

6-MAVZU

MP ASOSIY BUYRUQLAR SISTEMASI

Reja

- 6.1 Bazali buyruqlar sistemasi va uni guruhlarga ajratish.
- 6.2 Arifmetik, logik, yuklash (o'tkazish), boshqaruvchi, shartli va shartsiz, o'tish.
- 6.3 Boshqa buyruqlar guruhi to'g'risida tushuncha.
- 6.4 Buyruqlarning, amallar kodining bajarilishiga misollar.

MK protsessorlarining buyruqlar sistemasi.

Har qanday mikroprotsessor sistemasi kabi, MK protsessorining buyruqlar to'plami o'ziga 4 ta asosiy buyruqlar guruhini kiritadi:

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

Ko'pchilik takomillashgan MK larda portning registrlarini razryadlaridan mustaqil foydalanish uchun buyruqlarni bitli boshqarish guruhi ko'zlangan (bulli yoki bitli protsessor). Bitli protsessor buyruqlarining borligi programmalarining boshqarish kodlari hajmini va ularning bajarilish vaqtlarini qisqartirishga ruxsat beradi. MK qatorida kontrollerning resurslarini boshqarish buyruqlari guruhi ajratiladi, ular kiritish/chiqarish portlarini ishlash rejimlarini sozlash, taymemi boshqarish va boshqalar uchun ishlatiladi. MK larda kontrollerning ichki resurslari ma'lumotlar xotirasida akslantiriladi, shuning uchun resurslarni boshqarish maqsadida ma'lumotlarni yuklash buyruqiari sistemasi bilan taqqoslaganda arifmetik va mantiqiy buyruqlarning nisbatan kamroq takomillashgan guruhlarga ega ekanligi, ammo ma'lumotlarni yuklovchi va boshqaruv buyruqlarining nisbatan kuchliroq guruhlarga ega ekanligini ko'rishimiz mumkin. Bu xususiyat MK ning ishlatilish sohalariga bog'liq, u 1-navbatda o'rab turgan atrof muhitni boshqarib turishni va boshqaruvchi ta'sirini tashkil qilishni talab qiladi.

K580VM80A mikroprotsessori komandalar sistemasi 78 turdagi komandalardan iborat bo'lib, vazifasiga ko'ra ularning uzunligi bir, ikki yoki uch baytni tashkil etishi mumkin. Programma schetchigi xar doim komandaning birinchi bayti adresini o'zida saqlaydi. Ikki baytli komandalarda ikkinchi bayt 8 razryadli ma'lumotni yoki kirish/chikish interfeysining porti adresini, uch baytli komandalarda esa ikkinchi va uchinchi bayt 16 razryadli ma'lumotni yoki xotira yacheykasining adresini kursatishi mumkin (ma'lumotlar va adreslar 16-lik sanoq sistemasida yoziladi).

Komandalarga misollar:

- bir baytli komandalar: MOV A, B; LDAX B; RST 7; RAL;
- ikki baytli komandalar: MVI M, 85; SUI 8E; IN 21; OUT 3A;
- uch baytli komandalar: LDA 1234; LXI B, 45AE; CALL A34C; JC B800.

Komandalar sistemasini 5 guruh komanandalariga ajratish mumkin:

- ma`lumotlarni uzatish komandalari (14 ta komanda);
- mantiqiy komandalar (15 ta komanda);
- arifmetik komandalar (14 ta komanda);
- boshqarishni uzatish komandalari (28 ta komanda);
- boshqarish komandalari (7 ta komanda).

Barcha komandalar quyidagi 5 ta jadvalda keltirilgan bo`lib, ularda:

Ri va Rk - qabul qiluvchi va uzatuvchi registrlar (B, C, D, E, H, L, xamda adresi HL juftlikda ko`rsatilgan xotira yacheykasi – M); data - 8 razryadli ma`lumot; data 16 - 16 razryadli ma`lumot; addr - 16 razryadli adres; R - 8 razryadli registr (B, C, D, E, H, L, xamda adresi HL juftlikda ko`rsatilgan xotira yacheykasi – M); 2R - registr juftliklari (V, D, H, ayrim xollarda SP, PC); (XX) - adresi XX bo`lgan xotira yacheykasidagi ma`lumot; port - interfeys portining adresi.

ADD, ADC, ADI, ACI, DAD Qo`shish, SUB, SBB, SUI, SBI Ayirish, INR, INX Inkrement (bittaga oshirish), DCR, DCX Dekrement (bittaga kamaytirish), DAA O`nlik korreksiya, JMP Shartsiz o`tish, CALL Podprogrammani chaqirish, RET Podprogrammadan qaytish, JS Shartli o`tish, CS Shart bo`yicha podprogrammani chaqirish, RET Podprogrammadan qaytish.

Ma`lumotlarni uzatish komandalari

Tartib №	Mnemonika	Sikl-lar soni	Taktlar soni	Natija belgilari: S, Z, S, P, AC	Izox
1	MOV Ri, Rk	1	5	-----	Ri <= Rk
2	MVI Ri, data	3	10	-----	Ri <= data
3	LDA addr	4	13	-----	A <= (addr)
4	STA addr	4	13	-----	(addr) <= A
5	LDAX 2R	2	7	-----	A <= (2R)
6	STAX 2R	2	7	-----	(2R) <= A
7	LXI 2R, data16	3	10	-----	2R <= data16
8	LHLD addr	5	16	-----	HL <= (addr)
9	SHLD addr	5	16	-----	(addr) <= HL
10	SPHL	1	5	-----	SP <= HL
11	PUSH 2R	3	11	-----	(SP)+(SP-1) <= 2R
12	POP 2R	3	10	-----	2R <= (SP)+(SP-1)
13	XCHG	1	4	-----	DE <=> HL
14	XTHL	5	18	-----	(SP) <=> HL

Mantiqiy komandalar

Tartib №	Mnemonika	Sikllar soni	Taktlar soni	Natija belgilari: S, Z, S, P, AC	Izox
1	ANA R	1	4	0+++0	$A \leq A \& R$
2	XRA R	1	4	0+++0	$A \leq A \wedge R$
3	ORA R	1	4	0+++0	$A \leq A \vee R$
4	CMP R	1	4	+++++	$A == R$
5	ANI data	2	7	0+++0	$A \leq A \& \text{data}$
6	XRI data	2	7	0+++0	$A \leq A \wedge \text{data}$
7	ORI data	2	7	0+++0	$A \leq A \vee \text{data}$
8	CPI data	2	7	+++++	$A == \text{data}$
9	RLC	1	4	+----	$A7 \leftarrow A6 \leftarrow \dots \leftarrow A0 \leftarrow A7$
10	RRC	1	4	+----	$A0 \leftarrow A1 \leftarrow \dots \leftarrow A7 \leftarrow A0$
11	RAL	1	4	+----	$A7 \leftarrow A6 \leftarrow \dots \leftarrow A0 \leftarrow C \leftarrow A7$
12	RAR	1	4	+----	$A0 \leftarrow A1 \leftarrow \dots \leftarrow A7 \leftarrow C \leftarrow A0$
13	CMA	1	4	-----	$A \leftarrow (\text{inkor } A)$
14	CMS	1	4	+----	$C \leftarrow (\text{inkor } C)$
15	STC	1	4	1----	$C \leftarrow 1$

Arifmetik komandalar

Tartib №	Mnemonika	Sikllar soni	Taktlar soni	Natija belgilari: S, Z, S, P, AC	Izox
1	ADD R	1	4	+++++	$A \leftarrow A + R$
2	ADC R	1	4	+++++	$A \leftarrow A + R + C$
3	SUB R	1	4	+++++	$A \leftarrow A - R$
4	SBB R	1	4	+++++	$A \leftarrow A - R - C$
5	ADI data	2	7	+++++	$A \leftarrow A + \text{data}$
6	ACI data	2	7	+++++	$A \leftarrow A + \text{data} + C$
7	SUI data	2	7	+++++	$A \leftarrow A - \text{data}$
8	SBI data	2	7	+++++	$A \leftarrow A - \text{data} - C$
9	INR R	1	5	-++++	$\text{dst} \leftarrow R + 1$
10	DCR R	1	5	-++++	$\text{dst} \leftarrow R - 1$
11	DAA	1	4	+++++	$A \leftarrow 2/10 \text{ korr-ya } A$
12	DAD 2R	3	10	+----	$HL \leftarrow HL + 2R$
13	INX 2R	1	5	-----	$R16 \leftarrow R16 + 1$
14	DCX 2R	1	5	-----	$R16 \leftarrow R16 - 1$

Boshqaruvni uzatish komandalari

Tartib №	Mnemonika	Sikl-lar soni	Taktlar soni	Natija belgilari: S, Z, S, P, AC	Izox
1	PCHL	1	5	-----	PC <= HL
2	JMP addr	3	10	-----	PC <= addr
3	JC addr	3	10	-----	agar (C=1) PC <= addr
4	JNC addr	3	10	-----	agar (C=0) PC <- addr
5	JZ addr	3	10	-----	agar (Z=1) PC <- addr
6	JNZ addr	3	10	-----	agar (Z=0) PC <- addr
7	JM addr	3	10	-----	agar (S=1) PC <- addr
8	JP addr	3	10	-----	agar (S=0) PC <- addr
9	JPE addr	3	10	-----	agar (P=1) PC <- addr
10	JPO addr	3	10	-----	agar (P=0) PC <- addr
11	CALL addr	3	10	-----	Podprogrammaga o'tilsin
12	CC addr	3	11	-----	agar(C=1) CALL addr
13	CNC addr	5	17	-----	agar(C=0) CALL addr
14	CZ addr	3	11	-----	agar (Z=1) CALL addr
15	CNZ addr	5	17	-----	agar (Z=0) CALL addr
16	CM addr	3	11	-----	agar (S=1) CALL addr
17	CP addr	5	17	-----	agar (S=0) CALL addr
18	CPE addr	3	11	-----	agar (P=1) CALL addr
19	CPO addr	5	17	-----	agar (P=0) CALL addr
20	RET	3	11	-----	podprogrammadan qaytilsin
21	RC	1	5	-----	agar (C=1) RET
22	RNC	3	11	-----	agar (C=0) RET
23	RZ	1	5	-----	agar (Z=1) RET
24	RNZ	3	11	-----	agar (Z=0) RET
25	RM	1	5	-----	agar (S=1) RET
26	RP	3	11	-----	agar (S=0) RET
27	RPE	1	5	-----	agar (P=1) RET
28	RPO	3	11	-----	agar (P=0) RET

Boshqarish komandalari

Tartib №	Mnemonika	Sikllar soni	Taktlar soni	Natija belgilari: S, Z, S, P, AC	Izox
1	IN port	3	10	-----	port - dan kiritilsin

2	OUT port	3	10	-----	port - ga chiqarilsin
3	RST n	3	11	-----	Maxsus P/Pga utilsin
4	EI	1	4	-----	Uzilishlar taqiqlansin
5	DI	1	4	-----	Uzilishlarga ruxsat
6	HLT	1	4	-----	To'xtash komandasi
7	NOP	1	7		Navbatdagi komandaga o'tilsh

Ushbu komandalar asosida tuzilgan programma 16-lik sanoq sistemasidagi kodlarga o'tkazilib, mikroprotessorli xisoblash yoki boshqarish sistemasi xotirasining mos adreslariga joylashtirilganidan so'ng bajarilishi mumkin. Komandalarni 16-lik sanoq sistemasiga o'tkazish uchun quyidagi jadvaldan foydalaniladi:

Komanda joylashgan katakchanning avval qatorining raqami so'ngra ustunining raqami olinadi va ikkita 16-lik sanoq sistemasining raqamlaridan iborat komanda kodi xosil qilinadi. O'z ichiga 16 - razryadli adres yoki 16 razryadli ma'lumotni olgan komanda mikroprotessorli sistema xotirasiga quyidagi tartibda joylashtiriladi: 1 – bayt komanda kodi, 2 – bayt adres yoki ma'lumotning kichik bayti, 3 – bayt adres yoki ma'lumotning katta bayti.



6.1-рasm. KP580BM80 MP buyruqlar tizimining MP emulyatorida simulyatsiyasi.

Mikroprotessor va mikroehm bazali buyruqlar tizimi. MPning asosiy buyruqlari. bazali buyruqlar tizimi.

Hozirgi MP lar 20 ta dan 170 tagacha buyruqlarni o'z ichiga oladi. Har bir MP o'zining shaxsiy buyruqlar tizimiga ega. Bu buyruqlar 1 tadan to 8 ta baytgacha bo'lgan formatga ega bo'lishi mumkin [2.3].

Bajaradigan vazifasi bo'yicha har qanday MP dagi buyruqlar to'plamini quyidagi buyruqlar guruhiga bo'iish mumkin:

- 1.O'tkazish buyruqlar gurihi.
- 2.Arifmetik amallarning buyruqlar gurihi.
- 3.Logik amallarning buyruqlar gurihi.
- 4.Uzatish boshqarish buyruqlar gurihi.
- 5.Maxsus buyruqlar gurihi.

8-razyadli bitta kristalli K580 VM80A mikroprotsessoming buyruqlar tizimi jadvalda ko'rsatilgan. MP buyrug'i bu shunday ikkilik so'ziki, so'z (buyruq) MP tomonidan o'qilgandan keyin MP ni ma'lum harakatini bajarishiga majburlaydi.

Ko'pchilik buyruqlar MP ni xotirasidagi yoki biron bir registrdagi qiymatlarni bir tomondan ikkinchi tomonga e'tkazish uchun ishlatiladi.

Buyruqlaming uzunligi berilgan ikkilik qiymatlar so'zining uzunligi bilan moslashadi. 8-razyadli MP buyruq so'zining uzunligi 8 bitga, 16-razyadli MP niki esa 16 bitga teng. Buyruqlar ikki yoki uch so'zga teng uzunlikga bo'lish mumkin.

Buyruq bajarilishi uchun u buyruqlar registrga BRG, deshifratonga va boshqa boshqarish sistemasiga yuboriladi. U yerda buyruq identifikasiyalanadi (qanday buyruq ekanligi aniqlanadi). Buning natijasida MP ning boshqa qismlarga yuboriladigan xabarlar /impul'slar/ tashkil etadi. Bu xabarlar yordamida buyruqlarda ko'rsatilgan amallar bajariladi.

MP buyruqni BRG ga tanlash davri davomida yuklaydi. Bundan keyingi bajarish davrida MP buyruqni dekodlaydi va bu buyruqni bajarish jarayoni uchun boshqarish xabarini yaratadi. MP buyrug'ini sinchiklab qaraganda. buyruq ikki xil ma'lumotni o'z ichiga olishi kerak:

Birinchi buyruq MP ga nima qilish kerakligini aniqlaydi (qo'shish, ayirish, tozalash, surish, yuborish va sh.o'xsh. bajarish);

Ikkinchidan, qayta ishlanadigan qiymatni (operandni) o'rnini ya'ni adresini ko'rsatishi kerak. Buyruq ikki qismdan tashkil topgan: a) amallar kodi (AMK) - MP ga nima qilish kerakligini xabar qiladi; b) adres esa amalda qatnashadigan qiymatiarni o'rnini ko'rsatadi. Ayrim buyruqning ikkinchi va uchunchi baytlarda ishlaydigan qiymatlar ham bo'lishi mumkin.

Agar buyruqning uzunligi ikkita yoki uchta so'zdan tashkil topgan bo'lsa. U holda ulardan birinchisi - amallar kodi, ikkinchisi va uchinchisi esa qiymatlarning

joylashgan adresini ko'rsatadi. Bitta so'z, uzunligidagi hamma buyruqlar adressizdir.

8-razryadli bir kristalli K580IK80A mikroprotessorining buyruqlar tizimi (sistemi).

6.1-jadval. O'tkazish buyruqlar guruhi

O'tkazish buyruqlar guruhi						
№	Mnemokomandalar	Amal	Mashina davri	Mashina takti	Formati	Boshqaruvchi registry, so'zni belgilari
1	2	3	4	5	6	7
1	MOY R1 R2	(R1) (R2)	1	5	1	Hamma belgilar o'z qimatlarini saqlab qoladi
2	XSNS	(NL) (DE)	1	4	1	
3	SRNL	(SR) (NL)	1	5	1	
4	MOY R, M	(R) M(NL)	2	7	1	
5	MOY M, R	M(NL)(R)	2	7	1	
6	STAX RR	M(KR) (A)	2	7	1	
7	LDARR	(A) M(KR)	2	7	1	
8	LDA A16	(A) M(A16)	4	13	3	

Ikki baytli buyruqlar xotira yacheykalariga bevosita adreslash usuli orqali amalga oshiriladi, uch baytli buyruqlar esa xotira yacheykalariga to'g'ri adreslash usuli orqali amalga oshiriladi.

Bunda quyidagi qisqartirilgan belgilar ishlatiladi:

—< o'tkazish amali;

<—> — almashish amali;

AND — kon'yunksiya (I);

OR — dizyunksiya (I-U);

XOR — modul 2 buyicha qo'shish (ILI ni inkor etish);

INV — inversiya (aylantirish);

R — A,B,C,D,B,H,L registrlaridan bittasi;

RP — Juft B,D,I I yoki SP registrlaridan bittasi;

RPI — V yoki D juft registrlaridan bittasi;

RPH — juft registrining katta registri;

RPL — juft registrining kichik registri;

M — HL registri orqali vositali adreslanadigan xotira;

PORT — kiritish/chiqarish portining 8-razryadli adresi;

NI sakkizta 0,1,2,3,4,5,6,7 vaqtincha uzishning biltasi;

D8 — sakkizta razryadli bevosita operand;

D16 - o'n olti razryadli bevosita operand;

A16 — o‘n olti razryadli adres:

(R) — regisiming qiymati;

Adreslashning quyidagi usullari mavjuddir [1,8,10,21]:

- Registrli adreslash;
- Bevosita adreslash;
- To‘g‘ri adreslash;
- Vositali registrli adreslash;
- Stekli adreslash;
- Ayon bo‘lmagan adreslash;

Registrli adreslash. Bu adreslash 8-razryadli MPda bitta baytli buyruq orqali amalga oshiriladi, 8 bit (bayt) mashina so‘zini tashkil etadi va 256 ta har xil kombinasinyaning bittasidir.

Qanday qilib bir baytli buyruqdan qiymatlarga adreslash uchun foydalanish mumkin? Javob quyidagichadir:

Bitta baytli buyruqlarda xotiraga joylashgan qiymatlarga murojaat qilinmaydi. Ular registrlarga, juft registrlarga yuklangan qiymatlar bilan ishlaydi yoki bo‘lmasa adresi juft registrlarda bo‘lgan, o‘zi xotirada saqlanayotgan qiymatlar bilan birga ishlaydi.

Masalan: A registridan V registriga qiymatiarni o‘tkazadigan bir baytli buyruq: amallar kodidan, qiymatlar manbasining adresidan (A registri) va qiymatiarni qabullovchining adresidan (V registri) tashkil topgan bo‘ladi. (4.9- rasm).

D7	D6	D5	D4	D3	D2	D1	D0
----	----	----	----	----	----	----	----

O	1	O	O	O	1	1	1
---	---	---	---	---	---	---	---

MOV V registring adresi A registring adresi

6.9-rasm. A registring qiymatlarini V registrida o‘tkazadigan registr a adreslash usuli ishlatilgan buyruq.

Bu yerda, amallar kodi — 01- MOV;

A registring adresi — 111;

V registring adresi — 000;

Buyruqda manbaning va qabul qiluvchining adresi aniq ajratib (alohida) ko‘rsatilmagan. Bu buyruq 2 - mikrosiklda bajariladi. Birinchi mikrosiklda amallar kodi tanlanadi, ikkinchi mikrosiklda esa u bajariladi. (A registring qiymati V registriga o‘tkaziladi).

Registrli adreslash usuliga misollar:

MOV A, L - L registri qiymati A registrga o‘tkazilsin.

(A)<-(L).

SPHL - HL registri qiymati SP registriga o‘tkazilsin.

(SP)<-(HL).

ORAL - A ni qiymati bilan L registrini qiymati ustida razryad buyicha ILI amali bajarilsin $(A) \leftarrow (A) - (L)$

Ayon bo'lmagan adreslash usuli. Ayon bo'lmagan adreslash usuli bir davri buyruqdan tashkil topgan bo'lib 2 - mikrosiklda bajariladi. Birining mikrosiklda amallar kodi tanlanadi, ikkinchi mikrosiklda esa u bajariladi. Ayon bo'lmagan adreslash usuli asosan akkumulyatordagi qiymat ustida amalga oshiriladi.

Ayon bo'lmagan adreslash usuliga misollar:

SMA - Akkumulyatorning qiymatini teskarisiga aylantirish, ya'ni $(A) \leftarrow (A)$.

KAK. KAX - Akkumulyatorning qiymatini o'ng va chap tamonlarga bitta qadamga surish.

Bevosita adreslash usuli. Bu adreslash usulida buyruq amallar kodining (AMK) birinchi baytiga bevosita joylashtiriladi. Amallar kodidan keyin esa darhol bitta yoki ikkita baytga ega bo'lgan qiymatlar yoziladi (joylashtiriladi). Bu qiymatlar xotiradan olinmaydi, ularning buyruqlarini mashinaga programmasining o'zi kiritadi. Bunday buyruqni registrni bevosita yuklash deyiladi. Albatta, bu joyda AMK va qiymatlar ikkilik kodi bo'yicha ma'lumotga ega bo'ladilar. Bevosita adreslash usulining tuzilishi quyidagicha.

Ammalar kodi	1 bayt
Qiymatlar	2 yoki 3 bayt

6.2-jadval. Bevosita adreslash buyrug'ining tuzilishi.

Bevosita adreslash usuliga misollar:

LXIN qiymatlar NL registrlari qiymatlar bilan yuklan sin. $(L) \leftarrow (2 \text{ bayt}); (N) \leftarrow (3 \text{ bayt})$. Bu yerda, L registriga amallar kodining (AMK) ikkinchi baytida turgan qiymat yuklanadi. N registriga esa AMK ning uchinchi baytidagi qiymati yuklanadi (qiymatni operatorning o'zi kiritadi).

MVI A. qiymatlar akkumulyator (A) AMK ning ikkinchi baytida turgan qiymat bilan yuklansin $(A) \leftarrow (2 \text{ bayt})$.

LXI SR, qiymatlar SP registry qiymatlar bilan yuklansin $(SR) \leftarrow (2 \text{ bayt} - 3 \text{ bayt})$.

ANI, qiymat akkumulyatorni qiymati va AMK ning ikkinchi baytida turgan qiymat ustida razryad bo'yicha mantiqiy I amali bajarilsin $(A) \leftarrow (A) * (2 \text{ qiymat})$.

Mikroprotsessorda bevosita adreslash ikkita mikrosiklda amalga oshiriladi. Birinchi mikrosikl davomida buyruqni tanlash o'tkaziladi, ikkinchi mikrosiklda esa buyruq bajariladi.

Bu adreslash usulida qiymatiarni qaysi bir xotira massiviga joyiashtirish hamda bu qiymatlarga murojaat qilishga zarurat tug'ilganda foydalaniladi. To'g'ri adreslash

buyrug‘i 2 yoki 3 baytli uzunlikka ega bo‘lishi mumkin. Birinchi bayt amallar kodi uchun belgilangan, Ikkinchi baytlar adreslar uchun belgilangan. Adres qayta ishlashga belgilangan qiymatlar joylashgan xotira massivini ko‘rsatadi.

To‘g‘ri adreslash usulida kerakli qiymatiarni adresini ochiq berishga (ko‘rsatishga) imkoniyat bo‘ladi.

Ammalar kodi(AMK)	1	bayt
Adresning katta razryadi	2	bayt
Adresning kichkina razryadi	3	bayt

6.3 - jadval. Adreslashga misol.

Shunday qilib, to‘g‘ri adreslashni bajarishga sarf qilingan vaqt buyruqlarni bevosita adreslash usuliga sarf qilingan vaqtdan ikki marta ko‘pdir. To‘g‘ri adreslash usulini qiymatiarni xohlagan hotira massiviga joyiashtirish zaruriigi kerak bo‘lgan taqdirda qo‘llash maqsadga muvofiqdir. Ayrim MP da uchta mikrosiklda amalga oshiriladigan to‘g‘ri adreslash usuli qo‘llaniladi. Bunday holda buyruq 2 baytli uzunlikka ega bittasi amallar buyrug‘i. boshqasi adress uchun.

To‘g‘ri adreslash usuliga misollar:

LHLD, adres NL juft registri xotiradagi qiymat biian yukiansin. L regisiiriga xoiiradagi qiymaining kichik bayti. N registriga esa xotlradagi qiymatining katta bayti yukiansin. (L)<-(2-bayt);(N)<-(3-bayi)

S H LD, adres NL juft registridagi qiymat xotiraga joylashtirilsin (L) -> (adresning kichik bayti); (N)-> (adresning katta bayti).

IN, portni adresi akkumulyatorga ko‘rsatilgan portning adresidagi qiymat kiritilsin (A) <- (portning adresi).

OUT, portning adresi ko‘rsatilgan sdresdagi portga akkumulyatorning qiymati chiqarilsin (portni adresi)<-(A).

Vositali adreslash.

Vositali adreslash bitta so‘z uzunligiga ega bo‘lgan buyruq orqali amalga oshiriladi. Bunday adreslash vositali yoki ayrim paytda vositali registrii adreslash deyiladi. Bu yerda buyruqning qiymati amallar kodidan tashqari berilgan qiymatlar joylashgan xotiraning adresini ham ko‘rsatadi. 8-razryadli MP da vositali adreslash usulidan foydalanilganda, tegishli buyruq shuni ko‘rsatadiki, qaysiyam bir juft registrga yozilgan adreslar, ishlatilishi kerak bo‘lgan, xotiraga joylashgan qiymaining adresini ko‘rsatadi.

Vositali adreslash tez-tez foydalanib turiladigan xotira massiviga murojaat qilganida hamda asosan qiymatlar fayl yoki ro‘yxat ko‘rinishida tashkil etilganda qulaydir. Vositali adreslash usuli quyidagi ko‘rinishda yoziladi.

M Vositali yuklash	1	bayt
--------------------	---	------

4.4- rasm. Vositali adreslash usuli.

Vositali registrli adreslash usuliga misollar:

SUB M -Akkumulyatorida saklanayotgan qiymatdan xotira massividagi qiymat ayirib lashlansin, (A)<-(A) - ((N)(L)). Xotira massividagi qiymatlarning adreslari NL juft registrilarida saqlansin.

SRM M - Akkumulyatorning qiymati xotira massivining qiymati bilan taqqoslansin. (A) - ((N)(L)).

ADD M - Akkumulyatorning qiymati xotira massividagi qiymat bilan qo'shilsin (A)<-(A)+((N)(L)).

Stekli adreslash.

Stekli adreslashda operandning adresi stekning ko'rsatkichini registrda ko'rsatiladi, bu yerda operandga murojaat qilganda stek ko'rsatkichining qiymati avtomatik ravishda ikkitaga yo ko'payadi, yo kamayadi..

RUN buyrug'i bajarilganda stekli ko'rsatkichning SR-1 adresli xotirasiga juft registming katta razryadini qiymati yoziladi, SR-2 adresli xotirasiga esa juft registring kichkina razryadini qiymati yoziladi. SR stek ko'rsatkichining qiymati ikkitaga kamayadi.

ROR buyrug'i (qiymatiarni stekdan chiqarish) bajarilayotganda stek cho'qqisining qiymati juft registming kichkinasiga uzatiladi, stek cho'qqisining qiymatidan bitta katta bo'lgan adresdagi qiymat esa juft registming kattasiga uzatiladi. Shundan keyin ZR registrining (stekni ko'rsatkichning) qiymati ikkitaga ko'payadi. Stekning to'g'ri ishlashi uehun RUSH va ROR buyruqlari birgalikda ishlashlari kerak. Ayrim paytlarda bu buyruqlarning o'rnida ularga ekvivalent bo'lgan SALL (qism programmasini chaqirish) va RETURN (asosiy programmaga qaytish) buyruqlari ishlatiladi, Ya'ni SALL -RUSH; RETURN - ROR.

M(RP) — RPjaft registrini adresi buyicha saqlanadigan xotira yacheykasining qiymati;

I(PORT) — (PORT) adresidagi kiritish portini qiymati;

O(PORT) — PORT adresidagi chiqarish poriini qiymati;

COND - ushbu sakkizta shartning bittasi;

NZ — nol bo'imagani natija (z?=0);

Z — nolli natija (Z=I);

NC - katra razryaddan o'tkazishni (qarzni) katta razryadga ko'chirishni yo'qligi (CY=-O);

S — ko'chirishning yo'qligi;

RO — qiymatda birlarning sonini toqligi (R=O);

RH — qiymatda birlarning sonini juftligi (R=O);

R — "musbat" (S=0);

M — "manfiy" (S=I);

MTS — mashina davrining miqdori;

MT — mashina taktining miqdori;

F — baytlar da buyruqlamlg formati.

Buyruqlarni mnemonik shaklda yozish. MP ni buyruqlari bu ikkilik sonlar. Buyruqlar bir baytii, ikki baytli, ya'ni sakkiz va o'n olti razryadli bo'lishi mumkin. Amaliyotda hattoki bitta baytli sonlarni eslab qolish juda ham qiyindir. Sakkiz va o'n oltilik sanoq sistemasidagi sonlar bilan MP buymqlami belgilasa bo'lar edi. Biroq bu holda ham bu sonlarni (buyruqlarni) esiab qolish va nimaga mo'ljallanganligini ajratib olish juda qiyindir.

Bunday muammo mnemonik (belgilash) yo'li bilan amal qilinadi. Ya'ni bu yerda atalgan buyruqning qisqartirilgan yozuvi qo'llaniladi. Bunday maqsadda amalning nomini bajarilayogan buyruqning uchta harfidan foydalanib belgilanadi.

Masalan, tozalash buyrug'ining mnemonik belgilanishi quyidagi ko'rinishga ega bo'ladi: CLA.

KR5801K8C seriyali MP bazali buyruqlar tizimi, mnemokod ko'rinishida 4.3-jadvaida keltirilgan. Mnemokodlarni o'n oltilik sanoq tizimiga o'tkazish uchun avvalo kerak bo'lgan mnemokodni 2-jadvaldar. tanlash kerak, keyin esa jadvalning vertikal va gorizontal chegaralaridagi tanlangan mnemokod bilan kesishadigan katakchalaridan mnemokodga ekvivalent bo'lgan 16 lik mashina kodi topiladi.

Mnemokodni mashina kodiga o'tkazish paytida vertikal ustundagi 16 lik sonlar birinchi bo'lib, gorizontal qatordagi 16 sonlar esa ikkinchi bo'lib yozilishi shart.

Masalan: STAX V - 02 INX V - 03

POP N - E1 MVI L, - 2E

Savollar:

1. Bazali buyruqlar sistemasi va uni guruhlarga ajratish qanday?
2. Arifmetik, logik, yuklash (o'tkazish), boshqaruvchi, shartli va shartsiz, o'tish qanday amalga oshadi?
3. Boshqa buyruqlar guruhi to'g'risida tushuncha bering.
4. Buyruqlarning, amallar kodining bajarilishiga misollar keltiring.