuqtasini ramziy nomi), Value (qiymat), Type (tur), Location (joylashuv).

AVR Studio muhitining sozlagichi yana funksiyalar bilan ta’minlaydi: kursorgacha bajarish (Debug menyusidagi Run to Cursor komanda) va komandalarni ketma-ketlikda bajarish, to‘xtash bilan (Debug menyusidagi Auto Step komandasi).

Registrlarni va xotira yacheykalarni tarkibini ko‘rish va o‘zgartirish uchun View menyusidagi Registers, Memory, Memory 1, Memory 2, Memory 3 komandalari xizmat qiladi.

Protsessorni holatini kuzatish uchun Workspace oynasi, I/O zakladkasidagi Processor obyektini ochish lozim. Shunda quyidagi ma’lumot ko‘rsatiladi: dastur hisoblagichini tarkibi (Program Counter); stek ko‘rsatgichini tarkibi (Stack Pointer); bajarilishni boshidan beri o‘tgan taktlar soni (Cycle Counter); 16 razryadli X, Y va Z ko‘rsatgich-registrlari tarkibi; takt chastotasi (Frequency); bajarish uchun ketgan vaqt (Stop Watch).

Kiritish-chiqarish registrlarini tarkibini nazorat qilish uchun Workspace oynasi I/O zakladkasidagi I/O* obyektini ochish kerak (* - mikrokontroller turi). I/O obyektiga kiradigan kiritish-chiqarish registrlari, periferik qurilmalar turi bo‘yicha guruhlashtirilgan.

Registr va xotira yacheykalari tarkibini o‘zgartirilgan (modifikatsiyalangan) qiymati, faqatgina sozlashning joriy seansi vaqtida ishlaydi, dasturning asos matniga o‘zgartirishlar kiritilmaydi.

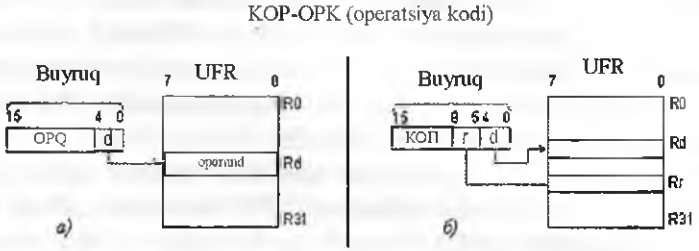
4.8. Operandlarni adreslashlash usullari

Qo‘llanilayotgan operandlarni soniga qarab AVR-mikrokontrollerlar komandalarining 3 turi bo‘ladi: adressiz, bir adresli va ikki adresli. Birinchi turida komandalarda faqatgina komanda tomonidan bajarilayotgan funksiyani aniqlaydigan - operatsiyalar kodi (KOP) bor. Ikkinchi va Uchinchi turdagi komandalarda, operatsiyalar kodidan tashqari adreslar qismiga ham ega. Operandaning adresini shakllantirish usulini adresatsiya (addressing) deb ataladi. Adreslash usuli yordamida fizik adres hisoblanadi.

Xotirani adreslash turiga qarab, AVR-mikrokontrollerda adreslash usullarini - UMR va kiritish-chiqarish registrlarini adreslash usuliga, OXQ adreslash usuliga va dastur xotirasini adreslash usuliga ajratish mumkin. Har xil turdagi adreslash usullarini qo‘llanilishi, dasturning hajmini va bajarilishi uchun ketadigan vaqtini qisqartirish imkonini beradi.

UMR va kiritish-chiqarish registrlarini adreslash uchun faqat bitta rejim ko‘zda tutilgan - to‘g‘ridan-to‘g‘ri registrli adreslash.

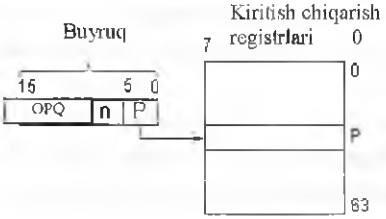
UMRni to‘g‘ridan-to‘g‘ri registrli adreslashda, operand - komandada ko‘rsatilgan, UMR tarkibi hisoblanadi. To‘g‘ridan-to‘g‘ri registrli adreslash komandalari bitta (Rd) yoki ikkita (Rr va Rd) UMR adreslashligi mumkin (4.20-chizma).



4.20-chizma. To‘g‘ridan-to‘g‘ri registrli adreslash

Ikkinchi holatda komandali bajarilishi natijalari Rd registrida saqlanadi. UMR to‘g‘ridan-to‘g‘ri registrli adreslash hamma arifmetik va logik komandalarda, hamda bitlar bilan ishlaydigan ba’zi bir komandalarda qo‘llaniladi. Chunki bu komandalar ALQda faqat UMR tarkibi ustida bajariladi. Ikkinchi operandasi konstanta (o‘zgarmas) bo‘lgan komandalarda, birinchi operandasida faqat UMRning katta qismidagi registrlaridan (R16 – R31) ishlatilishi mumkin.

Kiritish-chiqarish registrini to‘g‘ridan-to‘g‘ri registrli adreslashda, operand – komandada ko‘rsatilgan kiritish-chiqarish registirini tarkibida bo‘ladi. Kiritish-chiqarish registirini adresi komandaning 6ta razryadida saqlanadi (4.21-chizma).



4.21-chizma. Kiritish-chiqarish registrini to‘g‘ridan-to‘g‘ri registrli adreslash

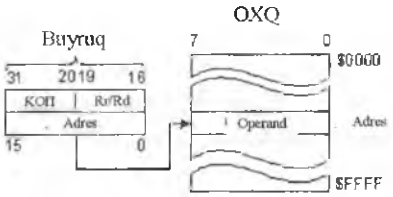
Kiritish-chiqarish registrini to‘g‘ridan-to‘g‘ri registrli adreslash, kiritish-chiqarish registirining IN o‘qish va OUT yozish komandalarida, hamda kiritish-chiqarish registri bilan ishlovchi bir qator boshqa komandalarda ishlatiladi.

To‘g‘ridan-to‘g‘ri registrli adreslashni qo‘llanilishiga misol:

```
; bitta UMRni to‘g‘ridan-to‘g‘ri registrli adreslash
CLR  R1            ; R1 registrini hamma razryadlarini tozalash
; ikkita UMRni to‘g‘ridan-to‘g‘ri registrli adreslash
ADD R11, R12     ; R11 va R12 registrlarini tarkibini qo‘shish
```

Ma’lumotlar operativ xotirasini adreslash uchun 5ta adreslash usuli qo‘llaniladi: bevosita, qisman, qisman siljish, qisman predekrementli va qisman postinkrementli.

1. Ma’lumotlar operativ xotirasini bevosita adreslashda, adresi komandada ko‘rsatilgan, OXQ yacheykasining tarkibi, operand hisoblanadi. Operandning adresi 32 razryadli komandaning 16 ta kichik razryadida joylashgan bo‘ladi (4.22.-chizma).



4. 22-chizma. Ma’lumotlar operativ xotirasini bevosita adreslash

Bevosita adreslash, LDS (Load Direct from Data Space) komandasida va STS (Store Direct to Data Space) komandasida ishlatiladi. OXQ da baytlarni. byte direktivasi rezervga oladi. .byte direktivasi bitta parametrga ega – ajratiladigan baytlar soni – va faqatgina. dseg (data segment) direktivasi yordamida aniqlanadigan ma’lumotlar segmentida ishlatilishi mumkin. Dasturning boshlanish segmenti .cseg (code segment) direktivasi yordami bilan ko‘rsatiladi. .dseg i .cseg direktivalari parametrlarga ega emas.

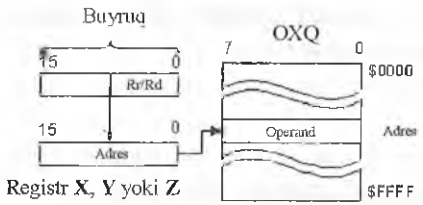
Bevosita adreslashni qo‘llanilishiga misol:

```
; bevosita adreslash
```

```
.dseg    ; ma'lumotlar segmenti (ma'lumotlar operativ xotirasi)
.org $0065          ; $0065 adresi bo'yicha
cnt1:  .byte 1      ;  1 ta baytni cnt1 uchun rezervlash

.cseg               ; dastur segmenti (dastur xotirasi)
;...
LDS  R10, cnt1   ; cnt1 ni R10 ga yuklash
```

2. Ma'lumotlar operativ xotirasini qisman adreslashda, adresi X, Y yoki Z registrlarida joylashgan (4.23.-chizma), OXQ yacheykasining tarkibi, operand hisoblanadi.



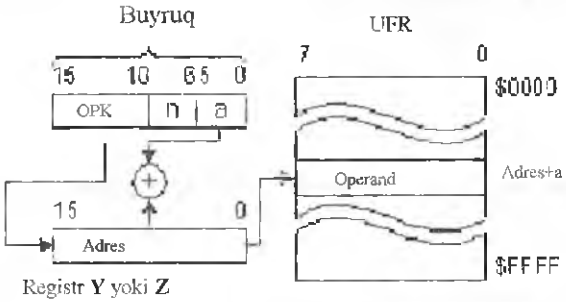
4.23-chizma. Ma'lumotlar operativ xotirasini qisman adreslash

Qisman adreslash LD (Load Indirect) komandasida va ST (Store Indirect) komandasida ishlatiladi. Misol uchun:

```
; qisman adreslash
.dseg  ; ma'lumotlar segmenti (ma'lumotlar operativ xotirasi)
cnt1:  .byte 1      ;  1 ta baytni cnt1 uchun rezervlash
.cseg               ; dastur segmenti (dastur xotirasi);...
LDI  R30, low(cnt1)  ; R30 ga cnt1 adresining kichik baytini
yuklash
LDI  R31, high(cnt1); R31 ga cnt1 adresining katta baytini
yuklash
LD   R1, Z          ; R1 ga cnt1 ni yuklash, ya'ni R1 <- (R31:R30)
```

3. Ma'lumotlar operativ xotirasini qisman siljishli adreslashda, komandada ko'rsatilgan ma'lumotlar operativ xotirasidagi operanda adresi, Y yoki Z registrlarini tarkibiga siljishni qo'shilish yo'li bilan

hisoblanadi. Siljish komandaning 6 ta razryadida joylashadi (4.24-chizma).

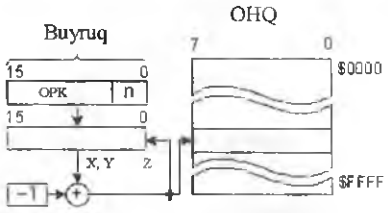


4.24-chizma. Ma'lumotlar operativ xotirasini qisman siljishli adreslash

Qisman siljishli adresatsiya LDD (Load Indirect with Displacement) komandasida va STD (Store Indirect with Displacement) komandasida qo'llaniladi. Misol uchun:

```
; qisman siljishli adreslash
.dseg    ; ma'lumotlar segmenti (ma'lumotlar operativ xotirasi)
arr:  .byte 5      ; arr massivi uchun 5 ta baytni rezervlash
.cseg              ; dastur segmenti (dastur xotirasi)
LDI  R30, low(arr)  ; R30 ga arr adresining kichik baytini yuklash
LDI  R31, high(arr) ; R31 ga arr adresining katta baytini yuklash
;...
LDD  R10, Z+0  ; arr massivini birinchi elementini R10 ga yuklash
LDD  R11, Z+1  ; arr massivini ikkinchi elementini R11 ga yuklash
```

4. Ma'lumotlar operativ xotirasini qisman predekrementli adreslashda (lot. decrementum – kamayish) tarkibi X, Y yoki Z registr komandalarida ko'rsatilgan komandani bajarilishidan oldin dekrementatsiyalanadi (bittaga kamayadi); dekrementatsiyalangan X, Y yoki Z registrining tarkibi, ma'lumotlar operativ xotirasini operand adresi hisoblanadi (4.25-chizma).

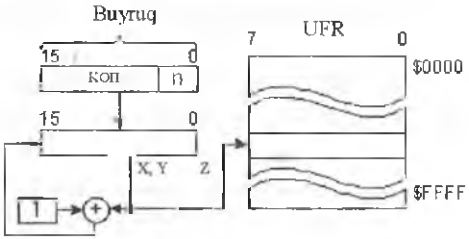


4.25-chizma. Ma'lumotlar operativ xotirasini qisman predekrementli adreslash

Qisman predekrementli adreslash, LD komandasida va ST komandasida qo'llaniladi. Misol uchun:

```
; qisman predekrementli adreslash
LD   R10, -Z          ; Z <- Z - 1, R10 <- (Z)
```

5. Ma'lumotlar operativ xotirasini qisman postinkrementli adreslashda (lot. incrementum – ko'payish, o'sish) X, Y yoki Z registrining tarkibi ma'lumotlar operativ xotirasini operand adresi hisoblanadi; X, Y yoki Z registr tarkibidagi komandalar bajarilganidan keyin inkrementatsiyalanadi, ya'ni bittaga ko'payadi (4.26.- chizma).

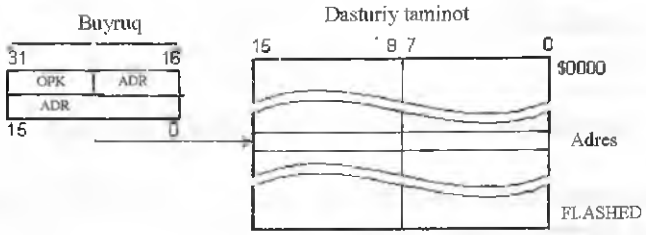


4.26-chizma. Ma'lumotlar operativ xotirasini qisman postinkrementli adreslash

Qisman postinkrementli adreslash, LD komandasida va ST komandasida qo'llaniladi. Misol uchun:

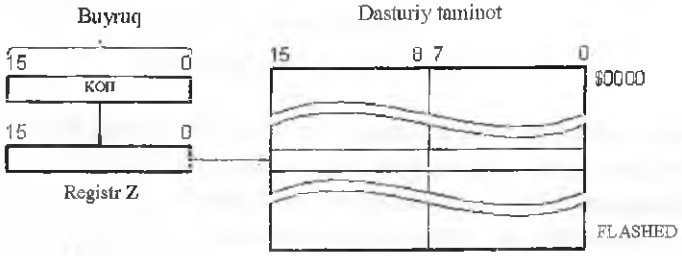
```
; qisman postinkrementli adreslash
LD   R10, Z+          ; R10 <- (Z), Z <- Z + 1
```

Dastur xotirasini bevosita adreslashda dasturni bajarilishi, komandada ko'rsatilgan adresdan davom ettiriladi (4.27- chizma). Dastur xotirasini bevosita adreslash JMP va CALL komandalarida qo'llaniladi.



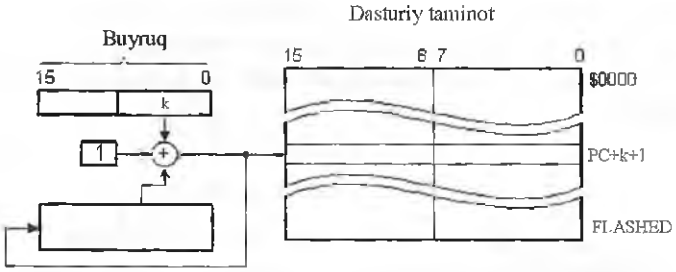
4.27-chizma. Dastur xotirasini bevosita adreslash

Dastur xotirasini qisman adreslashda dasturni bajarilishi, Z registri tarkibidagi adresdan davom ettiriladi, ya'ni dastur hisoblagichiga Z registri tarkibidagi yuklanadi (4.28.- chizma). Dastur xotirasini qisman adresatsiyalash IJMP va ICALL komandalarida qo'llaniladi.



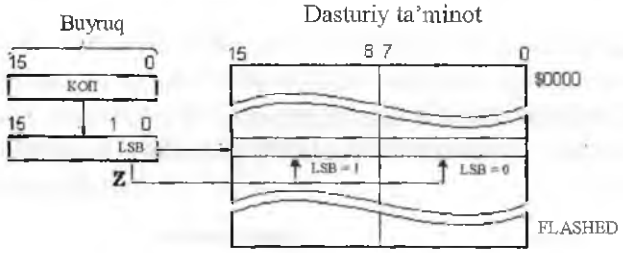
4.28-chizma. Dastur xotirasini qisman adreslash

Dastur xotirasini nisbatan adreslashda dasturni bajarilishi, (PC + k + 1) adresidan davom ettiriladi (4.29.-chizma). Dastur xotirasini nisbatan adresatsiyalash RJMP i RCALL komandalarida qo'llaniladi.



4.29-chizma. Dastur xotirasini nisbatan adreslash

Konstantani adreslashda konstantaning bayt adresi, Z registrida joylashadi (4.30-chizma). Dastur xotirasida konstantani adresatsiyalash LPM (Load Program Memory) komandasida qo'llaniladi.



4.30-chizma. Konstantani adreslash

Dastur xotirasiga ma'lumotlarni kiritish. db (define bytes) direktivasi imkon beradi. Kiritilgan xotira yacheykalariga yo'l ko'rsata olish uchun, direktivadan oldin belgi bo'lishi zarur. db direktivasi dastur segmentida joylashadi va .org direktivasi bilan qo'llanilishi mumkin.

Dastur xotirasida konstantani adreslashga misol:

```
LDI   R31, high(var<<1)   ; Z registrining katta bayti
LDI   R30, low(var<<1)    ; Z registrining kichik bayti
LPM R16, Z                ; $50 sonini R16 ga yuklash
;...
.org $0025                ; $0025 adres bo'yicha
var:   .db $50, 137       ; $50 va 137 konstantalar
```

Yuqorida keltirilgan misolda operandani chapga, ko‘rsatilgan razryadlar soniga siljitadigan, << operator ishlatilgan.

Dastur xotirasida postinkrementli konstantani adresatsiyalashda konstantaning bayt adresi, Z registrida joylashadi. Z registridan jo‘natilgan bayt ko‘rsatilgan registrga yuklanadi, Z registri tarkibidagi komanda bajarilganidan keyin inkrementatsiyalanadi. Dastur xotirasida postinkrementli konstantani adresatsiyalash LPM va ELPM komandalarida qo‘llaniladi. Misol uchun:

```
LDI   R31, high(var<<1); Z registrining katta bayti
LDI   R30, low(var<<1) ; Z registrining kichik bayti
LPM   R16, Z+          ; $50 sonini R16 ga yuklash, inkrement Z
LPM   R17, Z           ; 137 sonini R17 ga yuklash
;...
var:  .db $50, 137     ; $50 va 137 konstantalar
```

4.9. Mikrokontrollerlarni dasturlash

4.9.1. Siklik dasturlash tuzish

Har qanday boshqarish yoki ma’lumotlarni qayta ishlash jarayoni ba’zi bir algoritmik tuzilishlarni yig‘indisini tashkil qiladi. Algoritmik tuzilishlarning eng ko‘p tarqalgan turlari - tarmoqlanish (branching) va sikllar (loop) hisoblanadi [15-16].

Tarmoqlanish dasturning har xil qismlarini tuzish (algoritm tarmoqlarini ajratish) uchun qo‘llaniladi.

TSikllarda bitta operatsiya xotira yacheykasida yoki elementlarida ketma-ketlikda joylashgan bir qancha ma’lumotlar tarkibi ustidan bajariladi. Massivlarni va jadvallarni qayta ishlashda siklli dasturlarni qo‘llash maqsadga muvofiqdir.

Siklli dasturda 4 ta asosiy blokni ajratish mumkin.

1. Initsializatsiya bloki (lot. initium – boshlanish). Bunda o‘zgaruvchanlarga, hisoblagichlarga, indekslarga va ko‘rsatkichlarga boshlang‘ich qiymatlar kiritiladi.

2. Qayta ishlash bloki. Bunda kerakli hisoblashlar amalga oshiriladi

3. Sikllarni boshqarish bloki. Bunda keyingi qaytariladigan operatsiyadan oldin hisoblagichlarni va indekslarni (ko‘rsatkichlarni) qiymati o‘zgartiriladi, hamda sikldan chiqish sharti tekshiriladi.

4. Yakunlovchi blok. Bunda olingan natijalarni saqlash amalga oshiriladi.

2 va 3-bloklar siklni jismini (loop body) tashkil qiladi. Siklik dasturlarni tez ishlash samaradorligini oshirish va hajmini qisqartirish uchun sikl qismini operatsiyadan bo'shatish kerak.

Siklik dasturlarni tashkillashtirish uchun, hamda tarmoqlashtirish uchun dasturlarda shartsiz va shartli komandalar ishlatiladi. Bundan tashqari, sikllarni qurish uchun, bir vaqtda bir qancha ishni bajaradigan, sikllarni maxsus komandalari qo'llanilishi mumkin.

JMP, RJMP, IJMP va EIJMP shartsiz o'tish komandalari boshqaruvni komandada ko'rsatilgan dastur xotirasini adresiga uzatadi. JMP (Jump) komandasi boshqaruvni dastur xotirasini butun hajmi ichidan uzatishga imkon beradi. RJMP (Relative Jump) komandasi dastur hisoblagichi joriy tarkibiga nisbatan, ±2 Kso'z (±4 Kbayt) me'yorda o'tishlar bilan ta'minlaydi. IJMP (Indirect Jump) komandasi bo'yicha Z registrida ko'rsatilgan adres bo'yicha nisbiy o'tishlarni bajaradi; maksimal siljish 64 Kso'z (128 Kbayt)ni tashkil qiladi. EIJMP (Extended Indirect Jump) komandasi dastur xotirasini butun hajmi bo'yicha nisbiy o'tishlar bilan ta'minlaydi; dastur hisoblagichini kengaytirish uchun EIND registri qo'llaniladi.

Shartli o'tishlar komandalari ba'zi bir shartlar bajarilganda dastur xotirasidagi ko'rsatilgan adres bo'yicha boshqaruvni uzatadi.

BR*xx* (Branch if ... – o'tish, agar ...) komandalari, SREG holat registri razryadini tekshiruvi natijalari bo'yicha, dastur hisoblagichining joriy tarkibiga nisbatan, –64...+63 so'z oralig'ida o'tishni bajaradi. SREG holat registri kiritish-chiqarish registrining adres muhitida bo'ladi. Shartlar kodi (C, Z, N, V, S, H) arifmetik, logik komandalar va bitlar bilan ishlash komandalari bajarilganda, holat registrida shakllanadi. Agar natijaning katta razryadidagi siljish komandasi bajarilganda, S (carry – siljish) razryadi o'rnatiladi. Agar komandani bajarilish natijasi nolga teng bo'lsa Z (zero – nol) razryadi o'rnatiladi. Agar katta razryad natijasi 1 ga teng bo'lsa, N (negative – salbiy natija) razryadi o'rnatiladi. Agar komandani bajarilishida belgili sonni razryadli setkasi to'lib qolishi sodir bo'lgan bo'sa, V (overflow – to'lish) razryadi o'rnatiladi. Belgili sonni razryadli setkasi to'lib qolganda, S = N ⊕ V (sign – belgi) belgi natijasini

to‘g‘ri ko‘rsatadi. Agar komanda bajarilish paytida natijaning uchinchi razryadidan siljish sodir bo‘lgan bo‘lsa, H (half carry – yarimsiljish) o‘rnatiladi.

Operandlarni solishtirishda tarmoqlanish uchun qo‘llaniladigan, shartli o‘tishlar komandalari 4.1-jadvalda keltirilgan.

Operandlarni solishtirishda tarmoqlanish uchun qo‘llaniladigan, shartli o‘tishlar komandalari

4.1-jadval

hart	Logik ifoda	Komanda		Operandlar
		Tekshirish	O‘tish	
Rd > Rr	Z·(N ⊕ V) = 0	CP Rr, Rd	BRLT	belgili
	C + Z = 0	CP Rr, Rd	BRLO	belgisiz
Rd ≥ Rr	(N ⊕ V) = 0	CP Rd, Rr	BRGE	belgili
	C = 0	CP Rd, Rr	BRSh/BRCC	belgisiz
Rd = Rr	Z = 1	CP Rd, Rr	BREQ	belgili, belgisiz
Rd ≠ Rr	Z = 0	CP Rd, Rr	BRNE	belgili, belgisiz
Rd ≤ Rr	Z + (N ⊕ V) = 1	CP Rr, Rd	BRGE	belgili
	C + Z = 1	CP Rr, Rd	BRSh	belgisiz
Rd < Rr	(N ⊕ V) = 1	CP Rd, Rr	BRLT	belgili
	C = 1	CP Rd, Rr	BRLO/BRCS	belgisiz

Shartli o‘tishlar komandalariga yana CPSE (Compare and Skip if Equal – solishtirish va o‘tkazib yuborish, agar teng bo‘lsa) komandasi kiradi. Bu komanda ikkita UMR tarkibini solishtiradi va, agar tarkibi bir xil bo‘lsa keyingi komandadan keyin o‘tkazib yuboradi.

Muvofiq shart bajarilsa, SBRC, SBIS, SBIC (Skip if Bit in Register [I/O Register] is Set [Cleared]) komandalari keyingi komandani o‘tkazib yuboradi. Massivlarni qayta ishlashda siklik dasturlarda ma'lumotlar xotirasini predekrementli va postinkrementli nisbiy adreslashni, hamda ma’lumotlar xotirasini siljishli nisbiy adreslashni qo‘llash foydali bo‘ladi.

4.32-chizmada dasturni bir qismi keltirilgan. Bunda 100 soni, besh baytdan tashkil topgan massiv yacheykasiga kiritiladi. Sikldan chiqish va boshqaruvni uzatish shartlarini tekshirish uchun BRNE komandasi ishlatiladi. Siklni qaytarilish chegarasi 5 ga teng, qadam -1 ga teng. Sikl parametri R16 registri tarkibiga kiradi.

```
    ; ...
array:  .byte 5          ; 5 bayt array massivi uchun
    ; ...
    LDI     R16, 5       ; sikl qaytrilish chegarasi
    LDI     R17, 100     ; array massiviga kiritiladigan son
    LDI     R18, 1
    LDI     R30, low(array) ; array massiv adresining kichik bayti
    LDI     R31, high(array) ; array massiv adresining katta bayti

    loop:                ; sikl jismi
    STZ, R17         ; array massiviga 100 sonini kiritish
    ADD  R30, R18    ; array massivining keyingi bayti adresi
    SUB  R16, R18    ; o'tishlar soni schyotchigi, qadam -1 teng
    BRNE loop        ; qaytarish, agar schyotchik nolga teng
bo'lmasa

    ; ...
```

4.32-chizma. Massivni siklik qayta ishlash dastur qismi

4.9.2. Dastur osti dasturini tuzish

Katta va qiyin dasturda ba'zi bir tugatilgan funksiyalarni bajaradigan komandalarni ketma-ketligini ajratish mumkin. Agar bunday ketma-ketlikdagi komandalarni alohida modullar – dastur osti dastur (routine, subroutine) ko'rinishda joylashtirilsa, unda, dasturda bu komandalar dastur osti dasturlarni chaqirilish komandalariga almashtirilishi mumkin. Bunday modulh (tuzilishli) dasturlash quyidagi ustunlikni beradi:

- dasturni sozlash tezlashadi va osonlashadi;
- universal funksiyalarni amalga oshiradigan dastur osti dasturlar, boshqa dasturlarni tuzishda qo'llanilishi mumkin;

- har xil dastur tillarida yozilgan, translyatsiyalangan dasturdan keyin olingan modullarni, bitta dasturda birlashtirish mumkin.

Chaqirayotgan dasturni dastur osti dastur bilan o'zaro ta'siri uchun, quyidagi shartlar bajarilishi lozim:

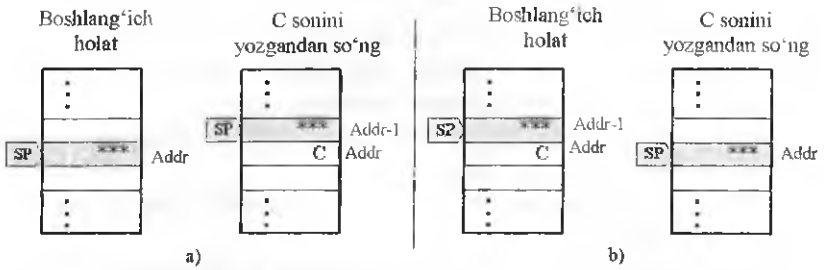
- chaqirayotgan dasturga, dasturning umumiy tuzilishidagi dastur osti dasturni holati ayon bo'lishi lozim;
- dastur osti dasturni chaqirilish va undan chiqish yo'llari aniqlanishi lozim;
- chaqirayotgan dastur va dastur osti dastur o'rtasida ma'lumotlar almashishni yo'llari tanlanilishi lozim.

Dasturning umumiy tuzilishidagi dastur osti dasturning holati, uning nomi bo'yicha aniqlanadi. Dastur osti dasturni assemblerdagi nomi, dastur osti dasturni bajarilayotgan qismini boshlanqich adresi hisoblanadi.

Dastur osti dasturini chaqirish deganda, unga komandaning bajarilish qadamini boshqarish tushiniladi. Boshqaruvni topshirish dastur osti dasturni boshlang'ich adresini dastur hisoblagichiga (RS) yuklash yo'li bilan amalga oshiriladi. Qaytarilish adreslarini saqlash uchun dastur osti dasturni *steki* (stack) ishlatiladi.

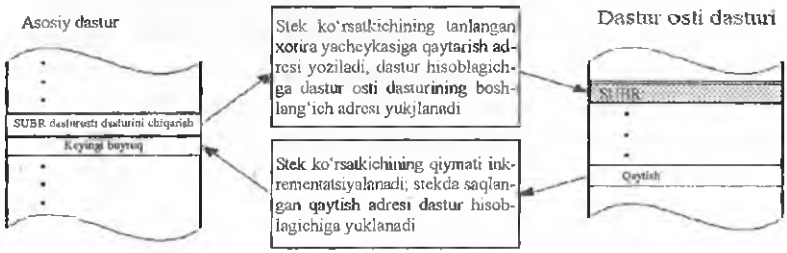
Navbatdagi bo'sh stekning adresi maxsus registrda – stek ko'rsatkichida SP (Stack Pointer) joylashadi. Son stekga yozilishida yacheykaga adresi bilan joylashtiriladi, bundan keyin stek ko'rsatkichini tarkibi bittaga kamayadi (4.33., *a* – chizma). Stekdan o'qishda, stek ko'rsatkichining bittaga ko'p tarkibli adres bo'yicha tarkibni tanlash amalga oshiriladi (4.33., *b* – chizma). Shunday qilib, stekka yozishda va undan o'qishda stek ko'rsatkichining tarkibi o'zgaradi.

Dastur osti dasturdan qaytarilgan adreslarini saqlash uchun stekni ishlatish mexanizmi, quyidagidan iborat. Qachonki dasturda, dastur osti dasturni chaqirish komandasi uchrasa, stek ko'rsatkichi tanlagan xotira yacheykasiga qaytarish adresi yoziladi, stek ko'rsatkichining qiymati dekrementatsiyalanadi; dastur hisoblagichiga dastur osti dasturning boshlang'ich adresi yuklanadi.



*4.33-chizma. Stekka yozish (a) va stekdan o'qish (b) operatsiyasi: *** – stekning navbatdagi bo'sh yacheykasi; Addr – adres.*

Keyin dastur osti dastur komandasi bajariladi. Dastur osti dasturda qaytarish komandasi uchraganda, stek ko'rsatkichining qiymati inkrementatsiyalanadi; stekda saqlangan qaytish ardesi dastur hisoblagichiga yuklanadi (4.34-chizma). Stekni dastur osti dastur bilan ishlatilishi, bitta dastur osti dastur keyingisini chaqirish imkonini beradi, ya'ni dastur osti dasturni joylash imkonini beradi. Dastur osti dasturni joylanish chuqurligi, stek hajmi bilan chegaralanadi.



4.34-chizma. Dastur osti dasturni chaqirish va chaqirilgan dasturga qaytish mexanizmi

Ko'p AVR-mikrokontrollerlarda stek operativ xotirada joylashadi. Mikrokontrollerlarda, hajmi 256 baytdan oshmaydigan operativ xotirada stek ko'rsatkichini saqlash uchun faqat bitta SPL (SP) registri qo'llaniladi. Operativ xotirasi bo'lmagan mikrokontrollerlarda uch bosqichli apparat steki bo'ladi.

Ichki operativ xotirada stekni tashkillashtirish quyidagicha bo'lishi mumkin:

```
LDI R16, low(RAMEND) ; RAMEND adresining kichik qismi
OUT     SPL, R16     ; initsializatsiya SPL
LDI R16, high(RAMEND) ; RAMEND adresining katta qismi
OUT     SPH, R16     ; initsializatsiya SPH
```

OUT komandasi UMR tarkibini kiritish-chiqarish registriga yozadi; RAMEND –ichki operativ xotiraning so'ngi yacheykasi adresining ramziy nomi. Dasturda AVR-mikrokontrollerlarini periferik qurilmalari adresida ramziy nomlarni ishlatish uchun, .include direktivasi yordamida periferik qurilmalarning adresini aniqlash faylini (inc-fayl) qo'shish kerak. Misol uchun, Atmega 8535 mikrokontrolleri uchun m8535def.inc faylini qo'shish lozim:

```
.include "m8535def.inc"
```

Qo'shilgan inc-faylga AVR Assembler oynasidagi Include Path joyiga yo'l ko'rsatilishi kerak. Bu oynani chaqirish Project menyusidagi AVR Assembler Setup komandasi orqali amalga oshiriladi. .include direktivasi qo'llanilayotganda, dasturga .device direktivasini qo'shish shart emas, chunki u inc-faylida joylashgan bo'ladi.

Translyatsiya listingida inc-faylini matnini paydo bo'lishini oldini olishda, listingni o'chiradigan .nolist direktivasi yordam beradi. .nolist direktivasi .list (listingni yoqish) direktivasi bilan birga ishlatiladi. inc-faylini translyatsiya listingidan chiqarish quyidagicha bo'lishi mumkin:

```
.nolist                  ; listing generatsiyasini o'chirish
.include "m8535def.inc"  ; inc-faylini qo'shish
.list                    ; listing generatsiyasini yoqish
```

Assembler tilida dastur osti dasturni chaqirish RCALL, ICALL, CALL va EICALL komandalari orqali amalga oshiriladi. RCALL (Relative Call – nisbiy chaqiruv) komandasi dastur osti dasturni chaqishni ta'minlaydi (boshlang'ich adresni dastur hisoblagichigini joriy qiymatiga nisbatan siljishi - ±2 Kso'z (±4 Kbayt) atrofida). ICALL (Indirect Call – qisman chaqiruv) komandasi dastur osti dasturni, dastur xotirasidagi adresdan qisman chaqiruvli amalga oshiradi (dastur osti dasturni adresini maksimal siljishi 64 Kso'z (128 Kbayt) ni tashkil

etadi). CALL (Call – chaqiruv) komandasi dastur osti dasturni, 4 Mso'z (8 Mbayt) gacha hajmga ega dastur xotirasidan chaqirishni ta'minlaydi. EICALL (Extended Indirect Call – kengaytirilgan qisman chaqiruv) dastur osti dasturni, 4 Mso'z (8 Mbayt) gacha hajmga ega dastur xotirasidan qisman chaqirishni ta'minlaydi. Ko'rsatgich stekini (SPH:SPL) tarkibi tegishli ravishda 2 yoki 3 taga kamayadi.

Dastur osti dasturni chaqiruv komandasini qo'llash misoli:

```
    RCALL    subr1      ; subr1 dastur osti dasturni nisbiy
chaqiruvi
    ; subr 2 dastur osti dasturni qisman chaqiruv
    LDI      R30, low(subr2)
    LDI      R31, high(subr2)
    ICALL    ; icall komandasi operandalarga ega emas
```

Dastur osti dasturdan qaytish uchun RET komandasi qo'llaniladi. RET komandasi bajarilayotganda, qaytarish adresi stekdan dastur hisoblagichiga yuklanadi.

Stek yana UMR tarkibini dastur osti dasturi bajarilayotgan vaqtida saqlash uchun ham qo'llanilishi mumkin. UMR tarkibini stekda saqlash uchun va chiqarish uchun PUSh va POP komandalari xizmat qiladi. PUSh komandasi registr tarkibidagini, stekka, stek ko'rsatgichida saqlanayotgan adres bo'yicha kiritadi (shunda, stek ko'rsatkichi qiymati bittaga kamayadi (SPH:SPL = SPH:SPL – 1)). POP komandasi teskari ishni bajaradi: stek ko'rsatkichi qiymati bittaga ko'payadi (SPH:SPL = SPH:SPL + 1); stek ko'rsatgichida saqlanilayotgan xotira yacheykasining tarkibi adres bo'yicha registrga yuklanadi.

Dastur osti dastur qo'llanilgan dasturlar, odatda asosiy dasturga nisbatan o'tish komandasidan boshlanadi. Bunda birinchi navbatta stekni initsializatsiya qilish bajariladi (4.35.-chizma).

Dastur osti dasturlar bilan ishlanilayotganda, chaqirilayotgan dasturdan dastur osti dasturga parametrlarni uzatishga va dastur osti dasturni bajarilish natijalarini chaqirayotgan dasturga qaytarishga imkon berish muhim rol o'ynaydi. Assemblerda chaqirayotgan dastur bilan dastur osti dasturni o'rtasida ma'lumot almashishga imkon

berish formatlashtirilmagan. Parametrlarni uzatish uchun UMR, operativ xotira yacheykalari va stek ishlatilishi mumkin.

```
.nolist                    ; listing generatsiyasini o'chirish
.include "m8535def.inc"    ; inc-faylini kiritish
.list                      ; listing generatsiyasini yoqish
RJMP   RESET               ; asosiy dasturga o'tish
PODPR:                     ; PODPR dastur osti dasturi
; ...
RET                        ; asosiy dasturga qaytish
RESET:                     ; asosiy dastur
LDI    R16, low(RAMEND)
OUT    SPL, R16            ; initsializatsiya SPL
LDI    R16, high(RAMEND)
OUT    SPH, R16            ; initsializatsiya SPH
; ...
RCALL    PODPR   ; PODPR dastur osti dasturini chaqirish
; ...
```

4.35-chizma. Dastur osti dastur qo'llanilgan dasturga misol

Parametrlarni umumiy maqsadlardagi registr orqali uzatish faqatgina kam sonli parametrlar uchun yariydi. Lekin bu, parametrlarni uzatishda, taxminiy parametrlarga eng tez yo'l beradigan, eng oddiy va ochiq yo'l hisoblanadi.

Parametrlarni operativ xotira orqali uzatish qattiq reglamentar qoidalarni talab qiladi. Misol uchun, uzatilayotgan parametrlarni yoki ularni adresini qiymatlarni to'g'ridan-to'g'ri saqlash uchun xotirada massiv (jadvalni) tashkillashtirish mumkin; boshlang'ich massivni adresini UMR ga kiritish. Boshlang'ich massiv adresiga ega bo'lib, chaqirayotgan dastur va dastur osti dasturi kerakli parametrlarga yo'l bo'ladi.

Parametrlarni stek orqali uzatishda dastur osti dasturni chaqirishdan oldin uzatilayotgan parametrlar stekga kiritiladi. Stek ko'rsatgichining qiymati asosiy adres sifatida kiritilganda va ma'lumotlar xotirasini qisman adreslashni qo'llanilganda, stekda joylashgan parametrlarga dasturda yo'l olish mumkin. Misol uchun, agar asosiy

dasturda, dastur osti dasturni chaqirishdan oldin, parametrlarning ba'zi bir qiymatlari stekga kiritilsa:

```
LDI  R16, $33  ; R16 <- $33
PUSh  R16              ; R16 registrining tarkibini stekga saqlash
unda dasturda unga kirish imkonini olish mumkin:
IN   R30, SPL  ; stek ko'rsatgichining kichik bayti
IN   R31, SPH  ; stek ko'rsatgichining katta bayti
LDD  R20, Z+3; $33 sonini stekdan R20 registiriga yuklash
```

IN komandasi kiritish-chiqarish registrini tarkibini UMR ga o'tkazish uchun xizmat qiladi. Shu singari stekni, uzatilayotgan parametrlar massivini adresini saqlash uchun ishlatish mumkin.

4.9.3. Uzilishlar asosida dastur tuzish

Mikroprotsessor (mikrokontroller) tizimi ishlaganda, kechiktirib bo'lmaydigan choralarni talab qiladigan hodisalar sodir bo'lishi mumkin. Bunday chora uzilishlar (interrupt) jarayoni, bilan ta'minlanadi, ya'ni joriy dasturni ishlashi to'xtatiladi, uzilish paytidagi holati saqlanib qolinadi, boshqa dastur bajariladi, keyin saqlanib qolingan protsessorni holati tiklanadi va uzilgan dastur ishi davom ettiriladi. Joriy dasturni uzilishidan kelib chiqgan signal – uzilish uchun so'rov (interrupt request – IRQ); deb ataladi; signalning manbai – uzilish manbasi; uzilish uchun so'rovdan keyingi harakatlar ketma-ketligi – uzilishni dastur osti dastur qayta ishlashi (interrupt handler, interrupt routine) deb ataladi.

Uzilishlar manbasini ikkita turga ajratishadi – appart va dasturiy. Apparat uzilishlar manbalariga tashqi va ichki periferik qurilmalar kiradi. Dasturiy uzilishlar manbalariga maxsus uzilishlar komandalari (trap) – boshqariladigan dasturiy uzilishlar va maxsus shart (exception) – boshqarilmaydigan dasturiy uzilishlar kiradi. Dasturiy manbadan uzilish uchun so'rov, to'g'ridan-to'g'ri uzilishlar komandasi yoki maxsus shart paydo bo'lganda aniqlanadigan bitni (bitlarni) o'rnatish orqali amalga oshiriladi. Apparat va dasturli uzilishlar manbalarining umumiy soni bittadan bir qanchagacha bo'lishi mumkin.

Har xil protsessorlardagi bir qancha manbalarning uzilishlar so‘roviga xizmat ko‘rsatish jarayoni turlicha bajariladi. Bunga qaramay uzilishlar mexanizmini amalga oshirish asosiy prinsiplari umumiy hisoblanadi. Uzilishlarni boshqarishni asosiy vositalari:

- uzilishlar vektori;
- uzilishlar imtiyozi;
- uzilishni niqoblash operatsiyasi;
- uzilishlar bayroqlari.

Uzilishlarni takomillashtirish jarayoni quyidagi bosqichlar bo‘yicha osonlashtirilishi mumkin:

- uzilishlar uchun so‘rovlarni qabul qilish;
- uzilishlar arbitraji;
- uzilishlarni takomillashtirish dastur osti dasturini bajarish.

Niqoblanmagan manbadan kelgan uzilishlar uchun so‘rovlar qabul qilinganda, darhol uni takomillashtirishni keyingi bosqichga – uzilishlar arbitrajiga o‘tiladi. Niqoblangan manbadan kelgan uzilishlar uchun so‘rovlar qiyinroq algoritm asosida takomillashtiriladi. So‘rov tushishi bilan tegishli uzilishlar bayrog‘i o‘rnatiladi. Keyin uzilishlarni umumiy niqoblash borligi tekshiriladi. Agar uzilishlarni umumiy niqoblash rejimi o‘rnatilgan bo‘lsa, unda hamma niqoblangan manbalardan uzilishlar uchun so‘rovlar rad etiladi va joriy dasturni bajarilishi davom ettiriladi. Agar uzilishlarni umumiy niqoblash rejimi o‘rnitilmagan bo‘lsa, unda bu uzilishga rad etish yoki ruxsat berish individual niqoblash borligidan (yo‘qligidan) aniqlanadi. Agar bu uzilish niqoblangan bo‘lsa, unda bu manbadan tushadigan uzilish uchun so‘rov rad etiladi va joriy dasturni bajarilishi davom ettiriladi. Aks holda bu manbadan tushadigan uzilish uchun so‘rovga ruxsat etiladi va u uchun xizmat ko‘rsatishni keyingi bosqichi – arbitraj boshlanadi.

Uzilishlar arbitraji uzilishlar uchun so‘rovlar navbatidan muhimlik darajasi katta uzilishlarni aniqlash uchun xizmat qiladi. Arbitrajdan keyin tanlanilgan uzilish uchun so‘rovni bajarish boshlanadi.

Mikroprotsessor tizimlarida uzilishlar mexanizmi, har xil kiritish-chiqarish qurilmalari o‘rtasida ma’lumotlar almashinish uchun qo‘llaniladi. Bunday ma’lumotlar almashinish uslubini uzilishlardagi almashinish deb ataladi. Uzilishlarga so‘rovlar uchun misol bo‘la oladi: analog – raqamli o‘zgartirish (ARO‘) natijalarini tayyorligi

bo‘yicha so‘rovlar, qurilmani ma’lumotlar uzatish (qabul qilish) uchun tayyorligi bo‘yicha, ba’zi bir registrlarni to‘lishi bo‘yicha, va h.k. Uzilishlar mexanizmini ishlatilishi, sekin ishlaydigan qurilmalar bilan tizim ishlayotganda, ishlash salohiyatini oshiradi.

AVR-mikrokontrollerlarda uzilishlar mexanizmi quyidagicha amalga oshiriladi. Uzilishlarni boshqarish, uzilishlar sxemasi yordamida bajariladi. Uzilishlar vektori dastur xotirasining boshiga joylashtiriladi; har bitta vektor bitta yacheykadan tashkil topadi. Kichik adresli uzilishlar, katta muhimlik darajasiga ega. Hamma uzilishlarni manbasi apparat vositalari (ichki va tashqi) hisoblanadi; dasturiy uzilishlar manbalari bo‘lmaydi. Hamma uzilishlar manbalari niqoblangan bo‘ladi. Umumiy niqoblash I bitini tozalash, SREG holat registridagi uzilishlarga global ruxsat berish bilan amalga oshiriladi. AVR-mikrokontrollerlarda uzilish vektorlari soni, turiga qarab, 3 dan 35 gacha bo‘ladi. Misol uchun, ATmega8535 mikrokontrollerlarida 21 ta uzilishlar vektori mavjud: 3 ta tashqi manbalardan va 18 ta ichki periferiyalardan tashkil topgan.

Tashqi uzilishlar bilan ishlash GICR (General Interrupt Control Register) registri va GIFR (General Interrupt Flag Register) bayroqlar registri yordamida amalga oshirilidi. GICR boshqarish registrini 7 (INT1) razryadini o‘rnatilishi INT1 tashqi uzilishlarga, 6 (INT0) razryadini o‘rnatilishi – INT0 tashqi uzilishlarga, 5 (INT2) razryadini o‘rnatilishi – INT2 tashqi uzilishlarga ruxsat beradi. GIFR bayroqlar registrini 7 (INTF1) razryadi INT1 uzilishlarga so‘rov kelganda, razryad 6 (INTF0) - INT0 uzilishga so‘rov kelganda, razryad 5 (INTF2) – INT2 uzilishga so‘rov kelganda o‘rnatiladi. O‘rnatilgan uzilishlar bayrog‘ini tozalash GIFR registrining mos razryadlariga 1 ni yozilishi bilan amalga oshiriladi.

INT0 i INT1 tashqi uzilishlarni ishga tushirish rejimi MCUCR boshqarish registridagi 0 – 3 (ISC00, ISC01, ISC10, ISC11) razryadlar kiritadi. ISC00, ISC01 razryadlariga 0, 0 mos qiymatlarini yozishni INT0 tashqi uzilishni ishga tushirish rejimi past daraja bo‘yicha kiritadi; 0, 1 – manfiy front bo‘yicha; 1, 1 – musbat front bo‘yicha; 1, 0 qiymatlari ishlatilmaydi. Xuddi shu tarzda ISC10, ISC11 razryadlari yordamida INT1 tashqi uzilishni ishga tushirish rejimi kiritiladi. INT2 tashqi uzilishni ishga tushirish rejimi, boshqarish registirining 6 (ISC2) razryadi orqali kiritiladi.

```
; uzilishlar vektori muhiti
.org $0000
  RJMP   RESET                ; asos dasturga o'tish
.org INT0addr
  RJMP   EXT_INT0             ; INT0 tashqi uzilish
.org OVF0addr
  RJMP   TMR0_INT             ; T/S0 taymer bo'yicha uzilish
; INT0 tashqi uzilishlarni qayta ishlaydigan dastur osti dasturi
EXT_INT0:
  ; ...
  RETI                        ; qaytish
; T/C0 taymeri bo'yicha uzilishlarni qayta ishlash dastur osti dasturi
TMR0_INT:
  ; ...
  RETI                        ; qaytish
  RESET:                      ; asosiy dastur
  ; stekni initsializatsiyalash
  ; ...
  ; INT0 tashqi uzilishni initsializatsiyalash
  LDI     R16, (1<<ISC01)|(1<<ISC00)
  OUT     MCUCR, R16          ; musbat front bo'yicha
  LDI     R16, (1<<INTF1)|(1<<INTF0)
  OUT     GIFR, R16           ; tashqi uzilishlarni bayroqlarini tozalash
  LDI     R16, 1<<INT0
  OUT     GICR, R16           ; INT0 tashqi uzilishga ruxsat berish
  ; T/C0 taymeri bo'yicha uzilishlarni initsializatsiyalash
  LDI     R16, 1<<CS00
  OUT     TCCR0, R16          ; chastota bo'linishi yo'q
  LDI     R16, 1<<TOIE0
  OUT TIMSK, R16              ; T/S0 taymeri bo'yicha uzilishga ruxsat berish
  SEI                         ; uzilishlarga umumiy ruxsat
  forever:
  NOP                         ; bo'sh komanda (no operation)
  RJMP    forever             ; uzluksiz -sikl
  ; ...
```

4.36-chizma. Uzilishlar qo'llanilgan dastur uchun misol

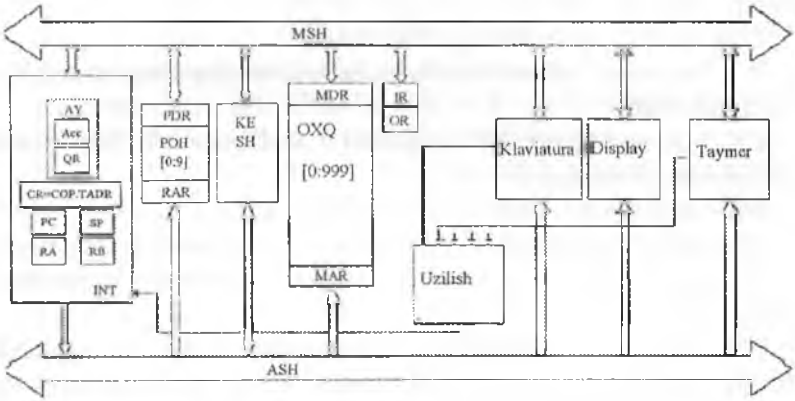
Ichki periferik qurilmatardan uzilishlarni boshqarish uchun, kiritish-chiqarish registrlarini adres muhitida ham maxsus registrlar ko‘zda tutilgan. Bundan tashqari, AVR-mikrokontrollerlarining har bir apparat qurilmalari bilan, kiritish-chiqarish registrlari adres muhitida joylashgan, boshqarish registrlari birlashgan. Misol uchun, o‘rnatilgan 8 razryadli T/C0 (Timer/Counter0) taymer-hisoblagichni boshqarish, TCCR0 (Timer/Counter0 Control Register) va TCNT0 (Timer/Counter0) registrlari tomonidan amalga oshiriladi. TCCR0 registrining 0 – 2 (CS00, CS01, CS02) razryadlari T/C0 taymer-hisoblagichiga ishlash rejimini kiritadi: 0, 0, 0 qiymatlarini mos ravishda CS00, CS01, CS02 razryadlariga yozilganda taymer-hisoblagichi to‘xtatilishi; 1, 0, 0 – TCNT0 registirning tarkibi generatorning har bir taktida inkrementatsiyalanadi; 0, 1, 0 – har bir 8-taktda; 1, 1, 0 – har bir 64-taktda; 0, 0, 1 – har bir 256-taktda; 1, 0, 1 – har bir 1024-taktda; 0, 1, 1 va 1, 1, 1 qiymatlari mos ravishda manfiy va musbat fronti bo‘yicha tashqi manbaning impulslar sonini hisoblash rejimini o‘rnatadi.

T/C0 taymer-schyotchigi TCNT0 registri to‘lganda, uzilishlar so‘rovlarni generatsiyalaydi. TIMSK niqoblar registrida T/C0 taymer-hisoblagichi to‘lganda 1 (TOIE0) razryadi uzilishga; TIFR bayroqlar registrida – 1 (TOV0) razryadi to‘g‘ri keladi. TOIE0 razryadini o‘rnatilishi TCNT0 registri to‘lganda uzilishga ruxsat beriladi; TCNT0 registri to‘lib, uzilishlar uchun so‘rov kelganda TOIF0 bayrog‘i o‘rnatiladi.

Uzilishlar qo‘llanilgan dastur uchun misol 4.36-chizmada keltirilgan.

O'QUV ELEKTRON HISOBLASH MASHINASIDA DASTURLASH

Bu elektron hisoblash mashinasi (EHM) o'quv modeli, protsessor, operativ xotira (OXQ) va yuqori operativ xotira (RON), kiritish (UVv) va chiqarish (UVv) qurilmalarini o'z ichiga oladi. O'z navbatida protsessor ham, markaziy boshqaruv qurilma (BQ), arifmetik qurilma (AQ) va registorlar tizimi (CR, PC, SP, RA, RB, va hokazolar)dan tashkil topgan(5.1-chizma)



5.1-chizma. EHM o'quv modelining strukturaviy sxemasi

EHM o'quv modelidagi protsessorning tarkibi.

Arifmetik qurilma (AQ)- bu qurilma chiqayotgan ma'lumotlarda arifmetik va mantiqiy amallarni bajaradi va natijani chiqish registriga joylaydi. Agar kerak bo'lsa, chiqish registoridagi ma'lumotlarni boshqa bir registrga yoki xotiraga saqlashi mumkin. Akkumulyator (Acc) va ma'lumotlar registrida aynan qanday amal bajarilishi kod operatsiyasi (COP) yordamida belgilanadi. Bajarilgan amalning natijasi akkumulyator (Acc)ga joylashtiriladi.

Akkumulyator (Accumulator) - bu arifmetik yoki mantiqiy operandlardan birini saqlash uchun qo'llaniladigan registr. Akkumulyator (Acc)da oldin bajarilgan va keyingi bajarilishi kerak bo'lgan amallarning natijasini saqlash mumkin. Shu qatorda akkumulyator (Acc) yordamida kiritish va chiqarish amallari bajariladi.

Ma'lumotlar registri (Data Register)- bu registr operativ xotira (OXQ) va protsessorni tez ishlash farqini anglash uchun mo'ljallangan. Xotira va protsessorda axborot almashinayotgan paytida vaqtinchalik saqlash uchun qo'llaniladi. Operand ma'lumotlar registri (RD)da joylashgandagina, biz arifmetik va mantiqiy amallarni bajarishimiz mumkin. Misol uchun: ma'lumotlar registri (RD)dagi qiymatni, akkumulyator (Acc)dagi qiymatga qo'shib va olingan natijani yana akkumulyator (Acc)ga joylashimiz mumkin.

Boshqarish qurilmasining (BK) tuzilishi

PC- buyruqlar adresi hisoblagichi, navbatdagi buyruq adresini o'z ichiga oladi.

CR- buyruq registri, buyruq kodini o'z ichiga oladi(ya'ni buyruq kodidan tashkil topgan.)

RB- baza registri adresi (baza adresidan tashkil topgan).

SP-stek ko'rsatkichi (stekning quyi adresidan tashkil topgan).

RA-adres registri, mavhum adreslash jarayonida adreslashni amalga oshirishdan iborat.

Buyruq adresi hisoblagichi (Command Register)- bevosita EHMda bajarilgan buyruq kodini saqlash uchun qo'llaniladi. Biror bir amal bajarilishidan oldin, buyruq adresi hisoblagichi (Pc)ga bajarilishi kerak bo'lgan, buyruq saqlangan xotira yacheyka adresi kiritiladi. Hisoblash ketma-ketligini o'zgartirish uchun, o'tish nuqtasini buyruq adresi hisoblagichiga kiritish yetarli bo'ladi.

Komanda registri (Command Register): to'g'ridan to'g'ri mashinada bajariladigan buyruq kodini saqlash uchun qo'llaniladi. Buyruqni bajarishga kirishishdan oldin, buyruq operatsiya kodi (COP) ni xotiradan chiqarib, buyruq operatsiya kodi (COP)ni xotiradan chiqarib, (COP) butun jarayon davomida saqlanib turadigan komanda registri (CR) ga joylashtirish kerak. Komanda registri (CR) maydoni quyidagilardan tashkil topgan: (COP)- operatsiya kodi, (TA)- adreslash turi, (ADR)- adres yoki to'g'ridan to'g'ri operand.

Bazaviy registr adresi (Register Base): o‘zida xotiradagi ba’zi bir obyektlarning bazaviy ko‘rsatkich adresini saqlaydi. Boshqacha qilib aytganda, xotira yacheyka adresini, baza registriga saqlash mumkin, undan keyin esa ana shu xotira adresida joylashgan operanddagi buyruqni bajarish mumkin.

Baza adresi-bu ba’zi bir ma’lumotlar struktura adresining boshi hisoblanadi (misol uchun: ma’lumotlar massivi).

Stek ko‘rsatkichi (Stack Pointer): stek cho‘qqisi adresidan tashkil topgan registr.

Stek cho‘qqisi- bu vaqt bo‘yicha eng oxirgi qaydlar kiritilgan yacheykadir.

Adresni bajaruvchi registr (Registr Address): mavhum adreslash jarayonida, bajarilayotgan adresni o‘z ichiga oladi.

Bajarilayotgan adres- bu operand yozilishi yoki o‘qilishi mumkin bo‘lgan yacheyka raqam kodi.

Registrlar Acc, DR, IR, OR, CR va 6 o‘nlik razryad uzunligiga ega, PC, SP, RA va RB lar 3 razryad uzunlikka ega.

Holat registori. Arifmetik qurilma (AQ) jarayoni bajarilishini yakunlash paytida, natija alomatlari Z, S, OV signallarini keltirib chiqaradi. Alomatlar signali natijasi, registr holatini yoki bayroq registrini aks ettiradi.

Bayroq registri- bu oxirgi arifmetik va mantiqiy amallar alomatlari natijasini belgilashga va saqlashga mo‘ljallangan registr.

Bayroq har bir sikl (AU)ga o‘rnatiladi va oldingi jarayon natijasining holati haqida ma’lumot beradi.

- Natija nolga teng: registrda yagona holatda bayroq Z(Zero) o‘rnatiladi.
- Natija manfiy: registrda yagona tartibda bayroq S (Sign) o‘rnatiladi.
- To‘lib ketishi: to‘lib ketganda stek razryadi yagona tartibda OV (Over Flow) ga o‘rnatiladi.

Yuqoridagi shartlar bajarilmagan holda mos kelgan signallar nol qiymatga ega bo‘ladi.

Operativ xotira qurilmasi.

Operativ xotira (OZU) yacheykasida buyruq va ma’lumotlar saqlanadi. (OZU)ning sig‘imi 1000ta yacheykaga teng. MWr

(Memory Write) signali asosida adres registrida (MAR)da ko‘rsatilgan ma’lumotlar registri (MDR) xotira yacheykasiga adresi bilan saqlanadi. MRd (Memory Read) signali asosida esa (MAR)da joylashgan xotira yacheykasi tarkibini adresi bilan birga o‘qishi yuzaga keladi va (MDR) [ ] ga jo‘natiladi. OZU yacheykalari 6 o‘nlik razryad uzunligiga ega.

Xotiradagi ma’lumotlar registri (Memory Data Register): xotira va protsessorning qolgan registri orasida bufer rolini bajarishda ishlatiladi. Shu orqali protsessorga buyruqlar (operandlar) o‘tkaziladi va xotiraga bajarilgan amallar natijasini jo‘natadi.

Xotiradagi adres registri (Memory Address Register): biror bir amal, jarayon (o‘tish yoki saqlash) yakuniga yetmaguncha, shu yacheykada xotira yacheyka adresini saqlash uchun mo‘ljallangan. Xotira registri adresi mavjudligi, operativ xotira qurilmasi va shu kabi mashina qurilmalarini tez ishlash farqini kompensatsiya qilishga ruxsat beradi.

To‘g‘ridan-to‘g‘ri ruxsatli o‘ta operativ xotira: operativ xotirali qurilmaning bir qismi deb hisoblanmaydi. Umumiy belgilanishi R0÷R9 gacha bo‘lgan 10ta registrdan iborat (RON). Ma’lumki (RON) – bu juda ham katta bo‘lmagan registrli xotira bo‘lib, unga faqatgina alohida bo‘lgan buyruq bilan ruxsat etiladi. Odatda RONga ko‘p marotaba ishlatiladigan adres, sikl hisoblagichi va boshqalar kiritiladi. Unga faqatgina RAR (Register Address Register) va RDR (Register Data Register) registrlari orqali ruxsat etiladi.

Odatda registrlar kattaligi EHM o‘quv modelida belgilangan, RON esa 6 o‘nlik razryad uzunligiga ega.

Xohlagan registrga raqamini kiritib murojaat qilish mumkin. Registr arxitekturasi ikki xil muhitning biriga: asosiy xotiraga yoki registrlarga joylashtirish ruxsat etiladi. Operandlarni 3 xil turda joylashtirish mumkin:

- Registr-registr: operandlar faqatgina registrda joylashgan bo‘ladi.
- Registr-xotira: operandlardan biri registrda ikkinchisi esa asosiy xotirada joylashgan bo‘ladi.
- Xotira-xotira: ikkala operand ham asosiy xotirada joylashgan bo‘ladi.

Buyruq formati: buyruq formati tarkibida yo'nalishi va alohida buyruqlar maydonini joylashishi tushuniladi. Ko'pincha EHM o'quv modelidagi buyruqlar bir adresli yoki adressiz hisoblanib, 6 razryad uzunligiga ega.

Buyruq formatida 3 xil maydon ajralib chiqadi:

•Amallar kodi (SOR) ni belgilovchi ikkita katta razryad [0:1].

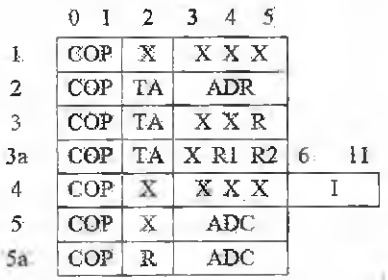
•2-razryad. Adreslash turini aniqlashi mumkin.

•[3:5]-razryad. Xotiradagi to'g'ri yoki mavhum adreslashni, registr raqamini, o'tish adresi yoki qisqa to'g'ridan to'g'ri (bevosita) operandni aniqlaydi. Ikki so'zli buyruqda bevosita operand [6:11] maydonini egallaydi.

Bir adresli formatda ishlash uchun protsessorda akkumulyator (Acc) ko'zda tutilgan. 1-operand va natija har doim akkumulyator (Ass)ga joylashadi. 2-operand bo'lsa [3:5] maydonda adreslanadi (joylashadi).

Adressiz buyruqda adreslash maydoni yo'q. Bevosita adreslash va ikki adresli hisoblanuvchi MUV buyrug'i ikkiso'zli buyruqlar istisno hisoblanadi.

EHM o'quv modelida quyidagi belgili formatlar ro'yxati keltirilgan.



5.2-chizma. EHM o'quv modelida buyruqlar formati

•COP (Sode OReration) – amallar kodi;

•ADR (Address Data Register) – xotiradagi operand adresi;

•ADC (Address Data to Cell)– buyruqga boshqarish uzatishini o'tish adresi;

•I (Immediate operand) – bevosita operand;

•R, R1, R2 – registr raqami;

•TA (Type of Addressing) –adres turi;

•X – razryad ishlatilmaydi.

EHM o‘quv modelidagi buyruqlar tizimi quyidagi 5.1-jadvalda to‘liq keltirilgan.

Buyruqlar tizimi

5.1-jadval

KOP	Mnemokod	Nomlanishi	Bajarilishi
00	NOP	Bo‘sh jarayon	Yo‘q
01	IN	Kiritish	Ass ← IR
02	OUT	Chiqarish	OR ← Asc
03	IRET	Uzilishdan qaytish	FLAGS.PC ← M(SP); INC(SP)
04	WRRB	RB baza registrini yuklash	RB ←CR[ADR]
05	WRSP	SP stek ko‘rsatkichini yuklash	SP ← CR[ADR]
06	PUSH	Stekga joylashtirish	DEC(SP); M(SP) ← R
07	POP	Stekdan chiqarish	R → M(SP); INC(SP)
08	RET	Qaytarish	PC → M(SP); INC(SP)
09	HLT	Stop	komanda -sikli tugallandi
10	JMP	shartsiz o‘tish	PC ← CR[ADR]
11	JZ	0 bo‘lganda o‘tish	if Asc = 0 then PC ← CR[ADR]
12	JNZ	0 bo‘lmaganda o‘tish	if Asc ≠ 0 then PC ← CR[ADR]
13	JS	Manfiy bo‘lganda o‘tish	if Asc< 0 then PC ← CR[ADR]
14	JNS	Musbat bo‘lganda o‘tish	if Asc> 0 then PC ← CR[ADR]
15	JO	To‘lib ketganda o‘tish	if |Acc| ≥ 99999 then PC ← CR[ADR]
16	JNO	To‘lmaganda o‘tish	if |Acc| ≤ 99999 then PC ← CR[ADR]

5.1-jadvalning davomi

17	JRNZ	TSikl	DEC(R); if R > 0 then PC ← CR[ADR]
18	INT	Dasturiy uzilish	DEC(SP); M(SP) ← FLAGS.PC; PC ← M(V)
19	CALL	Dasturosti chaqirish	DEC(SP); M(SP) ← PC; PC ← CR(ADR)
20	YO'Q		
21	RD	o'qish	Acc ← DD
22	WR	yozish	M(*) ← Acc
23	ADD	Qo'shish	Acc ← Acc + DD
24	SUB	Ayirish	Acc ← Acc – DD
25	MUL	ko'paytirish	Acc ← Acc x DD
26	DIV	bo'lish	Acc ← Acc/DD
27	YO'Q		
28	EI	Uzilishga ruxsat berish	IF ← 1
29	DI	Uzilishga rad etish	IF ← 0
30	MOV	ko'chirish	Rl ← R2
31	RD	O'qish	Acc ← R*
32	WR	Yozish	R* ← Acc
33	ADD	Qo'shish	Acc ← Acc + R*
34	SUB	Ayirish	Acc ← Acc - R*
35	MUL	Ko'paytirish	Acc ← Acc x R*
36	DIV	Bo'lish	Acc ← Acc/R*
37	IN	Kiritish	Acc ← BU(CR[ADR*])
38	OUT	Chiqarish	BU(CR[ADR*]) ← Ass
39	YO'Q		
40	YO'Q		
41	RDI	O'qish	Ass ←I
42	YO'Q		
43	ADI	Qo'shish	Ass ← Asc + I
44	SBI	Ayirish	Ass ← Asc -I
45	MULI	Ko'paytirish	Ass ← Asc x I
46	DIVI	Bo'lish	Ass ← Ass/I

5.1. -jadvalda quyidagi belgilar qabul qilingan.

Acc – akkumulyator

PC – buyruqlar hisoblagichi

SP – stek ko‘rsatkichi.

RB – baza registri.

IR – kiritish registri.

OR – chiqarish registri.

FLAGS – bayroqlar vektori: IF,OV,S,Z

IF – uzilishga ruxsat beruvchi bayroq.

DD – (ikkinchi) operand sifatida buyruqni shakllantiruvchi ma’lumot xotira yacheykasida to‘g‘ri yoki mavhum adreslash yoki bo‘lmasa 3 razryadli bevosita operand;

R – umumiy belgilangan registr tarkibi (RON).

R* – xotira yacheyka registri orqali RON yoki mavhum adreslash tarkibi;

M( ) – xotira yacheykasi tarkibi.

M(*) – buyruqga xotira yacheykasini to‘g‘ri yoki mavhum adreslash tarkibi;

CR – buyruq registri;

CR[ADR] – CR registrining 3 razryadli maydoni ADR;

CR[ADR*] – CR registrining ikkita kichik razryad maydoni ADR;

V – xotira adresi, uzilish vektoriga mos;

I – belgili 5 razryadli bevosita operand.

Adreslash turi, ularning kodi va belgilanishi

5.2 -jadval

Belgilanishi	KodTA	Adreslash turlari	Bajariluvchi adres	Komandalarga misol
	0	To‘g‘ri (registrli)	ADR (R)	ADD 23 (ADD R3)
#	1	bevosita	-	ADD #33
@	2	Asoslangan holda	OZU(ADR)[3:5]	ADD @33
[ ]	3	Nisbatan	ADR + RB	ADD [33]

5.2-jadvalning davomi

@R	4	mavhum registri	RON(R)[3:5]	ADD @R3
@R+	5	Indeks postin-krementli	RON(R)[3:5], R:=R+1	ADD @R3+
-@R	6	Indeks pred-dekrementli	R:=R-1, RON(R)[3:5]	ADD -@R3

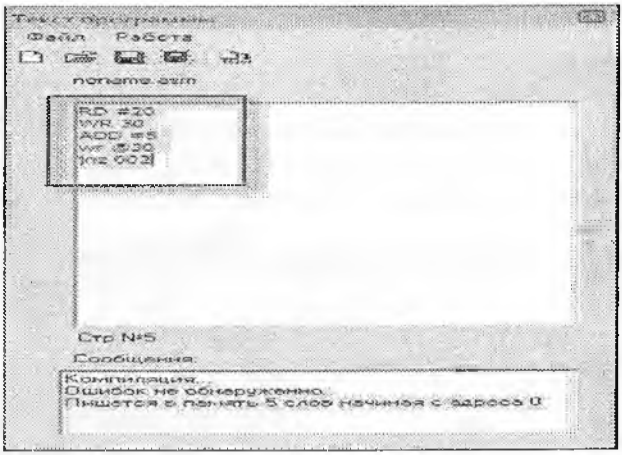
Ishni bajarish tartibi.

1. Mashina kodini shakllantirish uchun mnemokodlar ketma-ketligi berilgan, EHM2 (OZU) siga kiriting.

Har bir variant uchun mnemokodlar ketma ketligi o'qituvchi tomonidan beriladi.

OXQda buyruq kiritish uchun quyidagilami bajarish lozim.

1-qadam. «Текст программы» buyruqlar ketma-ketligini kiriting.

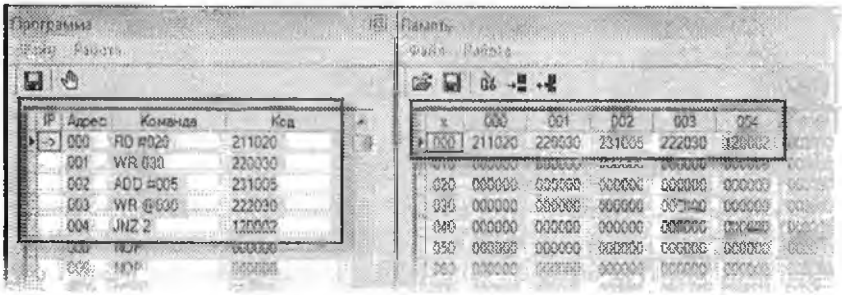


5.3-chizma. Dastur matnini kiritish

2-qadam. Kompilyatsiya jarayonini ishga tushirish.

Shu qatorda OXQ yacheykasida dizassemblerlashgan buyruq ishga tushadi. 000 adresidan boshlanib «программа» oynasida avtomatik tarzida paydo bo'ladi.

Kompilyatsiya ishga tushgandan keyin «Память» va «Программа» oynasi ochiladi.



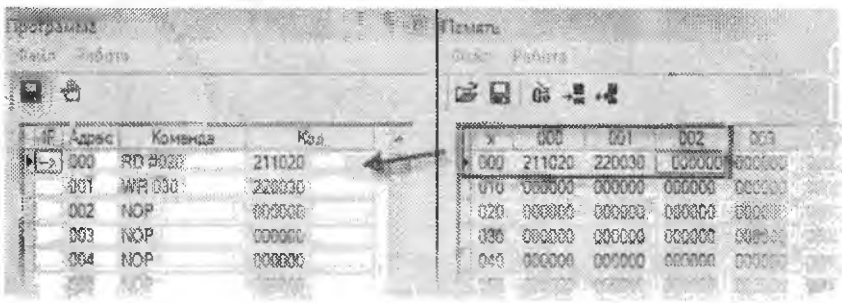
5.4-chizma. Dastur va dastururiy xotira

Bevosita buyruq kodi bilan ham ishlash mumkin. Buning uchun: OXQ ga 000 adresidan boshlab buyruq kodini ketma ket kiritish lozim

Keyingi yacheykaga o‘tish paytida, avtomatik tarzda assemblerlash buyruq «Текст программы» oynasida ko‘rinadi.

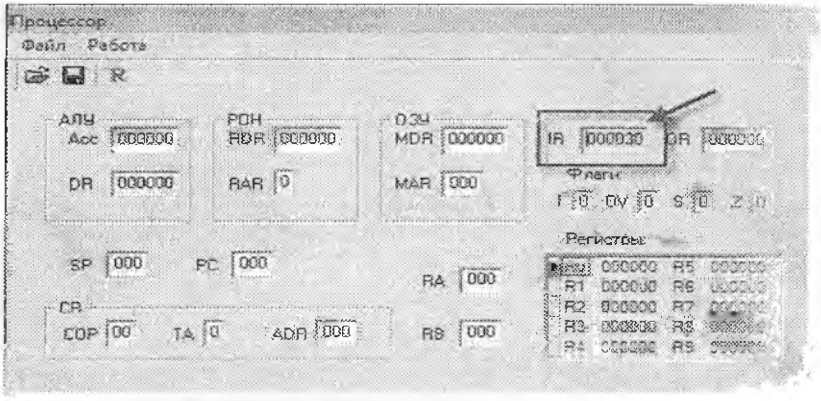
Misol uchun « operatsiya kodi».

RD-21, bevosita operand (20)ni adreslashimiz kerak. Bevosita amalni kodi - 1 xotira yacheykasiga (211020) kodini, RD #20 buyrug‘ini kiritamiz va «Память» oynasidagi GO tugmasini bosamiz.



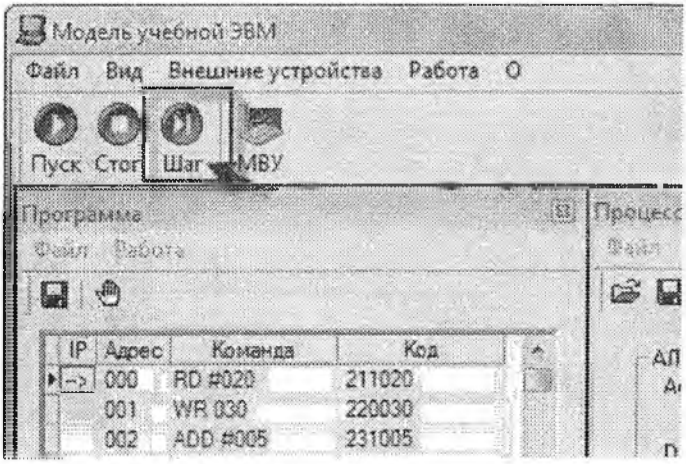
5.5-chizma. Operatsiya kodi

2. Variant bo‘yicha IR-kirish qurilmasiga boshlang‘ich qiymatlarni kiritamiz.



5.6-chizma. Operandni kiritish

3. «Shag» rejimida buyruq ketma ketligini bajarish



5.7-chizma.Buyruqlarni ketma-ket bajarish rejimi

4. Mnemokodlar va mos mashina kodini ketma-ketlik bilan jadval ko‘rinishida to‘ldirish.

Buyruqlar va kodlar

5.3-jadval

Adres	Mnemokod	Kod	Izox
000	RD #20	211020	ASS←20
001	WR 30	220030	M(30) ← ACC
002	ADD #5	231005	ASS ← ASS + 5
003	WR@30	222030	ASS ←M(M(30))
004	JNZ 002	120002	002 adres bo'yicha komandaga o'tish, agar ≠ 0

5. «Shag» rejimida buyruqlarni bajarayotganda dasturiy ruxsat berilgan obyektlarni o'rganishni belgilab qo'yish. (bu holatda bular Ass, RS va OZU yacheykasi) M(30), M(M(30)).

Agar dasturda sikl kelib chiqsa, u holda ikki martadan ortmagan qaytarilgan siklga tegishli har bir buyruqni ko'rib chiqish kerak bo'ladi.

Registrlar tarkibi

5.4- jadval

PC	Asc	M(30)	M(M(30)
000	000000	000000	000000
001	000020		
002		000020	
003	000025		
004			000025
002			
003	000030		
004			000030

6. Variantga mos holda EHM o'quv modeli holatini mikro-komanda darajasida yozish.

Mikrokomanda darajasidagi modellashtirish rejimida model holati

5.5-jadval

Адрес (PC)	Мнемокод	Микрокоманда	ОЗУ		CR			АУ		Ячейки	
			MAR	MDR	COP	TA	ADR	Acc	DR	020	030
000	RD #20	MAR := PC	000	000000	00	0	000	000000	000000	000000	000000
		MRd	000								
		CR := MDR		211020							
		PC := PC+1			21	1	020				
001		Acc := ADR									
	WR 30	MAR := PC						000020			
		MRd	001								
		CR := MDR		220030							
		PC := PC+1			22	0	030				
002		MAR := ADR									
		MDR:=Acc	030								
		MWr		000020							
	ADD #5	MAR := PC									000020
		MRd	002								
		CR := MDR		231005							
		PC := PC+1			23	1	005				
003		DR := ADR									
		Fnc =ALT							000005		
	WR @30	MAR := PC						000023			

7. Dasturni saqlash uchun:

«Текст программы» oynasidagi «Файл» menyusi tanlanib «Сохранить как» tanlanadi.

5.8-chizma. Dasturni saqlash oynasi

Ochilgan oynada faylni diskga joylash kerak va faylni nomlash kerak. Fayl tipini ASM qilib «Сохранить» tugmasi bosiladi.

Shartli o‘tishlarni tashkil qilish va ma’lumotlarni niqoblash

Ko‘pincha dastur ishining bajarilishi jarayonida akkumulyatordagi son razryadlarini tekshirish yoki o‘zgartirish (niqoblash) zaruriyati paydo bo‘ladi. Buni quyidagi amallar orqali amalga oshirish mumkin.

1) Akkumulyatordagi son va niqob mantiqiy ko‘paytiriladi. Agar niqobning mos razryadlariga 0 yozilgan bo‘lsa, bunda razryaddagi son tozalanadi, agarda niqob razryadiga 1 yozilgan bo‘lsa, u hech narsani o‘zgartirmaydi.

1- Misol 1:

ANI,22N buyrug‘i (22N soni niqob vazifasini bajaradi)

01110011	akkumulyatordagi son
00100010	niqob
00100010	akkumulyatordagi natija

2 – Misol:

ANI ,FO	buyrug‘i
01011111	akkumulyatordagi son
11110000	niqob
01010000	akkumulyatordagi natija

2) Akkumulyatordagi son va niqob mantiqiy qo‘shiladi. Agar shu niqob razryadida 1 soni bo‘lsa, bunda 1 soni razryadda o‘rnatiladi, agar shu razryadga 0 soni yozilgan bo‘lsa, u berilgan sonni o‘zgartirmaydi.

3 - misol		4 - misol	
ORI, 22N	buyrug‘i	ORI , FO	buyrug‘i
01110011	akkumulyatordagi son	01011111	akkumulyatordagi son
00100010	niqob	11110000	niqob
01110011	akkumulyatordagi natija	11111111	natija

3) Akkumulyatordagi son va niqob mantiqiy rad etuvchi "yoki" amalini bajaradi. Agar berilgan niqob razryadiga 1 soni yozilgan bo‘lsa, bunda razryad teskarilanadi (inversiyalanadi), agarda bu razryadda 0 soni yozilgan bo‘lsa, u o‘zgarmaydi.

5 - misol		6 - misol	
XRI,22N	buyrug‘i	XRI ,FO	buyrug‘i

01110011	akkumulyatordagi son	01011111	akkumulyatordagi son
00100010	niqob	11110000	niqob
01010001	akkumulyatordagi natija	10101111	natija

Shartli boshqarish buyruqlari: JZ, JNZ, JS, JNS, JO, JNO.

JS – If Asc < 0 then PC ← CR[ADR]. Agar operatsiya natijasi noldan kichik bo'lsa, (Ass < 0), belgi razryadida 1 bo'lsa, ADR buyrug'ining qismi PC registrga ko'chiriladi, shu bilan o'tish tashkil qilinadi;

JNS – If Asc > 0 then PC ← CR[ADR]. O'tish, agar natija noldan katta bo'lsa Ass > 0, belgi razryadidida nol;

JZ – If Asc = 0 then PC ← CR[ADR]. O'tish, agar operatsiya natijasi nolga teng bo'lsa, Ass = 0;

JNZ - If Asc ≠ 0 then PC ← CR[ADR]. Agar operatsiya natijasi nolga teng bo'lmasa, Ass ≠ 0;

JO – If |Acc| ≥ 99999 then PC ← CR[ADR]. Ushbu holatda boshqarish razryad setkasi to'lib ketganda uzatiladi;

JNO - If |Acc| ≤ 99999 then PC ← CR[ADR]. O'tish, agar razryadlar setkasi to'lib ketmasa.

Misol, agar musbat x sonini darajaga ko'tarish kerak. Birinchi navbatda berilgan sonni musbat ekanligi tekshiriladi. Agar son manfiy bo'lsa dastur ishni to'xtatadi. Agar musbat bo'lsa, dastur berilgan sonni darajaga ko'taradi va ishni to'xtatadi.

Ushbu algoritmni JS buyrug'i orqali amalga oshirish mumkin, agar natija manfiy bo'lsa.

Dastur:

```
000 IN        // sonni kiritish
001 JS 005    // o'tish,agar son manfiy bo'lsa
002 WR 10     // sonlarni 010 xotira yacheykasiga yozish
003 MUL 10    // sonlarni darajaga ko'tarish
004 OUT       // natijani chop etish
005 HLT       // to'xtatish
```

Qism dasturlar va stek

Misol. Sonning uchta massivi berilgan. Ularning maksimal elementlarining o'rta arifmetigini hisoblash talab qilinadi. Har bir massiv ikkita parametr orqali beriladi: birinchi elementning adresi va uzunligi.

- Birinchi massiv – boshlang'ich adresi 85, uzunligi 14;

- Ikkinchi massiv - boshlang'ich adresi 100, uzunligi 4;
- Uchinchi massiv - boshlang'ich adresi 110, uzunligi 9;

Dastur qism dasturga paremetrlarni yuklaydi, uni chaqiradi va podprogramma ishining natijasini ish yacheykalarida saqlaydi. So'ng o'rta arifmetik qiymatini hisoblaydi va natijani chiqish qurilmasida chiqaradi. Ish yacheykalari sifatida umumiy vazifa uchun mo'ljallangan registrlar ishlatiladi R6 va R7 – massivlarning maksimal elementlarini saqlash uchun.

Dasturda uch marta massivning maksimal elementini qidirishni bajarish kerak, shuning uchun mos keluvchi podprogrammani yozish kerak. Qism dastur parametrlarini R1 va R2 registrlari orqali oladi:

R1 – massivning boshlang'ich adresi;

R2 – massivning uzunligi.

Bu registrlar hozirgi adresning registri va hisoblagichning sikli sifatida podprogramma tomonidan ishlatiladi.

R3 – hozirgi maksimumni saqlash uchun ishlatiladi;

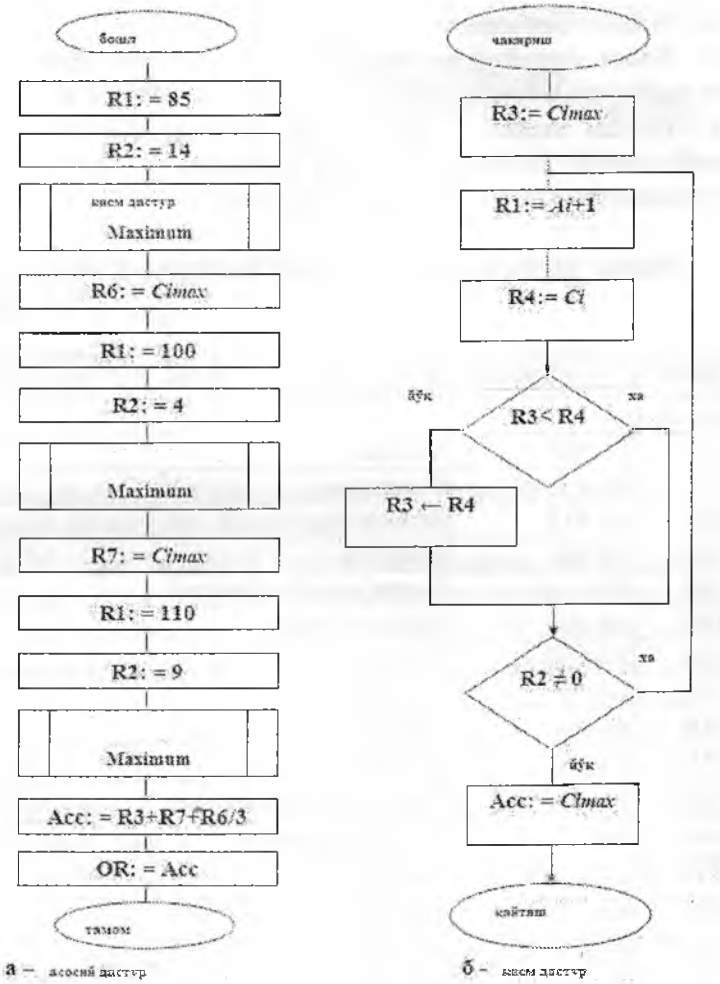
R4 – hozirgi elementni vaqtinchalik saqlash uchun.

Qism dastur natijani akkumulyator (Ass) orqali qaytaradi. Qism dasturda sikl JRNZ buyrug'i yordamida tashkil qilingan, hozirgi adresning modifikatsiyasi esa – postinkrement adresning vositalari bilan. Dastur matni 5.6-jadvalda keltirilgan.

2) Asosiy dasturning algoritmini blok sxemasini tuzish, 5.9.-chizma, a harfi:

R1, R2 – dastur bu registrlarga parametrlarini yuboradi: boshlang'ich adres va massivlarning uzunligi.

Cimax – massivning maksimal elementi: podprogramma bajarilgandan keyin birinchi massivning maksimal elementi R6 ga yuboriladi, ikkinchisi - R7 ga. Uchinchi massivning maksimal elementi, podprogramma bajarilgandan keyin akkumulyatorda joylashgan bo'ladi, unga ketma-ketlik bilan R6 va R7 ni tashkil etuvchilari kelib qo'shiladi. Qo'shish operatsiyasidan keyin, bo'lish operatsiyasi amalga oshiriladi, ya'ni o'rta arifmetigi topiladi. Natija OR qurilmasiga chiqariladi.



5.9-chizma. Asosiy dastur va qism dastur blok-sxemalari

3) Qism dasturni algoritmini blok sxemasini tuzish, 5.9-chizma, b harfi:

R3 – bu registrga qism dastur Cimax massivining hozirgi maksimumini yuboradi, ya'ni massivning qaysi elementi qism dasturni ish vaqtida maksimal hisoblanadi;

R1 – hisoblagich;

S*i* – massiv elementi.

4) Xotira quyidagicha taqsimlangan: 000-030 yacheykalari, dastur egallaydi; birinchi massiv – boshlang'ich adresi 085; uzunligi – 14; ikkinchi massiv - boshlang'ich adresi 100; uzunligi – 4; uchinchi massiv - boshlang'ich adresi 110; uzunligi – 9.

5) Dasturni tuzish va sozlash, 5.6-jadval;

Massiv elementlarini hisoblash dasturining matni

5.6-jadval

Adres	Komanda	Izoh
Asosiy dastur		
000	RD #85	Yuklash
001	WR R1	Parametrni
002	RD #14	Birinchini
003	WR R2	Massivni
004	CALL M	Dastur asosida chaqirish
005	WR R6	Natijani saqlash
006	RD #100	Yuklash
007	WR R1	Parametrni
008	RD #4	Ikkinchini
009	WR R2	Massivni
010	CALL M	Dastur asosida chaqirish
011	WR R7	Natijani saqlash
012	RD #110	Yuklash
013	WR R1	Parametrni
014	RD #9	Uchinchini
015	WR R2	Massivni
016	CALL M	Dastur asosida chaqirish
017	ADD R7	Hisoblash
018	ADD R6	O'rtachasini
019	DIV #3	Arifmetigini
020	OUT	Natijani chiqarish
021	HLT	Tamom
Maximum Qism dastur		

5.6-jadvalning davomi

022	M: RD @R1	Yuklash
023	WR R3	R3 dagi birinchi elementni
024	L2: RD @R1+	Adresni modifikatsiyasi va elementini o'qish
025	WR R4	Solishtirish
026	SUB R3	va almashtirish.
027	JS L1	agar R3 < R4
028	MOV R3,R4	
029	L1: JRNZ R2,L2	Sikl
030	RD R3	Asc dagi natijani o'qish
031	RET	Chiqish

Ma'lumotlarni kiritish-chiqarish,tashqi qurilmalar bilan ma'lumot almashishni dasturlash.

EHM o'quv modelida –tashqi uzilishlarning vektor mexanizmi keltirilgan. Tashqi qurilmalar uzilishlarga so'rovlarni ishlab chiqishi natijasida, uzilishlar kontrolleri kirishiga kelib tushadi. Uzilishlar so'rovini tashkil qilishga qodir bo'lgan TQ ulanishi bilan, unga 0-9 gacha diapazonda qiymat olayotgan kontroller kirishida mos nomerlanadi. Kontroller vektorni protsessorga uzatadi, u esa o'z navbatida uzilishlarga xizmat ko'rsatish jarayonini boshlaydi.

O'quv modelidagi uzilishlar mexanizmi operativ xotiradagi uzilishlar vektori jadvalini qo'llaydi. Jadvaldagi qator uzilishlar vektoriga, jadval elementlari xotira yacheykasiga moslashtirilgan. Ko'rilayotgan modelda uzilishlar jadvali 100-109 adresli xotira yacheykasini egallaydi. Shunday qilib, 0 vektorli qayta ishlovchi adresi 100 yacheykada, 2 vektor - 102 yacheykada joylashadi. Ish jarayonida 100-109 yacheykalarni boshqa maqsad uchun qo'llanilmaydi.

Navbatdagi buyruq bajarilishi bilan, protsessor uzilishlarni qayta ishlashni boshlaydi:

1. Kontrollerdan uzilishlar vektorini qabul qiladi.

2.So'z steki cho'qqisiga ishlab chiqaradi va joylashtiradi-DH (dastur hisoblagichi) navbatdagi qiymatiga uch kichik razryad ([3:5]), [1:2] razryadlar 16 raqamning o'nlik ekvivialentini saqlaydi, ular o'z navbatida (I, OV, S, Z) bayroq vektori qiymatlarini aniqlaydi.

Masalan, agar I=1, OV = 0, S = 1, Z = 1,bo‘lsa [1:2] razryadlarga 1110 =10112 son yoziladi.

3. 1 uzilishlarga ruxsat bayrog‘i 0 tenglashtiradi

4. Uzilishlar vektori jadvalida qayta ishlash adresini topib DH ga joylashtiradi va natijada uzilishlarni qayta ishlash qism dasturiga o‘tiladi.

Shunday qilib, uzilishlarni qayta ishlashni chaqirish stekga nafaqat qaytish adresini balki bayroq vektorining navbatdagi qiymatini ham joylaydi.

Shuning uchun qayta ishlash qism dasturining oxirgi buyrug‘i IRET bo‘lishi kerak.

Protsessor tarkibiga uzilishlarga ruxsat I bayrog‘i ham kiradi. I = 0 bo‘lsa, uzilishlarga so‘rov signallarga e’tibor bermaydi. Protsessor qayta yuklanishi natijasida hamma so‘rovlar taqiqlanadi. Uzilishlarga so‘rovni amalga oshirish uchun EI (angl. enable interrupt) buyrug‘i bajariladi. Yuqorida keltirilganidek, uzilishlarni boshqa uzilishlar bilan to‘xtatish mumkin emas. IRET buyrug‘i berilish bilan bayroqlar qiymati qayta tiklanadi I=1 va asosiy dasturda uzilishlarga so‘rov davom etadi. Agar boshqa uzilishlarni kiritish kerak bo‘lsa, EI buyrug‘idan foydalaniladi.

Agar dasturchi dastur tugallanishini ko‘rsatmasa, u holda protsessor apparat pog‘onasida uzilishlarni inkor etadi.

Shunday qilib, 1 bayroq uzilishlarga ruxsat beradi yoki taqiqlaydi. Agar bir nechta so‘rovlarni amalga oshirish kerak bo‘lsa, u holda dasturiy murojaat etiladi.

Qoidaga binoan, har bir TQ o‘zida uzilishlarga ruxsat bayroq registriga ega «0». Agar «0» holatda bo‘lsa uzilishlar taqiqlanganligini bildiradi. Ba’zi hollarda uzilishlarni dasturdan chaqirish osonroq bo‘ladi.

Agar CALL buyrug‘i ishlatilsa, IRET buyrug‘i bayroqlar qiymatlarini yo‘q qiladi. Shuning uchun EHM o‘quv modelida INT n (modelda n $\in$ {0, 1,9}), buyrug‘i ishlatiladi. n –uzilish vektori.

Ishni bajarish

1) EHM o‘quv modelini ishga tushiring;

2) Belgilangan vazifadagi variantlar asosida TQ tanlang.

Tashqi qurilma(TQ) menyusidan Menedjer bo‘limini tanlang

Menyuni oching. «Подключаемые устройства», oynasi paydo bo'ladi. 5.10-chizma)

5.10-chizma. Tashqi qurilmani ulash oynasi

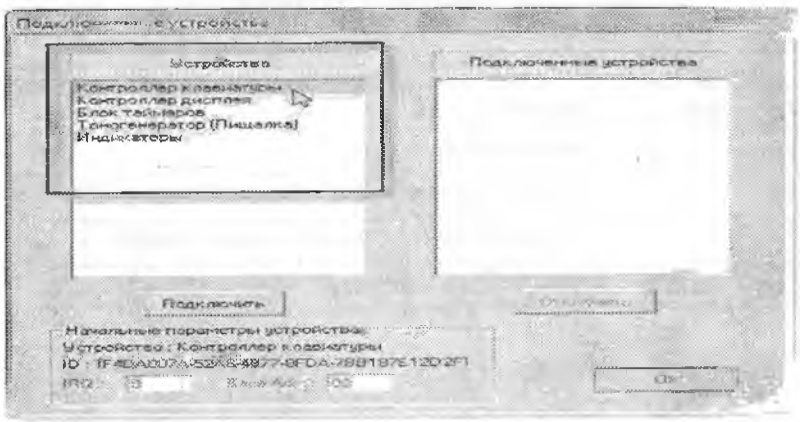
Oyna ikki bo'limga bo'lingan: Ustroystva (qurilma)-o'quv modelidagi tashqi qurilmalar; Podklyuchennыe ustroystva (biriktirilgan qurilmalar)-siz tomondan tanlangan qurilmalar.

Ustroystva(qurilma) tanlangandan so'ng Podklyuchit (ulanish) tasdiqi beriladi.

Masalan, ish jarayoni uchun klaviatura kerak bo'ladi:

-sichqonqaning chap tugmachasidan klaviaturani tanlang (5.11-chizma).

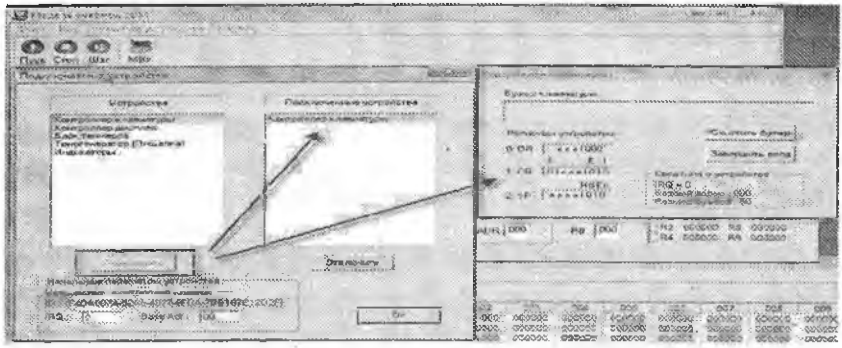
- Podklyuchit (ulanish) tugmachasini bosing Kontroller klaviaturi (klaviatura kontrolleri) oynasi paydo bo'ladi (5.12-chizma).



5.11-chizma. – Tashqi qurilmalarni tanlash

Klaviatura kontrolleri

Klaviatura kontrolleri tashqi qurilma hisoblanib, EHM klaviaturasidan ASCII-kodlarni qabul qiladi.



5.12-chizma. Klaviatura kontrolleri

50 simvolga teng o'rnatilgan simvollar bufer simvollariga ketma-ket joylashtiriladi. Simvollarni ketma-ket kiritilish jarayoni 5.8-jadvalda ko'rsatilgan.

1 adres bo'yicha kerakli kodlarni uzatish uchun (5.7-jadval)/ Klaviatura kontrolleri 4 ta buyruqni bajaradi.

Klaviatura kontrolleri buyruq kodlari

5.7- jadval

O'nlik son	Buyruq
101	buferni tozalash
102	Err bayrog'ini registr SR ga yuklash
103	registr CRga S bayroqni «1»ga o`rnatish
104	registr CRga S bayroqni «0»ga o'rnatish

Klaviaturadan simvollarni ketma-ket kiritish dasturi

5.8- jadval

Adres	Mnemokod	Izoh
000	RD #10	registr CR ga E bayrog'ini o'rnatish

5.8-jadvalning davomi

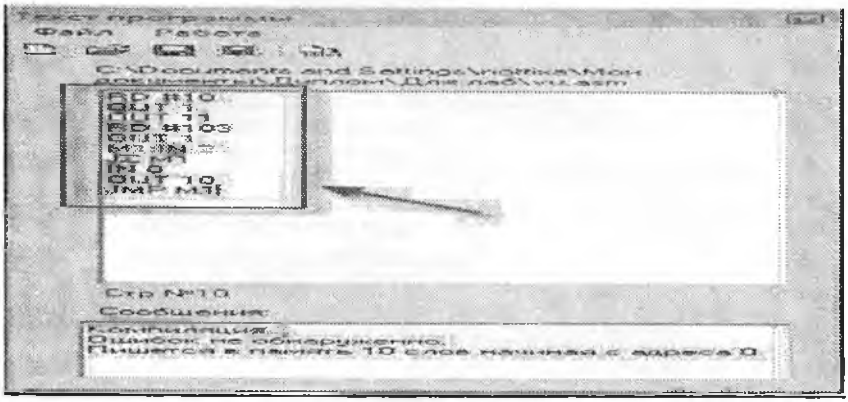
001	OUT 1	klaviaturani yoqish
002	OUT 11	displeyni yoqish
003	RD #103	kontrollerga komanda kodini uzatish
004	OUT 1	S ni «1» rejimga o'rnatish (ketma-ket kiritish)
005	M1: IN 2	Tugmacha bosilganligini tekshirish – tayyorlik barog'i Rdy
006	JZ M1	kutish Rdy = 1
007	IN 0	kiritilgan simvollarni buferda akkumulyatorga joylashtirish
008	OUT 10	ASCII-kodni displeyga chiqarish
009	JMP M1	kevingi tugmacha bosilganligini kutish

ASCII kodlari jadvali

5.9 -jadval

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p					А	Р	а	р
1			!	1	A	Q	a	q					Б	С	б	с
2			"	2	B	R	b	r					В	Т	в	т
3			#	3	C	S	c	s					Г	У	г	у
4			$	4	D	T	d	t					Д	Ф	д	ф
5			%	5	E	U	e	u					Е	Х	е	х
6			&	6	F	V	f	v					Ж	Ц	ж	ц
7			'	7	G	W	g	w					З	Ч	з	ч
8			(	8	H	X	h	x					И	Ш	и	ш
9			)	9	I	Y	i	y					Й	Щ	й	щ
A			*	:	J	Z	j	z					К	Ъ	к	ъ
B			+	;	K	[	k	{					Л	Ы	л	ы
C			,	<	L	\	l	|					М	Ь	м	ь
D			-	=	M	]	m	}					Н	Э	н	э
E			.	>	N	^	n	~					О	Ю	о	ю
F			/	?	O	_	o						П	Я	п	я

Qadam 1. 4.2-jadvalda keltitilgan dastur matnini kiriting (5.13-chizma).

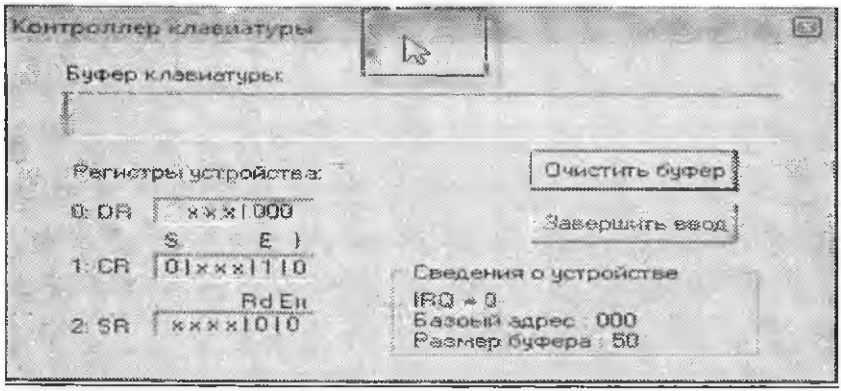


5.13-chizma. Dastur matni

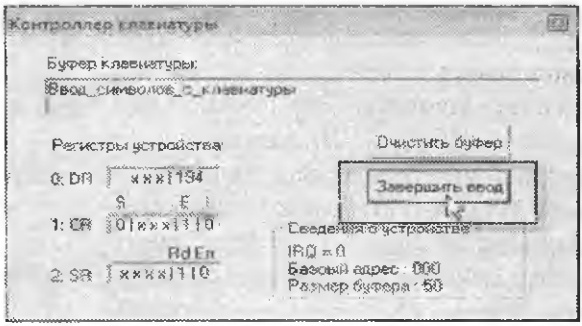
Qadam 2. Klaviatura kontrollerini yoqing.

Qadam 3. EHM modelini avtomatik rejimga o'tkazing Pusk tugmachasini bosib.

Qadam 4. Sichqoncha tugmachasi yordamida KK aktivlashtiring (5.14-chizma).



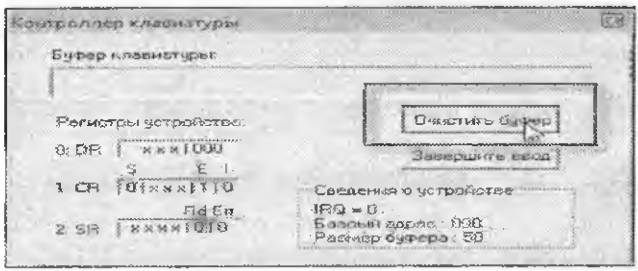
5.14-chizma. Klaviatura kontroller sharhlovchisi



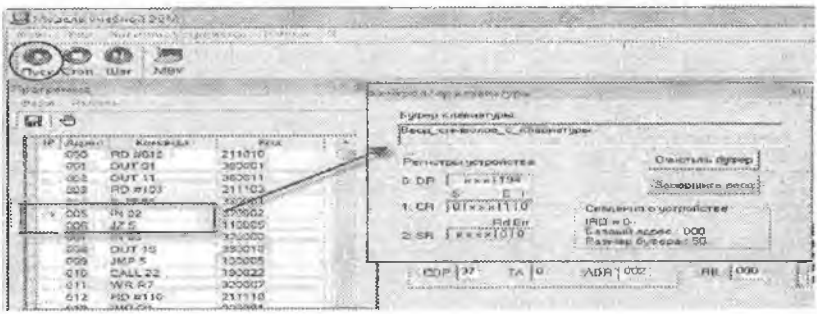
5.15-chizma. Klaviaturadan simvollarni kiritish

Qadam 5. Dastur bajarilishi. Klaviaturadan simvollarni kiritish jarayonida simvollarni klaviaturadan kiriting (5.15-chizma).

Qadam 6. Keraksiz simvollrni o'chirish uchun Ochistit tugmachasini bosish kerak (5.16-chizma).



5.16-chizma. Klaviatura buferini tozalash



·5.17-chizma. Dasturni ishga tushirish

Oadam 7. Klaviaturadan kiritishni to‘xtatish uchun Zavershit tugmachasini bosing.

Displey kontrolleri ma’lumotlar registriga kelib tushgan ASCII-kodlarnini yoritadi. Видеохотiraga «intefeys oynasi»orqali murojaat etiladi. AR registr adresiga Видеохотira yacheykasi raqamini yuklash kerak,shundagina ko‘rsatilagan adres bo‘yicha kiritish-chiqarish amalga oshiriladi.

1 adres bo‘yicha mos kodlarni chiqarishda displey kontrolleri 2ta buyruqni bajaradi (5.10-jadval).

Displey kontrolleri buyruq kodlari

5.10-jadval.

o‘nlik son	Buyruq
101	Displeyni tozalash(**Ochistit**), bunda Видеохотira AR adres registrini 000 holatga o‘rnatadi va Err bayrog‘ida xatolikni ko‘rsatadi
102	Err bayrog‘idagi xatolikni registr SR ga yuklash

«TATU» matnini displeyda chop etish dasturi *5.11-jadvalda* keltirilgan.

Displeyda chop etish dasturi

5.11-jadval

Adres	Mnemokod	Izoh
000	RD #11	Displeyni yoqish va o‘rnatish
001	OUT 11	Avtoinkrement bayrog‘i
002	RD #0	Boshlang‘ich adresni berish
003	OUT 13	Chiqariladigan simvol
004	RD #84	«T» xarf kodini kiritish
005	OUT 10	Displeyda chop etish
006	RD #65	«A» xarf kodini kiritish
007	OUT 10	Displeyda chop etish
008	RD #84	«T» xarf kodini kiritish
009	OUT 10	Displeyda chop etish
010	RD #85	«U» xarf kodini kiritish
011	OUT 10	Displeyda chop etish
010	HLT	To‘xtatish

Qator raqamini displeyda chop etish

5.12-jadval

Adres	Mnemokod	Izoh
000	RD #10	Yoqish
001	OUT 11	Displey
002	RD #0	Boshlang‘ich berish
003	WR R1	chiqarish adresi
004	RD #49	kodni kiritish
005	WR R2	raqam «1»
006	RD #8	sonni kiritish
007	WR R3	siklni takrorlash
008	M1: RD R1	navbatdagi adresni hisoblash
009	OUT 13	va displey adres registriga uzatish
010	ADD #16	adresni 16 ga oshirish –
011	WR R1	o‘zgartirilgan adresni saqlash
012	RD R2	raqam kodini hisoblash – qator raqami
013	OUT 10	raqam kodini displeyga chiqarish
014	RD @R2+	R2 tarkibini 1 ga oshirish.
015	JRNZ R3, M1	Dekrement R3 va sikl boshiga o‘tish, agar R3 ≠ 0
016	HLT	to‘xtatish

Qadam 1. 4.3-jadval da berilgan dasturni «Текст программи» oynasiga kiriting.

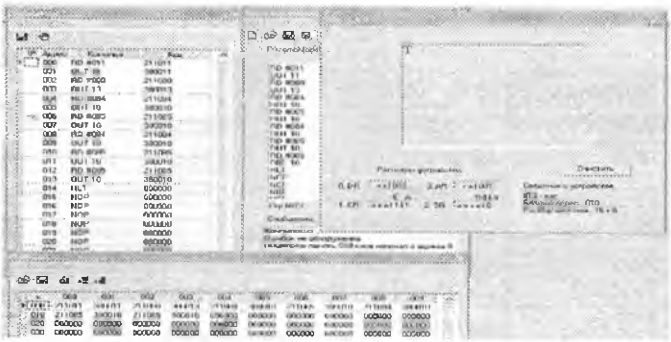
Qadam 2. TQ menedjeri orqali displey kontrollerini ulang.

Qadam 3. Shag rejimini RD #84 buyrug‘igacha bajaring (4.9-chizma).

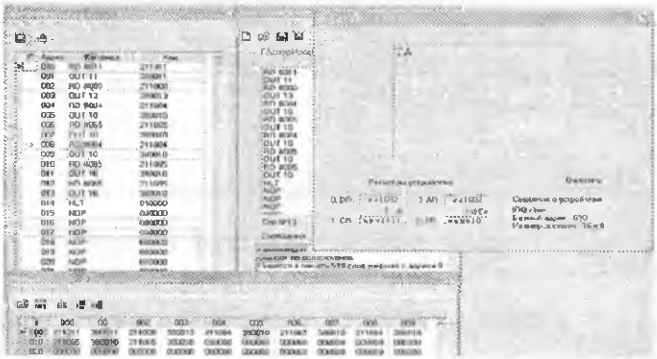
Qadam 4. Programma oynasida dastur bajarilish jarayonida ko‘rsatkich RD #65 buyrug‘iga kelishi bilan, displey «A» simvolni yoritadi (4.10-chizma).

Qadam 5. Programma oynasida dastur bajarilish jarayonida ko‘rsatkich RD #84 buyrug‘iga kelishi bilan, displey «T» simvolni yoritadi (4.10-chizma).

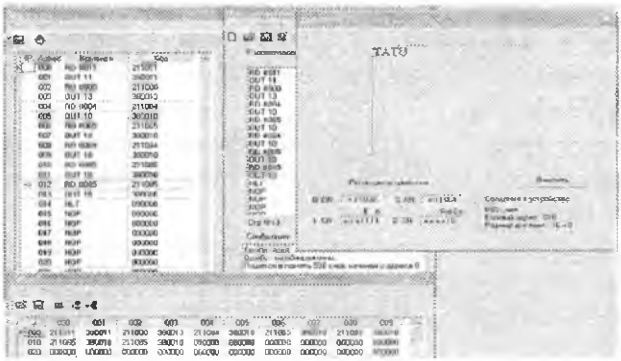
Qadam 6. Programma oynasida dastur bajarilish jarayonida ko‘rsatkich RD #85 burug‘iga kelishi bilan, displey «U» simvolni yoritadi (4.11-chizma) va displeyda.



5.17-chizma. Birinchi simvolni displeyga chiqarish



5.18-chizma. Simvol «TA» ni displeyda chop etish



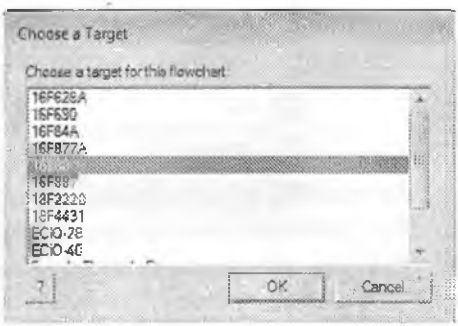
5.19-chizma. Simvol «TATU» ni displeyda chop etish

2-ILOVA

FLOWCODE V4 DASTURIY TA'MINOTI

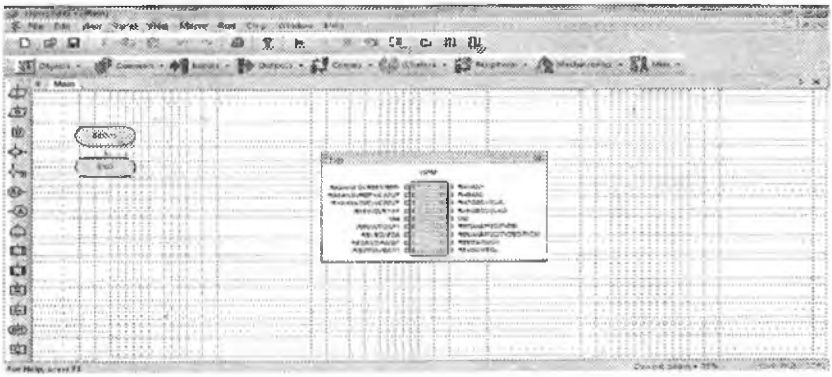
Mikrokontrollerni dasturlashda Flowcode V4 dasturiy ta'minoti keng ishlatiladi. Ushbu ilovada Flowcode V4 dasturiy ta'minoti foydalanishini ko'rib chiqamiz.

FlowCode V4 for PICmicros dasturini ishga tushiramiz va *File* menyusidan *New* bo'limini tanlaymiz yoki Ctrl + N tugmalarini bosamiz.



5.20-chizma. Mikrokontroller turini tanlash

Chizmada ko'rsatilgan ro'yxatdan 16F88 mikrokontrollerini tanlaymiz. Natijada quyidagi oyna hosil bo'ladi:



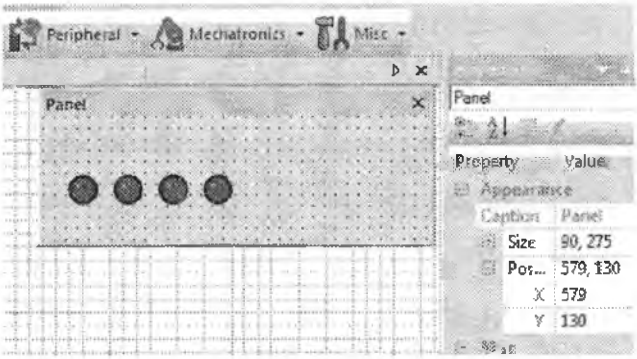
5.21-chizma. Loyihalash oynasi

Biz ishni svetodiodlarni joylashtirishdan boshlaymiz. Buning uchun avval, *Viev* menyusidan *Panel* bo'limini aktivlashtirish kerak.

Komponentlar panelidan (Components Toolbox) *Outputs* blokidan *LED* svetodiodini tanlab, panelga joylashtiramiz.

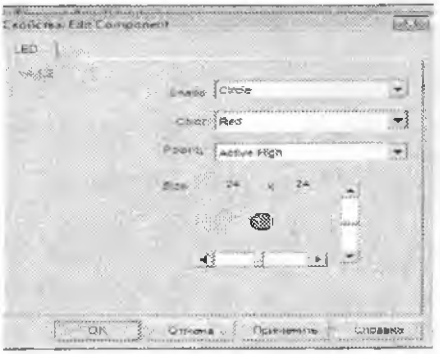
5.22-chizma. Componentlarni tanlash

Ushbu misolda 4 ta svetodiod tanlangan.



5.23-chizma. 4 ta svetodiod tanlash

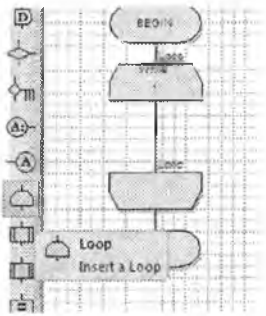
Svetodiod rangi, shaklini o'zgartirish mumkin. Buning uchun, *Ext Properties* bo'limini tanlanadi va kerakli parametrlardan foydalaniladi.



5.24-chizma. Svetodiodlarni moslashtirish

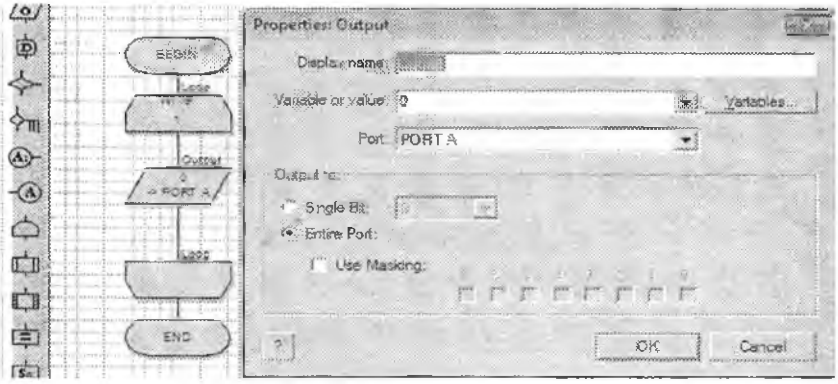
Endi bularni ketma-ket yonib o'chirish uchun buyruqlar ketma-ketligi-algoritmini tuzamiz. Buning uchun komandalar panelidan (Comand Toolbox) foydalanamiz.(5.25-chizma).

Avvalo jarayon, bir marotaba emas,b alki, takrorlanib turishi uchun blok sxemaga joylashtiramiz. Buning uchun komandalar panelidan *Loop(Insert a Loop)* blokini sichqoncha bilan belgilab, blok sxema ichiga joylashtiramiz:



5.25-chizma. Dastur algoritmi

Svetodiodlarni yonishi va o'chishini ta'minlash uchun *output* blokidan foydalanamiz. Komandalar panelidan *output* blokini sxemadagi sikl orasiga joylaymiz. Joylashtirilgan blokni sichqoncha yordamida ikki marta tanlaymiz va natijada quyidagi oyna hosil bo'ladi:



5.26-chizma. Moslashtirish oynasi

-Value bo'limiga 1 raqamini bersak blok svetodiodni yondirishga xizmat qiladi, agar 0 bersak o'chirishga xizmat qiladi.

-Biz foydalanayotgan 16F88 MK ikkita chiqish portiga ega (Port A, Potr B) bo'lib, ularning har biri 8 bitdan ma'lumot uzatish imkoniga ega. Svetodiod misol qilingani uchun faqat Port A ning to'rtta (0,1,2,3) bitini chiqishlaridan foydalanamiz.

-Birinchi *output* blokini diodini yonishi uchun sozlaymiz:

Varibli or value: 1

Port: Port A

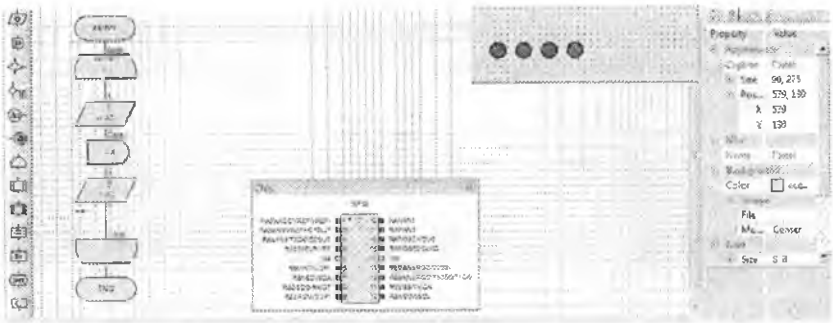
Output to: Single Bit: 0

OK

-Xuddi shu kabi *output* blokini diodni o'chirish uchun sozlaymiz, faqat bunda *value: 0*, bo'ladi.

-Bu holda bir qancha svetodiodimiz yonganini va qachon o'chganini sezmaymiz. Buni aniq ko'rish uchun yonish va o'chirish bloklari orasida vaqt kutilishini tashkil qilishimiz kerak. Bu ishni bajarish uchun yonish va o'chirish bloklari orasiga *Delay*blokini joylashtirib, uni 1 sek.ga sozlaymiz.

-Shu joylashtirilgan uch blok birinchi svetodiodni yonib, 1 sek turib yana o'chisrini ta'minlab beradi.



5.27-chizma. Mikrokontroller ish jarayonini moslashtirish

-Bizda to'rtta svetodiod borligi uchun bunday uchlik bloklar sonini yana uchtaga oshiramiz. Faqat bulardagi *output* bloklarida *Single Bit* bo'limini ketma-ket mos ravishda 1, 2, 3 qilib sozlaymiz.

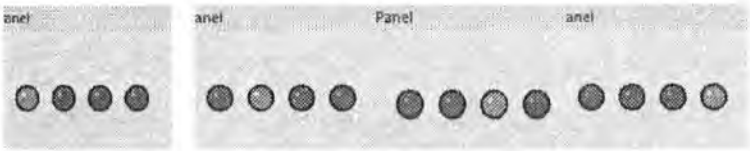
Endi svetodiodlarni tuzilgan algoritmga bog`laymiz, ya'ni mikrokontroller bilan aloqasini o'rnatamiz.

Buning uchun 1-svetodiodni tanlab, sichqonchaning chap tugmasini bosamiz va *Connections* bo'limini tanlaymiz.

5.28-chizma. Svetodiodlarni algoritmga bog'lash

Port: *Port A* va Bit: 0 ko'rinishida tanlab Done tugmasini bosamiz. Qolgan svetodiodlar ham shu kabi tanlanadi, faqat Bit: bo'limi mos ravishda 1, 2, 3 kabi tanlanadi.

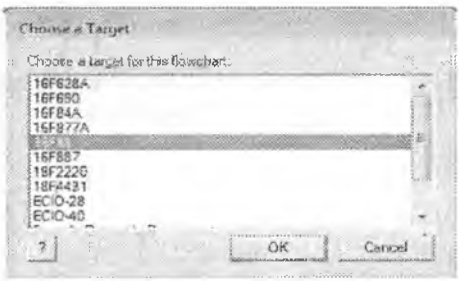
Mana endi dasturimizni saqlab ishga tushirishimiz mumkin, buning uchun Run menyusidan Go bo'limini yoki F5 klavishini bosiladi.



5.29-chizma. Svetodiodlar ish jarayoni

Flowcode virtual dasturi yordamida PIC 16F88 mikrokontrolleri asosida ishlovchi 7 segmentli svetodiodda klaviatura yordamida kiritilgan raqamni aks ettirish jarayonini hosil qilishni o'rganish

FlowCode V4 for PICmicros dasturini ishga tushiramiz va *File* menyusidan *New*bo'limini tanlaymiz yoki Ctrl + N tugmalarini bosamiz (5.30-chizma).

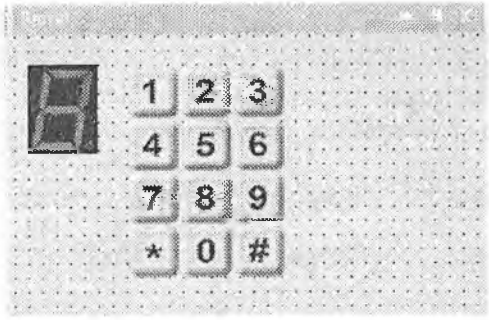


5.30-chizma. Mikrokontroller turini tanlash

Chimada ko'rsatilgan ro'yxatdan 16F88 mikrokontrollerini tanlaymiz.

Ishni 7 segmentli svetodiodni va klaviaturani joylashdan boshlaymiz.

Komponentlar panelidagi (Components Toolbox) *Outputs*blokidan *led7seg* svetodiodni tanlab, panelga joylashtiramiz. So'ngra yana shu paneldagi *Inputs* blokidan *KevPad* (klaviatura)ni joylashtiramiz (5.31-chizma).

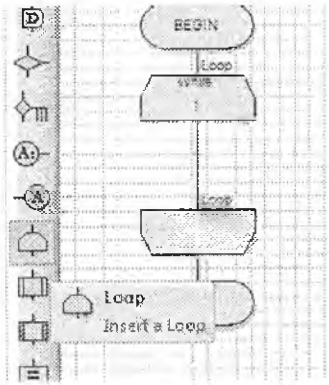


5.31-chizma. KeyPad va led7seg ni tanlash

Svetodiod va klaviatura ko'rinishini o'zingiz ixtiyoriy tanlashingiz mumkin. Buning uchun element ustiga sichqonchaning o'ng tomoni bosilib, *Ext Properties* bo'limi tanlanadi va kerakli parametrlar tanlanib, OK tugmasi bosiladi.

Endi ko'zlangan maqsadga erishish uchun buyruqlar ketma-ketligini – algoritmni tuzamiz. Bunda biz komandalar panelidan (Comand Toolbox) foydalanamiz.

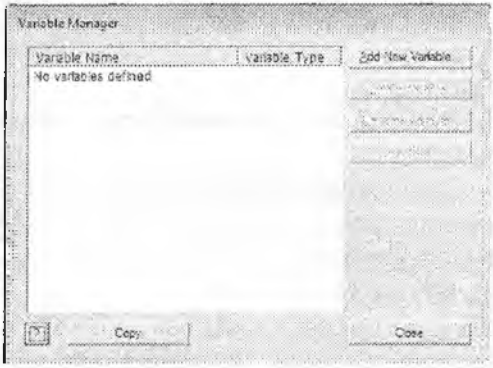
- Avvalo, jarayon bir marotaba emas, balki, takrorlanib turishi uchun blok-sxemaga siklni joylaymiz. Buning uchun komandalar panelidan *Loop (Insert a Loop)* blogini sichqoncha bilan ushlab, blok-sxema ichiga joylashtiramiz:



5.32-chizma. Algoritm yaratish oynasi

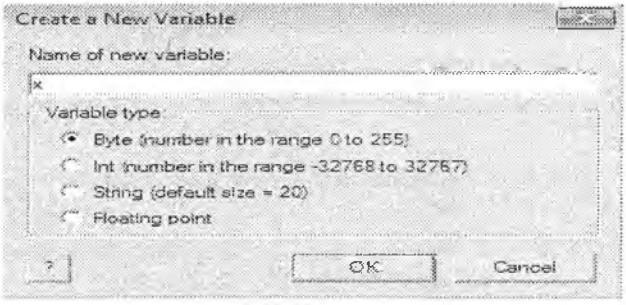
-Endi klaviaturadan kiritishni tashkil qilamiz. Buning uchun bizga bitta o‘zgaruvchi kerak bo‘ladi (5.33-chizma). Bu o‘zgaruvchi klaviaturadan kiritilgan raqamni qabul qilish uchun kerak bo‘ladi. Shuning uchun *ubyte*tipida bo‘ladi. Avvalo, o‘zgaruvchini e’lon qilib olamiz:

Edit menyusining Variables... bo‘limiga kiramiz.



5.33-chizma. O‘zgaruvchilar kiritishni sozlash

Add new variable...ni tanlaymiz.

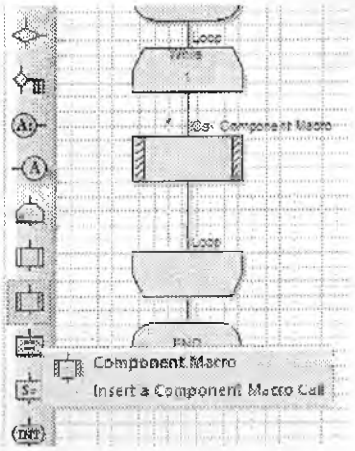


5.34-chizma. O‘zgaruvchi turini sozlash

Chizmada ko‘rsatilgani kabi sozlab OK tugmasini bosamiz. Keyin Close tugmasini bosamiz.

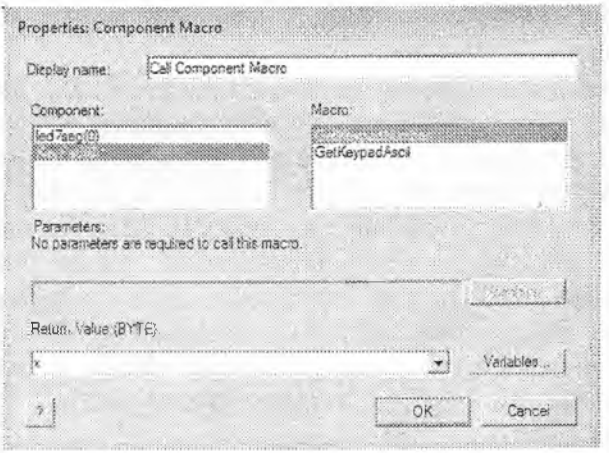
Klaviatura orqali kiritishni tashkil qilish:

Blok-sxemadagi sikl bloklari orasiga *Component Macro* blokini joylashtiramiz.



5.35-chizma. Klaviatura orqali kiritishni tashkil qilish

Joylashtirilgan blokka sichqonchani olib borib, chap tomonini ikki marta bosamiz va quyidagi oyna hosil bo'ladi:

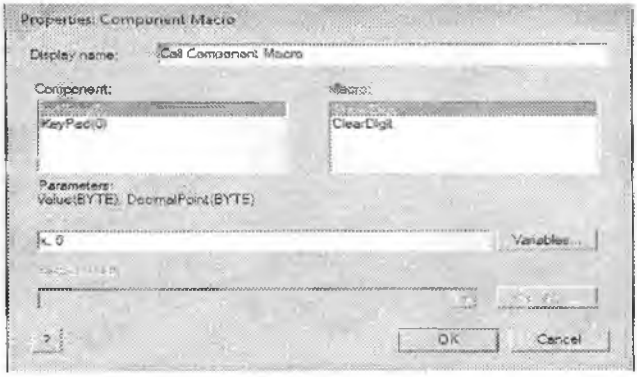


5.36-chizma. Tugmachalardan kiritishni sozlash

Ko‘rsatkichlarni rasmdagidek sozlab OK tugmasini bosamiz.

-Endi kiritilgan raqam – x ni 7 segmentli svetodiodda aks ettirishni tashkil qilamiz:

Yana bir *Component Macro* blokini sxemaga joylaymiz. Va uning ko‘rsatkichlari quyidagicha bo‘ladi:



5.37-chizma. led7seg aks etishni sozlash

Bu yerda x – aks ettiriladigan raqam. Verguldan keyingi *0* esa svetodioddagi nuqta bo‘lagi yonmasligini ta’minlaydi. Agar *0* o‘rnida *1* raqami bo‘lsa u holda nuqta ham yonadi (5.37-chizma).

OK tugmasini bosamiz.

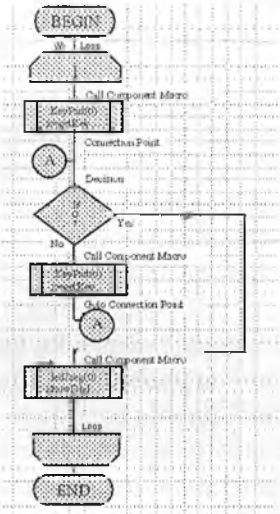
-Bu holda klaviaturadan qanday raqamni kiritmaylik svetodiodda faqat 8 raqami aks etadi. Chunki klaviatura ishlatilmagan holatlarda sikl bir necha marta aylanadi va klaviaturadan kiruvchi ma’lumotni olishga harakat qiladi. Ma’lumot yo‘qligi uchun o‘zgaruvchiga tasodifiy qiymatni saqlaydi. Svetodiod esa imkoniyatidan tashqari raqam kelganda barcha segmentlarini aktivlashtiradi. Bunday kamchilikni bartaraf qilish uchun quyidagi ketma-ketlikni tashkillashtiramiz.

Avvalo, biz shunday sikl tashkil qilishimiz kerakki, bu sikldan qachon klaviatura orqali raqam kiritilganda chiqsin va svetodiodni aktivlashtirsin. Bu siklni *Connection Point* bloklari yordamida bajaramiz.

Ikki *Component Macro* bloklari orasiga *Declare Connection Point* blokini joylashtiramiz.

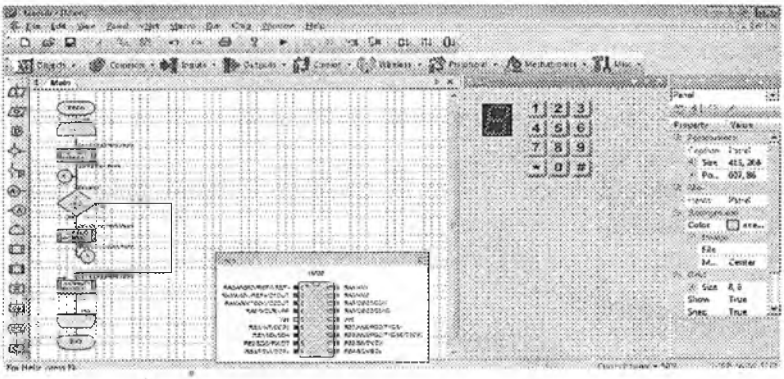
Uning ortidan *Decision (Shart)* blokini joylaymiz.

Joylashgan blokning chiziqli qismiga, ya'ni no tarmog`iga birinchi *Component Macro* blokini nusxasini joylaymiz va ketidan *Jump to Connection Point* blokini joylashtiramiz.



5.38-chizma. Dastur algoritmi

Shart blokini sozlash uchun uni sichqoncha bilan ikki marta tanlaymiz. Va if:bo'limiga x<=9 shartini kiritamiz va OK tugmasini bosamiz (5.39-chizma).



5.39-chizma. Dasturiy boshqariladigan qurilma va uning algoritmi

O‘quv platasida mikroposessor tizimlarini loyihalash

MK(mikrokontroller) – mikroprosessor tizimining soda ko‘rinishi bo‘lib, unda hamma yoki tizimni ko‘pchilik tugunlari bitta mikrosxema shaklida qilingan.

AVR – Atmel firmasining sakkiz bitli mikrokontrollerlari 1996 yilda yaratilgan.

AVR ni bir qancha kengaytmasi mavjud. Kimdir Advanced Virtual RISC (Kengaytirilgan virtual RISC) deb ta’kidlaydi, boshqalar AVRni yaratgan ikki insonni ismini bosh harflar olingan deyishadi Alf Egil Bogen Vegard Wollan RISC.

AVR mikrokontorllerlari Garvard arxitekturasiga ega (dastur va ma’lumotlar hotirasi alohida bo‘ladi) buyriqlar tizimi RISC g‘oyasiga yaqin. AVR protsessori 32 ta 8 – bitdan iborat umumiy foydalanishga mo‘ljallangan registirlarga ega, registrli fayllarga birlashtirilgan.

Ko‘pchilik buyruqlar 1 ta yacheyka xotirani egalaydi(16 bit).

Ko‘pchilik buyruqlar 1 davrda bajariladi.

Ularni quyidagi guruhlarga ajratish mumkin:

AVR mikrokontrollerlarini bir nechta guruhlarga ajratish mumkin:

- Mantiqiy operatsiyalar;
- Mantiqiy va surish operatsiyasi;
- Bitlar bilan ishlash;
- Uzatish operatsiyalari;
- Boshqarishni uzatish operatsiyasi;
- Tizimni boshqarish operatsiyasi.

Periferiya qurilmalarini boshqarish ma’lumot adres maydoni asosida boshqariladi. Qulaylik uchun qisqartirilgan IN/OUT buyruqlardan foydalaniladi.

#include<avr/io.h> //Sarlavhali faylni ulash (MK ni registirlarini fizik manzillarini ro‘yxati)

#include<util/delay.h> // Sarlavhali faylni ulash (_delay_ms() va delay_us()funksiyalarini yozilmasi)

intmain(void) //Asosiy funktsiya (dasturboshi)

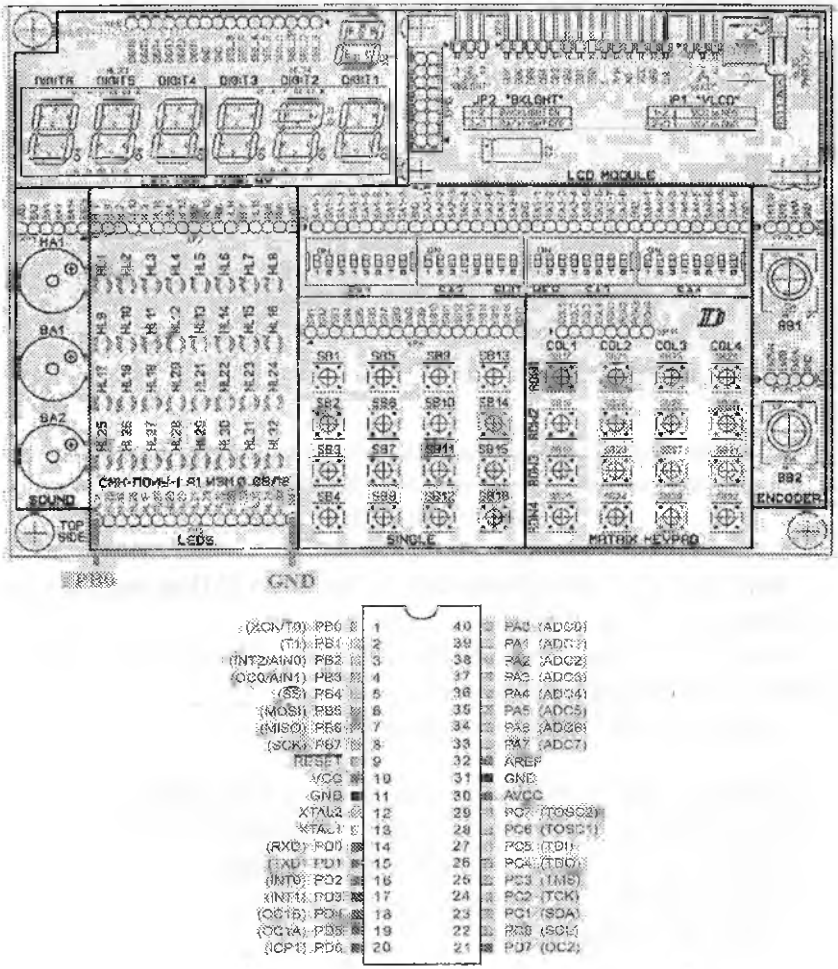
{

DDRB= 0xFF; // PORTBni chiqish deb e’lonq ilamiz

PORTB=0x00; //Bportda 0[V] deb o‘rnatamiz

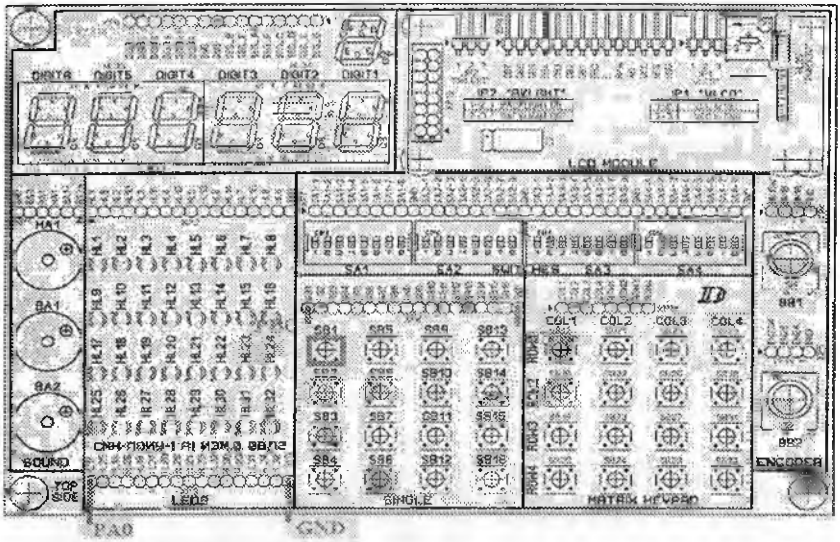
while(1) // Uzluksiz davr

```
{
PORTB = 0x01; //PORTB0ga  5[V]berilsin
_delay_ms(50); //PORTB0 50 ms yoqib turilsin
PORTB = 0x00; //PORTB0ga  0[V]berilsin
_delay_ms(50); //PORTB0 50 ms o‘chirib turilsin
}
}
```



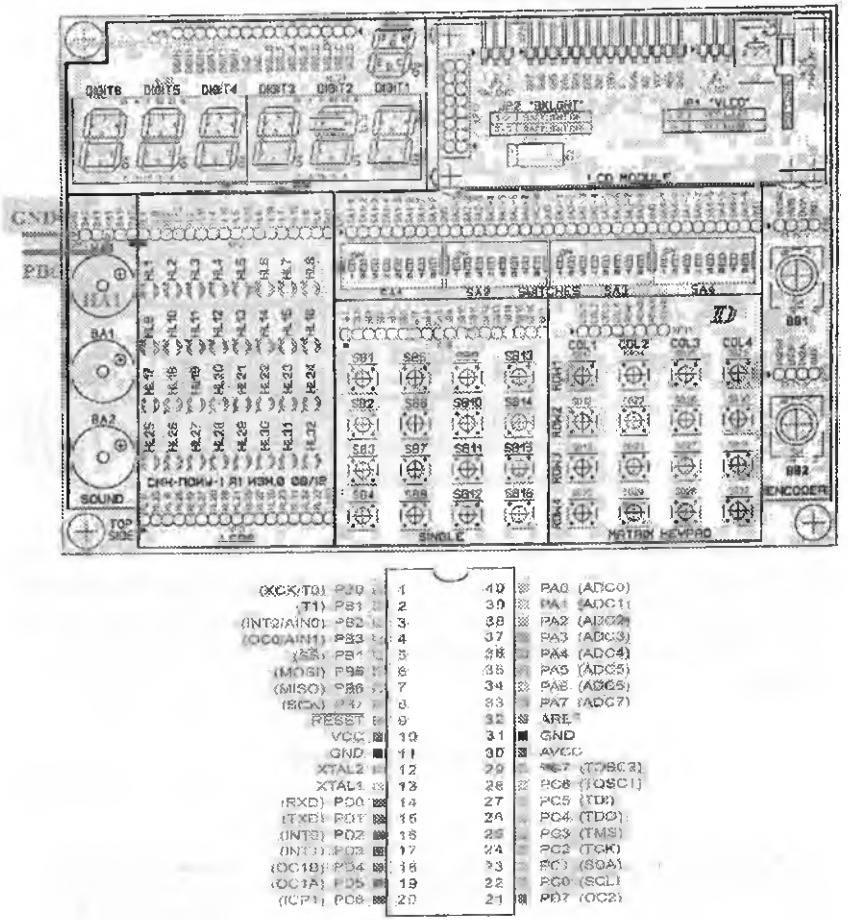
5.40-chizma. O‘quv platasining umumiy ko‘rinishi

Sozlash platasi asosida mikrokontrollerni tugmacha va svetodiodiga mantiqiy nol berish orqali mikrokontorllerni boshqa elektrodini yoqish.



5.41-chizma. Sozlash platasi asosida mikrokontrollerni tugmacha va svetodiodiga mantiqiy nol berish orqali mikrokontorllerni boshqa elektrodini yoqish

```
#include<avr/io.h> //Sarlavhali faylni ulash (MKni registrlarini fizik manzillarining ro'yxati)
#include<util/delay.h> // Sarlavhali faylni ulash (_delay_ms() va _delay_us()funksiyalarini yozilmasi)
 intmain(void) //Asosiy funktsiya (dastur boshi)
{
DDRA= 0xFF; // PORTAni chiqish deb e'lon qilamiz
PORTA=0x00; //Aportda 0[V] deb o'rnatamiz
DDRB= 0x00; // PORTBni kirish deb e'lonqilamiz
PORTB=0xFF; //Bportda1[V] deb o'rnatamiz
while(1) // Uzluksiz davr
{
PORTA = ~ PORTB; //~ belgi razryadli invertirlash
```



5.42-chizma. Sozlash platasi asosida mikrokontrollerni bitta elektrodi orqali tovushni hosil qilish.

#include<avr/io.h> //Sarlavhali faylni ulash (Mkni registirlarini fizik manzillarini ro'yxati).

```
#include<util/delay.h> // Sarlavhali faylni ulash (_delay_ms() va
_delay_us()funksiyalarini yozilmasi)
 intmain(void) //Asosiyfunktsiya (dasturboshi)
{
DDRB= 0xFF; // PORTBni chiqish deb e'lonqilamiz
PORTB=0x00; //Bportda 0[V] deb o'rnatamiz
 while(1) // Uzluksiz davr
{
PORTB = 0x01; //PORTB0ga  5[V]berilsin
_delay_ms(50); //PORTB0 50 ms yoqib turilsin
PORTB = 0x00; //PORTB0ga  0[V]berilsin
_delay_ms(50); //PORTB0 50 ms o'chiq turilsin
}
}
```

Embedded Arduino board
LCD displey asosida tizimlarni loyihalash

Arduino IDE muhitida LCD-displeya Hitachi HD44780 mavjud. LCD displey tiniqligi boshqaruv kirishiga berilayotgan kuchlanish o'lchamiga bog'liq. Kuchlanish 0,5-1 V oralig'ida bo'lishi kerak, lekin muhit darajasiga ham bog'liqdir 1.1-jadvalda LCD-Arduino pinauti keltirilgan.

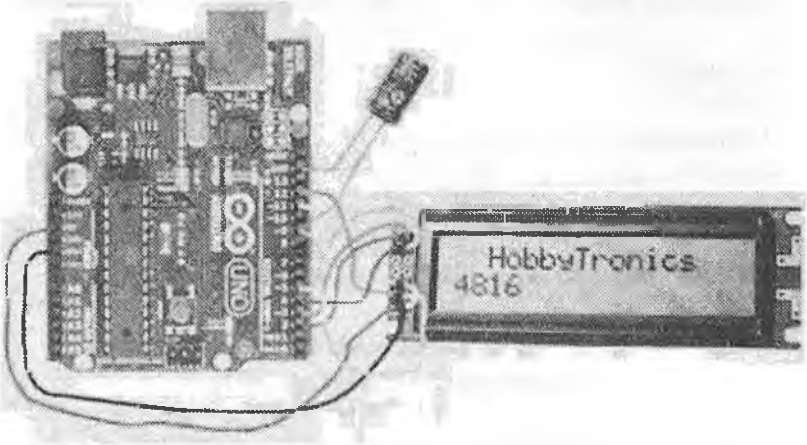
LCD-Arduino pinauti

1.1-jadval

LCD Pin	Символ	Назначение	Arduino Pin
1	Vss	Obshiy (0 V)	Obshiy (0 V)
2	Vdd	Pitanie (4.5 – 5.5 V)	+5V
3	Vo	Uprav.kontrastnostyu	9
4	RS	H/L register select signal	12
5	R/W	H/L read/write signal	Obshiy (0 V)
6	E	H/L enable signal	11
7	DB4	H/L data bus for 4-bit mode	5

I.1-jadvalning davomi

8	DB5	H/L data bus for 4--bit mode	4
9	DB6	H/L data bus for 4-bit mode	3
10	DB7	H/L data bus for 4-bit mode	2



5.1-rasm. Ulanish jarayoni

Liquid Crystal kutubxonasi barcha LCD Hitachi HD44780 bilan ishlaydi.

Ulanish sxemasi:

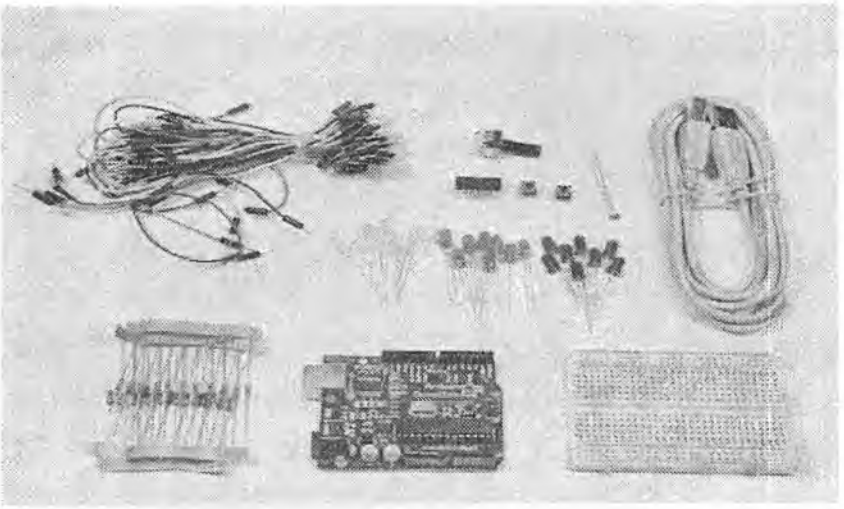
```
 * LCD RS pin raqamli chiqish 12
 * LCD Enable pin raqamli chiqish 11
 * LCD R/W  umumiy
 * LCD VO pin (pin 3)  ShIMga-chiqish 9
 * LCD D4 pin raqamli chiqish 5
 * LCD D5 pin raqamli chiqish 4
 * LCD D6 pin raqamli chiqish 3
 * LCD D7 pin raqamli chiqish 2
 */
#include // kutubxonani ulaymiz
Liquid Crystallcd (12, 11, 5, 4, 3, 2);        // initsializatsiyalash
```

```
Void setup()
 {
pinMode(9, OUTPUT);
analogWrite(9, 50);    // ShIM chiqishga o‘rnatish
lcd.begin(16, 2);      // qator va ustunlar miqdorini o‘rnatish
lcd.print("  HobbyTronics"); //ma’lumotlarni LCD ga uzatish
}
voidloop()
{
 // kursorni v 0-chi ustunga, 1 qator ( 0 dan boshlanadi)
lcd.setCursor(0, 1);
lcd.print(millis()/1000);  // qayta yuklashdan so‘ng sekund chop
etish
}
```

O‘rnatilgan ma’lumot uzatish tizimini loyihalash (Interfeyslar USART va SPI)

Arduino — apparat hisoblash platformasi hisoblanib, asosiy komponentlari oddiy kiritish/chiqarish platasi va Processing/Wiring dasturiy tilni qo‘llaydigan dasturiy muhit hisoblanadi. Arduino avtonom interaktiv obyektlarni yaratishda to‘g‘ridan to‘g‘ri kompyuterga ulangan holda dasturlash jarayonini amalga oshirish mumkin bo‘lgan qurilma hisoblanadi (masalan, Macromedia Flash, Processing, Max/MSP, Pure Data, Super Collider). Mikrokontroller Atmega 328 arzon va qo‘llanishga qulay hisoblanadi.

Arduino — elektron konstruktor va foydalanuvchi uchun elektron qurilmalarni yaratish mumkin bo‘lgan platforma hisoblanadi. Platforma jahonda qulayligi va dasturiy tillarning oddiyligi uchun keng qo‘llaniladi. Qurilma programmatorsiz USB orqali dasturlanadi. Platadagi mikrokontroller Wiring dasturiy tilga asoslangan Arduino tilida dasturlanadi. Arduino platformasi apparat qismi Arduino platformasining bir necha versiyalari mavjud bo‘lib, ulardan Leonardo AT mega 32u4 mikrokontrolleri bazasi asosida yaratilgan. Uno esa Atmel AT mega 328 mikrokontrolleri asosida yaratilgan.



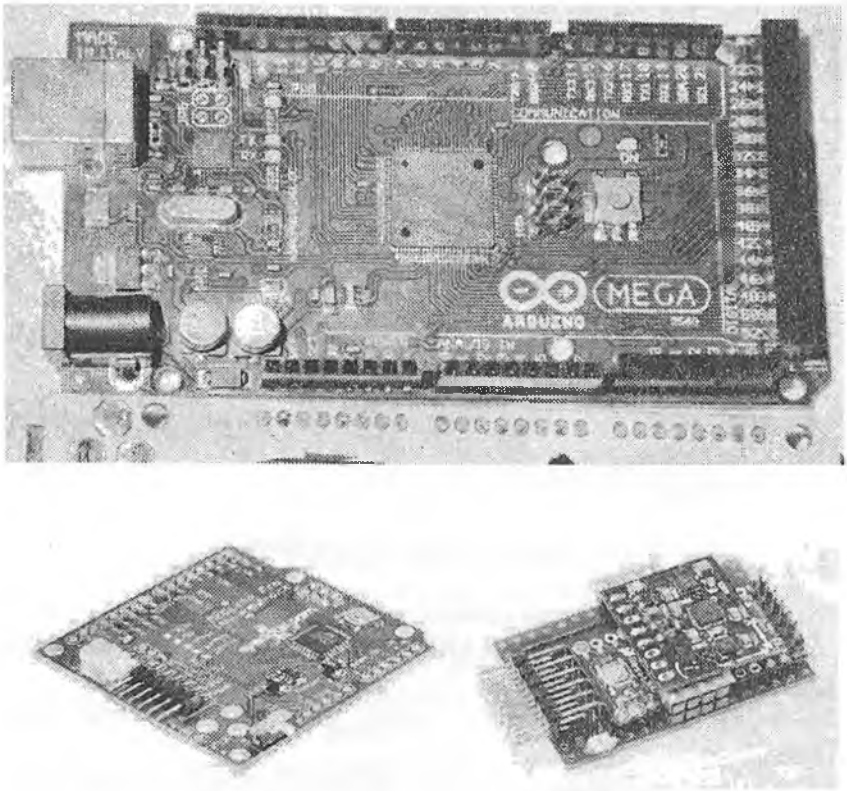
5.2-rasm. Arduino platformasi

Arduino platformasi versiyalari:

- Due — yangi plata ARM mikroprotsessori asosida 32bit Cortex-M3 ARM SAM3U4E.
- Leonardo — Arduno yangi platforma ATmega 32u4 mikrokontroller asosida .
- Yun - yangi plata, o'rnatilgan WiFi ni qo'llaydi ATmega 32u4 and the Atheros AR9331 asosida.
- Micro — AT mega32u4 asosidagi yangi loyiha .
- Uno —Arduino USB ko'p foydalaniladigan versiya. Uni o'zida standart port USB saqlaydi. Platforma o'z navbatida kengaytirilgan platalar va foydalanuvchi turli funksiyalari bilan to'ldirilishi mumkin.
- Nano — kompakt platforma hisoblanib, maket ko'rinishida foydalaniladi. Nano kompyuterga USB Mini-B kabeli asosida ulanadi.

Kengaytirilgan platalar asosan turli qurilmalarni boshqarishda va ma'lumotlar olishda foydalaniladi:

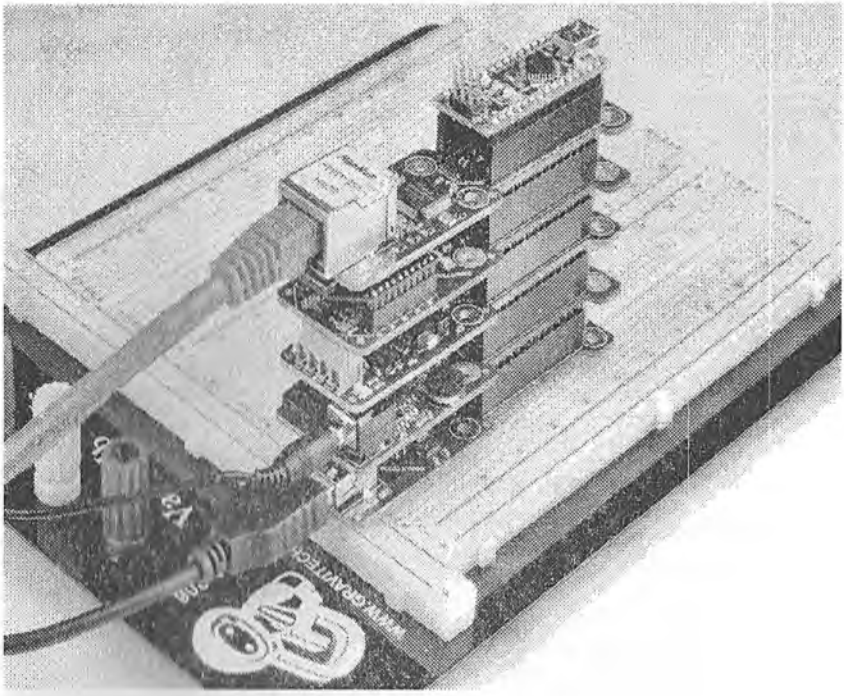
- Plata WiFi - 802.11 b/g simsiz tarmoq standarti bilan ulanishda foydalaniladi.



5.3-rasm. Arduino platasi umumiy ko'rinishi

Plata Xbee Shield - Maxstream Xbee Zigbee modullari asosida bir necha Arduino bilan 35 metr radiusdagi maydonni va 90 metr maydondan tashqari simsiz aloqani ta'minlaydi.

- Plata Motor Shield – doimiy tok dvigatellarini boshqarish va datchiklar holatini o'qishda foydalaniladi.
- Plata Ethernet Shield – internetga ulanishni ta'minlaydi.
- Keltiriladigan masalalarda svetodiodlarni boshqarishda murakkab dasturlash jarayonlari orqali ko'rish mumkin.
- Ishni bajarish uchun 5 ta svetodiod (turli rangli) va 5 rezistor (5.4-rasm).

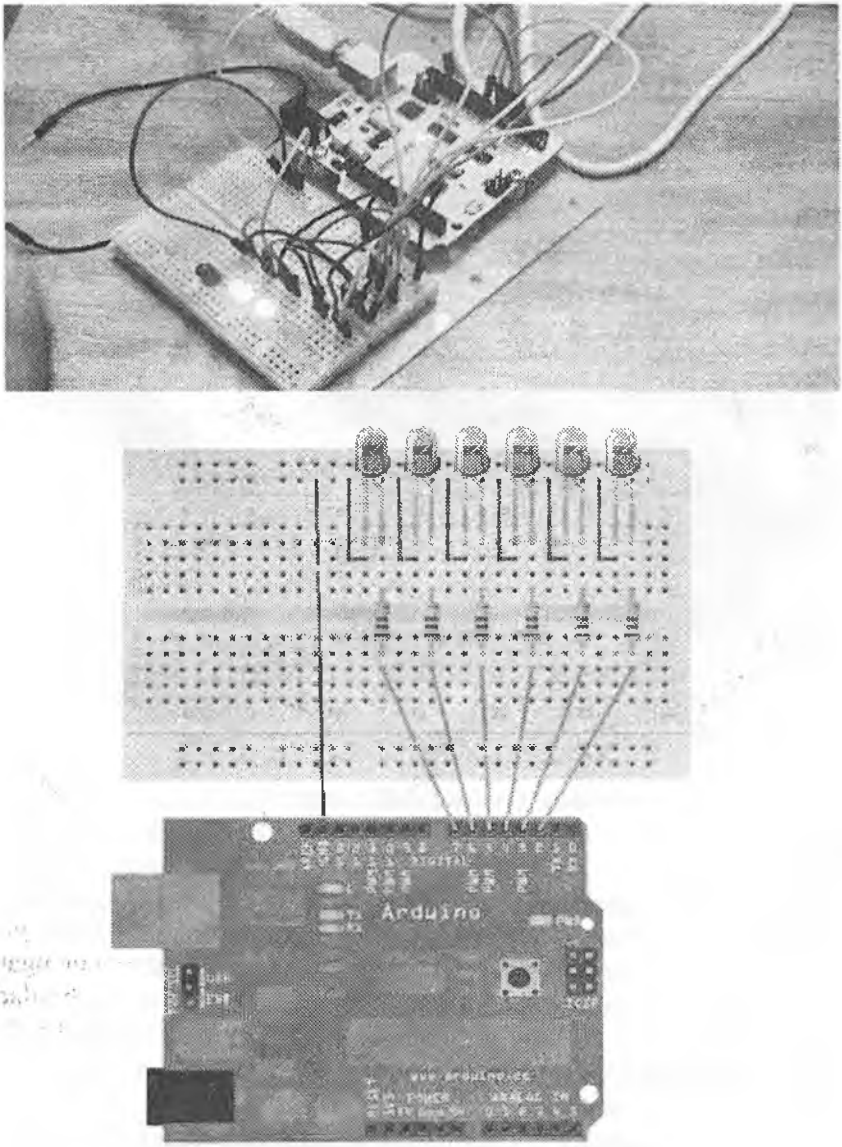


5.4-rasm. Plata Ethernet Shield

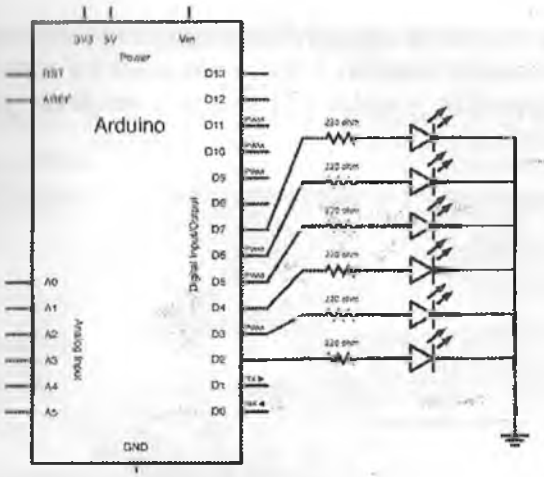
Keltirilgan sxemada 6 svetodiodlar raqamli pinlarga 2-7 ulangan (можно подключить 5 светодиодов к пинам 4-8 pinlarga ulangan. Ushbu ulanish pin (digitalWrite) dastur asosida kuchlanish beriladi va tok rezistor asosida oqib o'tib svetodiod anod (+) ga uzatiladi . Elektrik tok svetodiod yorug'lik energiyasiga aylanadi (fotonlar) , shuning uchun katod (—) yerga (Ground) ulanishi kerak.

Mavjud 6 svetodioddan , 1- chini pin №2 ga (yoki pin №4 ga), 2-chi svetodiod pin №3 ga (yoki pin №5 ga), 3-chi pin №4 ga (yoki №6 ga), 4-chi pin №5 ga (yoki pin №7 ga), 5-chi pin №6 ga (yoki pin №8 ga) va 6-chi svetodiod pin №7 ga.

Qurilmani dasturiy ta'minoti Arduino IDE muhitida bajariladi va Servis menyusida Monitor porti mavjud (Ctrl+Shift+M).



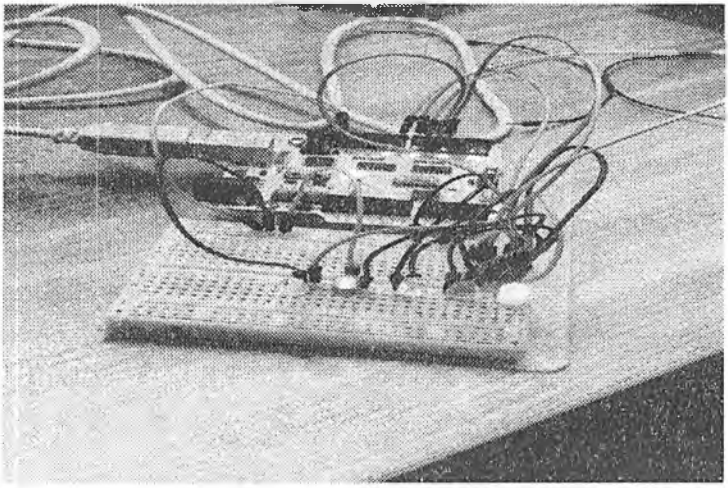
5.5-rasm. Arduino ga ulangan qurilmalarni boshqarish



5.43-chizma. Arduino ga ulangan qurilmalarni boshqarish prinsipial sxemasi

5.44-chizma. Monitor porti

COM-port asosida ulangan Arduinoga kompyuter orqali ixtiyoriy simvollarni uzatish mumkin. Uzatish natijasida ba'zi bir harakatlarni boshqarish mumkin, masalan «1» raqami svetodiodni yoqish va «0» raqami svetodiodni o'chirish vazifasini bajarladi.



5.6-rasm. Arduino platasiga ulanish

Uzatilgan simvollar to'plami svetodiodlarni boshqaradi.

Monitor COM-porta orqali Arduino IDE ga uzatilgan masalan «10110» qator svetodiodlar №1, 3 va 4 yoqadi va svetodiodlar №2, №5 (pozitsiyaga nisbatan) o'chiradi.

Dastur qismida klass Si – Serial dan foydalaniladi:

- Serial.begin – portdan (COM-porta) foydalanishga tayyorlik;
- Serial.print – ma'lumotlarni Arduino dan kompyuterga uzatish va Monitor portda aks ettirish;
- Serial.read – Arduino yordamida ma'lumotlarni kompyuterdan o'qish (masalan: «10110» qatorni o'qish). Serial funksiyasi to'plami Arduino bilan kompyuter orasidagi aloqachi vazifasini bajaradi.

Barcha Arduino platalari hech bo'lmaganda bitta ketma-ket port-ga ega port UART yoki USART. Serial ma'lumot almashish uchun raqamli k\ch portlardan

0 (RX) va 1 (TX), USB foydalaniladi.

ADABIYOTLAR

1. Майоров С.А, Кириллов В.А, Приблуда А.А. Введение в микро ЭВМ. –Л.: Машиностроение. Ленингр. отд-ние, 1988, 304с.
2. Жмакин А.П. Архитектура ЭВМ. -Петербург, -СПб.: БХВ 2006, 320 с.
3. Преснухин Л.Н. Микропроцессоры. В 3 кн. Кн. 2: Средства сопряжения. Контролирующие и управляющие системы. Учебник для техн-х вузов /В.Д. Вернер, Н.В. Воробьев, А.В. Горячев и др.– М.: Выш.шк., 1987, 303 с.
4. Максимов Н.В, Партыка Т.Л., Попов И.И. Архитектура ЭВМ и вычислительных систем. Учебник. –М.: ФОРУМ; ИНФРА-М, 2005, 512 с.
5. Пескова С.А., Кузин А.В. Архитектура ЭВМ и вычислительных систем. Учебник. –М.: ФОРУМ; ИНФРА-М, 2006, 352 с.
6. Цилькер Б.Я. Организация ЭВМ и систем / Б.Я. Цилькер, С.А. Орлов. –СПб.: Питер, 2007, 672 с.
7. Таненбаум Э. Архитектура компьютера. 5-е изд. – СПб.: Питер, 2007, 844 с.
8. Yunusov J.Yu., Abasxonova X.Yu. Raqamli qurilmalar va mikroprosessor tizimlari. O‘quv qo‘llanma –Toshkent: Iqtisod, 2010, 256 v.
9. Зарубин А.А. Микропроцессорное программное управления. Архитектура ІХА. Методические рекомендации к практическим занятиям. СПбГУТ.- СПб, 2004.

10. Калабеков Б.А. Цифровые устройства и микропроцессорные системы. –М.: Горячая линия-Телеком., 2003, 336 с.
11. Гребешков А.Ю. Микропроцессорные системы и программное обеспечение в средствах связи: Учеб. пособие. – Самара: ПГУТИ, 2009, 298 с.
12. Новиков Ю.В., Скоробогатов П.К. Основы микропроцессорной техники. - М.: ИНТУИТ, 2010, 440 с.
13. Кустарев П.В. Специализированные процессоры. Процессоры для встраиваемых приложений: Конспект лекций.-СПб.: СПбГИМО(ТУ), 2002, 30 с.
14. Белов А.В. Создаем устройства на микроконтроллерах.-СПб.: Наука и техника, 2007, 304 с.
15. Хартов В.Я. Микроконтроллеры AVR. Практикум для начинающих.- М.: МГТУ им. Н.Э.Баумана, 2007, 240 с.
16. Голубцов М.С. Микроконтроллеры AVR: от простого к сложному.- М.: СОЛОН-Пресс, 2003, 288 с.
17. Васильев А.Е. Микроконтроллеры. Разработка встраиваемых приложений: Учеб. пособие. – СПб: СПбГПУ, 2003, 210 с.
18. Грищенко В.И., Ладыженский, Моатаз Юнис. Основные направления развития современных сетевых процессоров. ДонТУ, Информатика, кибернетика и вычислительная техника, №14, 2011, 123-127с.
19. Грищенко В.И., Ладыженский Ю.В. Моделирование маршрутизаторов на многоядерных сетевых процессорах. ДонТУ, Информатика, кибернетика и вычислительная техника, №12, 2010, 169-176с.

20. Моатаз Юнис, Грищенко В.И., Ладыженский Ю.В Обобщенная архитектура сетевого процессора. – ДонТУ, Информатика и компьютерные технологии-2011, 386-391с.
21. Сетевые процессоры фирмы Intel, Компоненты и технологии, №8, 2003.
22. Грищенко В.И., Ладыженский, Моатаз Юнис Перспективные архитектуры и тенденции развития современных сетевых процессоров, 4 Международная научно-техническая конференция « Моделировани и компьютерная графика», – ДонТУ, 2011, 93-9с.
23. ARM System Developer's Guide: Designing and Optimizing System Software. Dominic Sums and Chris Wright. The Morgan Kaufmann Series in Computer..., Apr 8, 2004.
24. Computers as Components, Third Edition: Principles of Embedded Computing System Design. Marilyn Wolf. The Morgan Kaufmann Series... ,May 23, 2012
25. PIC Microcontroller. Muhammad Ali Mazidi, Rolin D. McKinlay and Danny Causey, Feb 16, 2007.

MUNDARIJA

СОДЕРЖАНИЕ

4. МИКРОКОНТРОЛЛЕРНЫЕ СИСТЕМЫ

CONTENTS

4. MICROCONTROLLER SYSTEMS

QAYDLAR UCHUN

X.YU.ABASXANOVA, U.B.AMIRSAIDOV

MIKROPROTSESSORLAR

Toshkent – «Fan va texnologiya» – 2016

Muharrir:	F.Ismoilova
Tex. muharrir:	M.Holmuhamedov
Musavvir:	D.Azizov
Musahhih:	N.Hasanova
Kompyuterda sahifalovchi:	Sh.Mirqosimova

E-mail: tipografiyacnt@mail.ru Tel: 245-57-63, 245-61-61.
Nashr.lits. AI№149, 14.08.09. Bosishga ruxsat etildi: 18.11.2016.
Bichimi 60x84 $^{1}/_{16}$. «Timez Uz» garniturasi.
Ofset bosma usulida bosildi.
Shartli bosma tabog'i 16,75. Nashriyot bosma tabog'i 17,0.
Tiraji 300. Buyurtma №232.

«Fan va texnologiyalar Markazining
bosmaxonasi» da chop etildi.
100066, Toshkent sh., Olmazor ko‘chasi, 171-uy.

ISBN 978-9943-11-342-8