

Predavanje 4.

8-bitni mikroprocesori

O mikroprocesorima

Mikroprocesor (μ P) je složena struktura koja je nastala kao logičan slijed razvoja logičko-sekvencijalnih struktura. U konfiguraciji sa drugim komponentama omogućava realizaciju složenih i različitih zadataka. Treba napomenuti da mikroprocesor kao samostalna jedinica ne predstavlja upotrebljivu strukturu. Ne baš tako davne 1955. godine ljudi su se divili čudu tehnike koje se zvalo IBM 650 i koje je sa svojih 2000 elektronskih cijevi zapremalo 8 kubnih metara, težilo dvije tone i koštalo 200.000,00 dolara. Prije nekih dvadesetak godina divili smo se džepnom računaru TI-59 koji je, sa svojih 166.500 tranzistora, težio 300 grama, zapremao 480 kubnih centimetara i koštao 200 dolara - memorije ova dva računara su približno jednake dok TI-59 radi desetak puta brže.

Danas se rijetko ko posebno divi mikroracunarima koji imaju stotinama puta veću memoriju, mnogostruko brže obrađuju podatke i mnogo manje koštaju. Tajna gotovo neshvatljivo brzog razvoja računarske tehnike je mikroprocesor, "mozak" svakog kompjutera koji se razvijao znatno brže od ma kog ljudskog pronalaska.

Historijski, firma DATA POINT projektovala je krajem 60-tih godina prošlog vijeka male računare i terminale želeći da se u terminal ugradi mali računar te tako proizvedu "inteligentni terminali". Projekat takvog malog računara su ponudili firmama INTEL i TEXAS INSTRUMENTS (TI) na realizaciju. TI uvidjevši da ne može postići sve performanse, odustao je od ugovora. INTEL je ugovorio posao i uložio ogromna sredstva u ispitivanje i razvoj. Rezultat je bio mikroprocesor koji zadovoljava sve zahtjeve DATA POINT-a izuzev brzine rada (bio je 10 puta sporiji). Zbog toga je DATA POINT raskinuo ugovor na štetu INTEL-a. Uloživši ogromna sredstva u razvoj mikroprocesora INTEL je odlučio da ga ponudi tržištu kao serijski proizvod INTEL 8008 (decembar 1971. godine). Pojavom mikroprocesora na tržištu nastao je pravi "bum", pa su uskoro i ostale firme lansirale svoje prvence (ROCKWELL, INTEL 8080 - 1973. god., MOTOROLA 6800, TI SPB 400, GENERAL INSTRUMENT, itd.). 1974. godine pojavljuje se prvi 16-bitni μ P.

Mikroprocesori danas spadaju u najkomplicovanija integralna kola. Male dimenzije im omogućavaju smanjenu potrošnju u odnosu na prethodnike, mnogo veće brzine rada i, možda najvažnije, mogućnost upotrebe praktično u bilo kom uređaju. Sve ovo je učinilo mikroprocesor jednim od najvećih dostignuća 20.stoljeća

Intel

Sve je počelo 1960. godine proizvodnjom prvih integrisanih kola, ogromnih skupova tranzistora, dioda, otpornika i kondenzatora smeštenih na silicijumsku pločicu. Od integrisanih kola koja su postajala sve složenija i sve jeftinija često su, jasno, sklapani računari, a samim tim i njihove procesorske ploče. Samo deset godina kasnije, 1970, Intelov inženjer Marsijan Hof (Marcian Hoff) dolazi na revolucionarnu ideju da projektuje integrisano kolo koje bi samostalno obavljalo određen dio posla standardne procesorske ploče. Pokazalo se da i na tadašnjem stupnju razvoja tehnologije integrisano kolo može da preuzme kompletnu funkciju procesorske ploče i tako je nastao prvi mikroprocesor - Intel 4004. To je bio četvorobitni mikroprocesor. Sastojao se od oko 2250 tranzistora i u svakom sekundu izvršavao 60.000 operacija.

Značajan komercijalni uspeh mikroprocesora 4004 učinio je da se Intel zainteresuje za ovakvu proizvodnju i da godinu dana kasnije ponudi tržištu mikroprocesor 8008, osmobitnog naslednika 4004. Smatra se da 4004 i 8008 čine prvu generaciju mikroprocesora.

Na drugu generaciju nije trebalo dugo čekati - 1974. Intel predstavlja mikroprocesore 8080 i, nešto docnije, 8085. Premda i dalje osmobitni, ovi su mikroprocesori značajno moćniji od svojih prethodnika i predstavljaju prve mikroprocesore koji u potpunosti zaslužuje epitet "opšte namene". Karakteristike 8080, a naročito, 8085 su učinile da ovi čipovi mnogostruko nadžive svoj "prirodni vijek" pa se i danas nalaze u mnogim računarima i specijalizovanim kontrolerima.

Zilog

Z-80 je zapravo dizajnirala mala grupa "odbjeglih" Intelovih inženjera na čelu sa [Federicom Faggin](#)ijem koji su osnovali nezavisnu firmu Zilog. Z-80 je u značajnoj mjeri kompatibilan sa 8085 - svaki program pisan za 8085 radit će i na Z-80! Obrnuto, naravno, ne važi - Z-80 je hardverski i softverski unapređen što znači da racionalnije komunicira sa periferijom i poseduje određen broj moćnih instrukcija koje se bave manipulacijom većim grupama podataka. Z-80 je u toku narednih godina pretrpio nekoliko revizija pa su se pojavile ubrzane varijante nazvane Z-80A, Z-80B i Z-80H. Ravidirana je i cijena - dok su prvi primjerci Z-80 koštali 250 dolara, danas se ovaj mikroprocesor nabavlja za par dolara!

Ostali

Do sada smo govorili isključivo o Intelovim mikroprocesorima i njihovim neposrednim parnjacima kao što je Z-80. Na tržištu, međutim, paralelno egzistira još nekoliko porodica mikroprocesora među kojima je najvažnija Motorolina. Motorola je počela sa osmobicnim mikroprocesorima 6800, 6802, 6803, 6808 i 6809 koji su međusobno prilično slični ali je pravu popularnost postigla tek dizajnirajući šesnaestobitni mikroprocesor MC68000.

Nabrajajući mikroprocesore nikako ne smijemo da zaboravimo 6502, jedini veliki tržišni uspjeh relativno nepoznate firme MOS Technology. 6502 je osmobicni mikroprocesor na kome su zasnovani mnogi slavni kućni računari: Apple II, Commodore 64 i 128, BBC, Electron... Rasklapanjem ovih kompjutera, istini za volju, nećete pronaći baš čip čija je oznaka 6502: varirajući pojedine hardverske karakteristike svog remek-dela, firma MOS Technology proizvela je i mikroprocesore 65C02, 6510, 6512, 8512 i 65102 koji se, sa programerske tačke gledišta, vrlo malo razlikuju od osnovnog modela. Iako mikroprocesor 6502 nije imao šesnaestobitnih i trideset dvobitnih naslednika, može se smatrati da je dio njegove filozofije ugrađen u trenutno veoma moderne RISC mikroprocesore.

Mikroprocesor INTEL 8085

Da bi se mogao razumjeti rad mikroprocesora, najbolja osnova ta to je Intel 8085. To je 8-bitni mikroprocesor proizveden sredinom 1970-ih. Kompatibilan je sa 8080 i zahtjeva manje eksternog hardvera što omogućuje izgradnju jednostavnijih i jeftinijih računarskih sistema.

Zahtjevao je samo jedan napon od +5V za razliku od 8080 koji je zahtjevao 3 vrste napona: +5V, -5V i +12V. Korišten je i kao mikrokontroler zbog osobine da nije zahtevao puno dodatnog hardvera.

➤ Arhitektura 8085

- Adresna sabirnica (Address Bus) - širine 16 bita = mogućnost adresiranja do 64KB memorije.
- Sabirnica za podatke (Data Bus) - širine 8 bita = mogućnost pristupa podacima širine 8 bita u jednom koraku, odavde potiče naziv 8-bitni mikroprocesor.
- Kontrolna sabirnica (Control bus) - prenosi kontrolne (upravljačke) signale

Intel 8085 može da pristupi 65.536 memoriskim 8-bitnim lokacijama (64k). Ima odvojen adresni prostor od 256 8-bitnih lokacija za ulazno/izlazne operacije (I/O ports). Takođe sadrži i registre A, B, C, D, E, H, L - univerzalni, 16-bitne Program Counter (PC) i Stack Pointer (SP), i 8-bitni flag registar F za tri maskirajuća prekida (RST), jednim nemaskirajućim (TRAP) i jednim spoljašnjim prekidom (INTR) koji se takođe može maskirati.

➤ Tehnički podaci

- Frekvencija 5MHz;
- 6500 tranzistora na $1 \mu\text{m}^2$
- Sabirnica podataka 8bita;
- Adresna sabirnica 16 bita
- U AH verziji je 20% manja potrošnja energije nego kod normalnog 8085 procesora.
- Direktno adresiranje memorije od 64 Kbajta;
- 1.3 μs ciklus naredbe (0.8 μs kod 8085AH-2 /0.67 kod 8085AH-1);
- 4 vektorska interapt ulaza (jedan nije moguće maskirati);
- Decimalna, binarna i aritmetika sa dvostrukom preciznošću.

Pored Intel-a, ovaj procesor proizvode i drugi proizvođači.

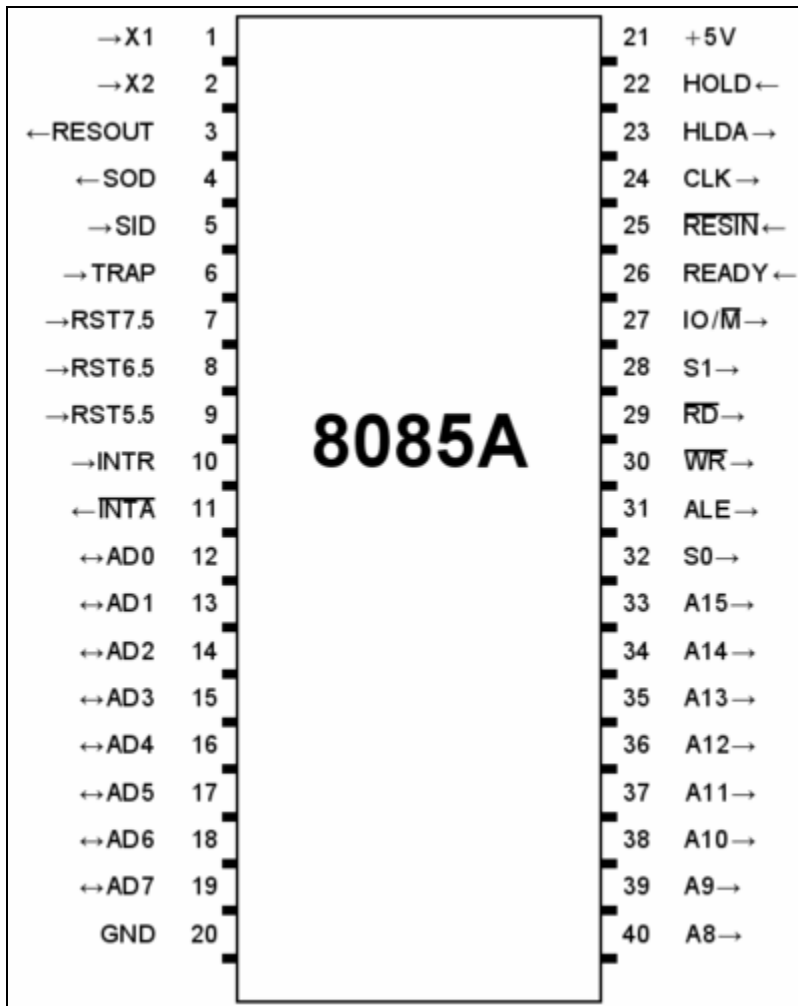
➤ Funkcionalni opis

8085A je 8-bitni paralelni centralni procesor. Dizajniran je u N-kanalnoj CMOS tehnologiji sa smanjenim opterećenjima i zahtijeva samo jedno napajanje od 5V. Osnovna brzina oscilatora je 3MHz za 8085A ili 5Mhz za 8085A-2, i na taj način poboljšava performance 8080A sa većim brzinama sistema. Konstruisan je da se uklapa u minimalni sistem od tri IC-a: 8085A CPU, 8156 RAM/IO i 8355 ili 8755A ROM ili EPROM/IO.

8085A ima 12 8-bitnih adresabilnih registara. Četiri od njih mogu funkcionisati kao dva 16-bitna para registara. Šest ostalih se koristi i kao 8-bitni i kao 16-bitni registri.

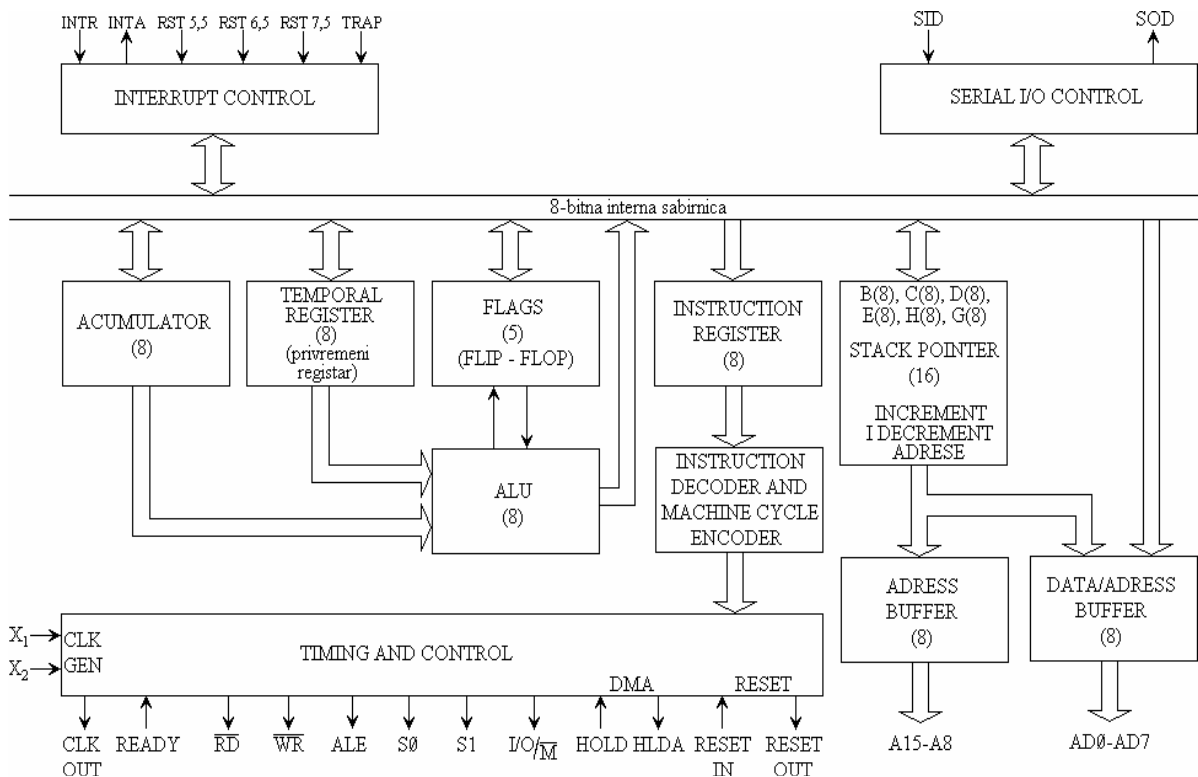
8085A koristi multipleksiranu sabirnicu podatka. Adresa je podijeljena između 8 gornjih (više značajnih) bita adresne sabirnice i 8 donjih (manje značajnih) bita adrese sabirnice.

➤ Pinovi mikroprocesora i njihove funkcije



Slika 4.1: Konfiguracija mikroprocesora

Na slici 4.2 data je blok šema Intel 8085 mikroprocesora, a u tabeli 4.1 su objašnjeni signali mikroprocesora.



Slika 4.2: Arhitektura mikroprocesora 8085AH

Tabela 4.1: Značenje pojedinih pinova kod mikroprocesora 8085AH

OZNAKA	PIN	I/O	FUNKCIJA
A₁₅ – A₈, A_H	21 – 28	O	Adresna sabirnica, gornji adresni bajt za adresu memorije ili adresa porta.
AD₇–AD₀, A_L,D₇–D₀	12 – 19	I/O	Multipleksirane adrese i sabirnice podataka (bus). Donjih 8 bita od adrese se pojavljuje na busu za vrijeme prvog clock ciklusa (T-stanja) mašinskog ciklusa. Za vrijeme drugog i trećeg ciklusa pojavljuje se podatak na busu.
ALE	30	O	Address Latch Enable se pojavljuje za vrijeme prvog clock stanja mašinskog ciklusa i omogućava da se adrese lečuju u periferalima. Opadajuća ivica ALE je podešene da garantira latchovanje adresa. Ista linija može se koristiti za strobovanje statusnih informacija.

S0, S1	29, 33	O	Mašinski status signali, pokazatelji iskorištenosti CPU-a															
			<table border="1"> <thead> <tr> <th>S1</th> <th>S0</th> <th>Statusna informacija</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>Čekati</td> </tr> <tr> <td>0</td> <td>1</td> <td>Pisati</td> </tr> <tr> <td>1</td> <td>0</td> <td>Čitati</td> </tr> <tr> <td>1</td> <td>1</td> <td>Dobaviti kod funkcije</td> </tr> </tbody> </table>	S1	S0	Statusna informacija	0	0	Čekati	0	1	Pisati	1	0	Čitati	1	1	Dobaviti kod funkcije
			S1	S0	Statusna informacija													
			0	0	Čekati													
			0	1	Pisati													
1	0	Čitati																
1	1	Dobaviti kod funkcije																
Zajedno sa IO/\overline{M} definira tekući mašinski ciklus.																		
IO/\overline{M}	34	O	Input-Output/Memory Izlazni signal daje obavještenje da li se operacija čitanje/pisanje vrši preko: <ul style="list-style-type: none"> - ulazno/izlaznog uređaja $IO/\overline{M} = 1$ - preko memorije $IO/\overline{M} = 0$. 															
\overline{RD}	32	O	Read (Low-aktivan) Čitanje je analogno WR. Sa rastućom ivicom čita se podatak iz memorije.															
\overline{WR}	31	O	Write (Low-aktivan) Pisanje je signal aktivan sa nulom. Pokazuje da se podatak može upisati u memoriju ili I/O periferal. Podatak je zapamćen na rastuću ivicu WR.															
READY	35	I	Ready Ako je Ready $\rightarrow 1$ za vrijeme ciklusa čitanja ili pisanja to znači da su memorija ili periferal spremni poslati ili spremni podatak. Ako je Ready $\rightarrow 0$ - CPU će čekati cijeli broj clock ciklusa da Ready ode u logičko 1 prije nego se kompletira Read ili Write ciklus. U slučaju da je periferal spor, tj. Kada CPU da signal čitanja ili pisanja tog periferala praktično se čeka upravljački signal da periferal izda ili primi podatak.															
HOLD	39	I	Hold Hold pokazuje da drugi master zahtijeva korištenje adresnog ili data busa. CPU nakon prijema Hold Request će napustiti korištenje busa, čim se kompletira tekući bus transfer može se nastaviti interno procesiranje. Procesor može zauzeti bus samo ako prestane Hold signal. Kada je Hold potvrđen adrese, podaci i kontrolni signali su u three state.															

HLDA	38	O	Hold Acknowledge Pokazuje da je CPU primila Hold signal (zahtjev) i da će napustiti bus u slijedećem clock ciklusu. HLDA ← 0 kada je Hold zahtjev == 0. pola clock ciklusa CPU ponovo preuzima bus.
INTR	10	I	Zahtjev za prekidom (Interrupt Request) Koristi se kao prekid opšte namjene. Uzorkovan (semplovan) je za vrijeme zadnjeg clock ciklusa pri završetku instrukcije i za vrijeme HOLD i HALT stanja. Ako je aktiviran, programski brojač će biti zabranjen da se inkrementira i INTA (potvrda INTR-a) će se generisati. Za vrijeme ovog ciklusa RESTART ili CALL instrukcija može biti ubačena da se skoči na servisnu rutinu. INTR je omogućen ili onemogućen softverski. Nije poželjno nakon RESETA da se odmah ima interrupt. U opštem, Interrupt signal za prekid programa se softverski omogućuje ili maskira.
INTA	11	O	Potvrda interapta (Interrupt Acknowledge) Kada se ima zahtjev za INTR i ako je softverski omogućen prekid procesor će nakon završetka tekuće instrukcije generisati INTA' – potvrda prekida. Ovaj signal koristi se umjesto čitanja RD' da se pročita vektor adresa servisne rutine za prekid.
RST5.5 RST6.5 RST7.5	9 8 7	I	Restart-Interrupts Ova tri ulaza imaju isti timing (vremenski dijagram kao INTR) izuzev što uzrokuju interni restart koji se automatski aktivira. Prioritet ovih ulaza adekvatan je navedenom redoslijedu (RST7.5 ima najveći prioritet). Imaju veći prioritet od INTR. Softverski mogu biti maskirani SIM (Set Interrupt Mask).
TRAP	6	I	Nemaskirani interrupt najvećeg je prioriteta.
RESIN	36	I	Reset Input Sa 0 signalom se procesor vraća u osnovno stanje. Preko RESET-a programski brojač se postavlja na nulu.
RESOUT	3	O	Reset Output Može se koristiti kao sistemski reset. Ovaj signal je sinhroniziran sa taktom procesora.
X1, X2	1, 2	I	Priključci za spoljni takt.

CLK	37	O	Clock Taktni izlaz korišten kao sistemski takt.
SID	5	I	Serial Input Data Ulaz za serijski prenos podataka, preko RIM komande se susjedni bit puni u akumulator.
SOD	4	O	Serial Output Data Izlaz za serijski prenos podataka, preko SIM naredbe se može postaviti na 0 ili 1.
V_{CC}	40		Napajanje (+5V)
V_{SS}	20		Masa (0V)

Registri

Osmobitni registri mogu se upariti za 16 bitne naredbe. To su: A/FLAG, B/C, D/E i H/L. Ovo uparivanje registara važno je za stack-naredbe PUSH i POP, kao i kod adresnih naredbi i kod računanja adresa.

Tabela 4.2:

REGISTAR	
OZNAKA	FUNKCIJA
A	Akumulator (8 Bit)
B	Registar opšte namjene (8 Bit)
C	Registar opšte namjene (8 Bit)
D	Registar opšte namjene (8 Bit)
E	Registar opšte namjene (8 Bit)
H	Registar opšte namjene (8 Bit)
L	Registar opšte namjene (8 Bit)
FLAG	Registar stanja (8 Bit)
INT	Interrupt registar (8 Bit)
IC	Instrukcijski brojač (16 Bit)
SP	Stack pointer (16 Bit)

Registri opšte namjene B,C,D i E služe za logičke i aritmetičke osmобitne operacije, koje su takođe moguće sa H i L registrima. Registri H i L su bitni kod 16-bitnih operacija. Rezultati svih operacija se smještaju u akumulator A. U tabeli 4.2 date su oznaske registara.

Parovi registara imaju posebnu ulogu, zavisno od vrste adresiranja. Parovi B/C i D/E registra se mogu koristiti kod implicitnog adresiranja da se podaci čitaju ili pišu u akumulator. Par H/L ovo ne dozvoljava, ali se zato može upisati

ili čitati iz akumulatora ako je direktno adresiran. Jedna naredba omogućava direktnu razmjenu sadržaja registara D/E i H/L, kao i razmjenu sadržaja trenutnog pokazivanja stack pointera sa H/L i obrnuto, kao i koiranje sadržaja H/L u IC. Parovi registara mogu biti inkrementirani, dekrementirani i sabrani sa H/L. Tehnika ovih registarskih parova je u modelu 8086 dosta proširena.

Registar stanja

U registar stanja smješteno je 5 od 8 bita. Ovi biti imaju ulogu da kod uslovnog skoka ili kod poziva odluče da li je uslov za skok ili poziv ispunjen. To su: **N** (negativ), **S** (sign), **Z** (zero), **H** (half carry), **P** (parity) i **C** (Carry)

Registar stanja							
7	6	5	4	3	2	1	0
N/S	Z	-	H	-	P	-	C

Biti u registru se postavljaju kod svake logičke i aritmetičke 8-bitne operacije, ali ne i kod naredbe za kopiranje i razmjenu. Kod 16-bitnih naredbi samo naredbe sabiranja u paru H/L postavlja Carry bit. Važno je napomenuti da kod operacija inkrementiranja i dekrementiranja nema ovog Carry bita, kod 8-bitnih operacija prekoračenje može biti provjereno preko Zero bita, a kod 16-bitnih operacija samo preko OR naredbe.

Interapt registar

Upotreba Interapt registra zavisi od toga da li je zahtjev za čitanjem ili pisanjem stigao. Uglavnom ovaj registar služi za provjeru interrupt stanja i maskiranje pojedinog interrupta.

Interapt registar (R/W)							
7	6	5	4	3	2	1	0
-	0	-	R7.5	MSE	M7.5	M6.5	M5.5
-	I7.5	I6.5	I5.5	INTE	M7.5	M6.5	M5.5

Kod pisanja SIM-naredbe (Set Interrupt Mask) u registru šesti bit mora uvijek biti 0 i da se registar promjeni mora treći bit (MSE) biti 1, da bi se preuzeo maskirani interrupt na bitima 0-2.

Interapti i reseti

Za razliku od prethodnika 8080, kod procesora 8085 upravljanje interaptima je unaprijeđeno. Pored INTR (kontroliranog preko interapt kontrolera), procesor 8085 raspolaže sa još 4 interapt ulaza. Vektorski interapti i reseti kod procesora 8085 upravljaju fiksnim adresama i taj se koncept zadržao i kroz naredne generacije procesora.

Procesor 8085 ima 5 ulaza za interrupte: INTR, RST 5.5, RST 6.5, RST 7.5 i TRAP. INTR je zahtjev za prekidom i predstavlja interapt opšte namjene. Uzorkovan je i provjerava se samo za vrijeme zadnjeg clock – impulsa pri završetku instrukcije. Ako je aktivan i nije maskiran programski brojač (PC) će biti zabranjen za inkrementiranje i INTA izlaz će generisati signal. Za vrijeme ovog ciklusa ponovni start (ili CALL

instrukcija) može biti ubačena da bi se skočilo na servisnu rutinu. INTR je softverski omogućen ili onemogućen. Nije poželjno odmah nakon RESET-a ima interrupt.

Svaki od tri RESTART ulaza ima programabilnu masku. TRAP je također prekid, ali ne može biti maskiran. RST prekidi izazivaju interni restart, ako su prekidi omogućeni i ako prekid nije maskiran. TRAP izaziva RESTART nezavisno od stanja interapt enable ili maske. Na status tri RST prekidne maske, može se djelovati samo naredbom SIM i RESET IN.

Interrupti su aranžirani po fiksnom prioritetu koji određuje koji će interrupt biti prepoznat ako je više interrupta u toku. TRAP ima najveći prioritet, pa onda RST 7.5, RST 6.5, RST 5.5 i INTR ima najniži prioritet. Ova šema prioriteta ne uzima u obzir rutinu koju je započeo interrupt većeg prioriteta.

Tabela 4.3:

HARDVERSKI INTERAPTI			
OZNAKA	PRIORITET	ADRESA GRANANJA	TIP TRIGERA
TRAP	1	24H	Rastuća ivica ili log. jedinica
RST 5.5	2	2CH	Rastuća ivica ili log. jedinica
RST 6.5	3	34H	Log. jedinica do semplovanja
RST 7.5	4	3CH	Log. jedinica
INTR	5	programski definisan	Log. jedinica

TRAP je koristan za neočekivane događaje kao što je nestanak napajanja ili greška na sabirnici. TRAP je specifičan po tome što onemogućavaprekide, ali zadržava prethodno omogućene interapt statuse.

Adresiranje

Intel 8085 raspolaže adresnim prostorom od 64 kB za memorisanje i 256 adresa za portove. Ovo se reguliše preko jednog izlaza IO/\overline{M} . Kod pristupanja portu koristi se registar H, a kod memorisanja registar L. Kod pristupa portu mogu se koristiti kako adrese A0-A7, tako i adrese A8-A15. 8 manje značajnih adresnih bita se zajedno sa sabirnicom podataka multipleksiraju, što znači da dijele iste priključke A7-A0. 8 više značajnih adresnih bita imaju vlastite priključke A15-A8. Da bi dao znak da je na sabirnici odgovarajuća adresa, procesor generiše signal ALE (Address Latch Enable). Adresa tada može biti sačuvana u eksternoj logici. Pri tome jedan dio logike (najčešće D flip-flop) preuzima sadržaj A7-A0 i daje 8 manje značajnih bita praznoj adresnoj sabirnici. Preko izlaza S0 i S1 daje se status trenutnog mašinskog ciklusa.

Definisano je pet načina adresiranja:

➤ **Implicitno adresiranje**

- Instrukcija koju je potrebno izvršiti implicitno govori gdje se nalazi operand (podatak nad kojim treba izvršiti određenu operaciju)
- Ovo su 1-bajtni instrukcije i brzo se izvršavaju
- Primjer: ADD M; $(A) \leftarrow (A)+(M)$

➤ **Registarsko adresiranje**

- Operand nad kojim se vrši operacija nalazi se u nekom od internih registara
- I ove instrukcije su 1-bajtni
- Primjer: MOV D, B; $(D) \leftarrow (B)$

➤ **Neposredno adresiranje**

- Operand (podatak) se nalazi u okviru instrukcije
- Vršiti neposredno upisivanje podatka na mjesto odredišta
- Ove instrukcije mogu biti 2-bajtni ili 3-bajtni
- Primjer: MVI B, 65H; $(B) \leftarrow 65H$
LXI B, 1234H; $(B, C) \leftarrow 1234H$

➤ **Direktno adresiranje**

- Mogu biti 2-bajtni ili 3-bajtni instrukcije
- 2. i 3. bajt formiraju adresu u memoriji na kojoj se nalazi operand
- Primjer: LDA 0123H; $(A) \leftarrow [0123H]$
IN 00H; $(A) \leftarrow [00H]$
OUT 01H; $(A) \rightarrow [01H]$

➤ **Indirektno adresiranje**

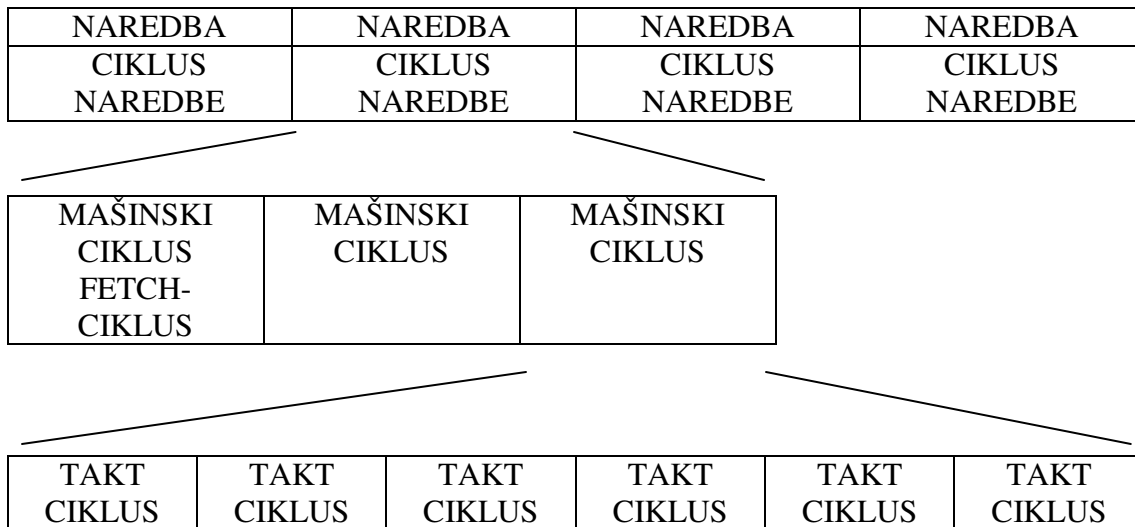
- Adresa operanda (adresa lokacije operanda u memoriji) nalazi se u nekom od nekom paru unutrašnjih registara (B-C, D-E ili H-L)
- Primjer: MOV A, M; $(A) \leftarrow [H-L]$
LDAX B; $(A) \leftarrow [B-C]$

Mašinske naredbe**➤ Nastajenje naredbe**

Jedan asemblerski program se sastoji od niza 8-bitnih naredbi, u posebnim slučajevima od naredbi koje se sastoje od dva bajta koja slijede jedan iza drugog. Izvršavanje je sekvencijalno. Kod dužine riječi od 8 bita moguće je maksimalno 256 različitih naredbi, od kojih se samo 246 može implementirati kod 8085. Kod svake naredbe prvi bajt sadrži kod operacije (Op-Code), tj. operator je. Često je jedan operand, npr. Akumulator, implicitno sadržan tako da je cijela naredba duga jedan bajt. Naredba može biti duga i 2 ili 3 bajta:

- Naredba od 1 bajta: samo kod operacije.
- Naredba od 2 bajta: kod operacije i operand (8-bitna konstanta ili 8-bitna port adresa).
- Naredba od 3 bajta: kod operacije i operand (16-bitna konstanta ili 16-bitna adresa).

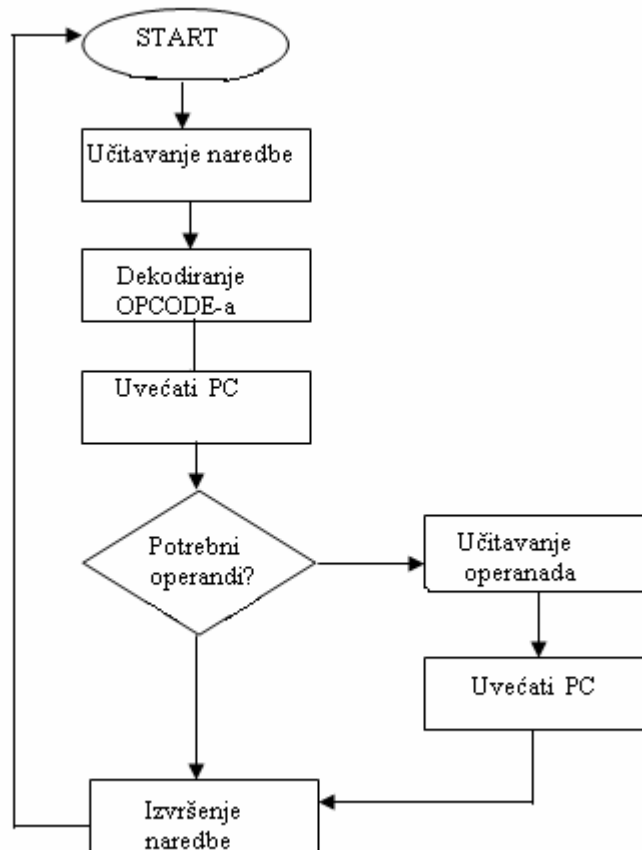
Izvršavanje naredbe odgovara von Neumann-ovoj šemi. Prvo se naredba, na koju pokazuje sadržaj registar brojača naredbi (PC, IC), dobavlja i stavlja u dekodirani naredbi. Tu se naredba dekodira. Jedna naredba zahtijeva jedan do pet mašinskih ciklusa. Jedan mašinski ciklus se sastoji od tri do šest takt ciklusa. (Slika 4.3)



Slika 4.3

➤ Izvršenje naredbe

Dijagram toka izvršenja naredbe prikazan je na slici 4.4



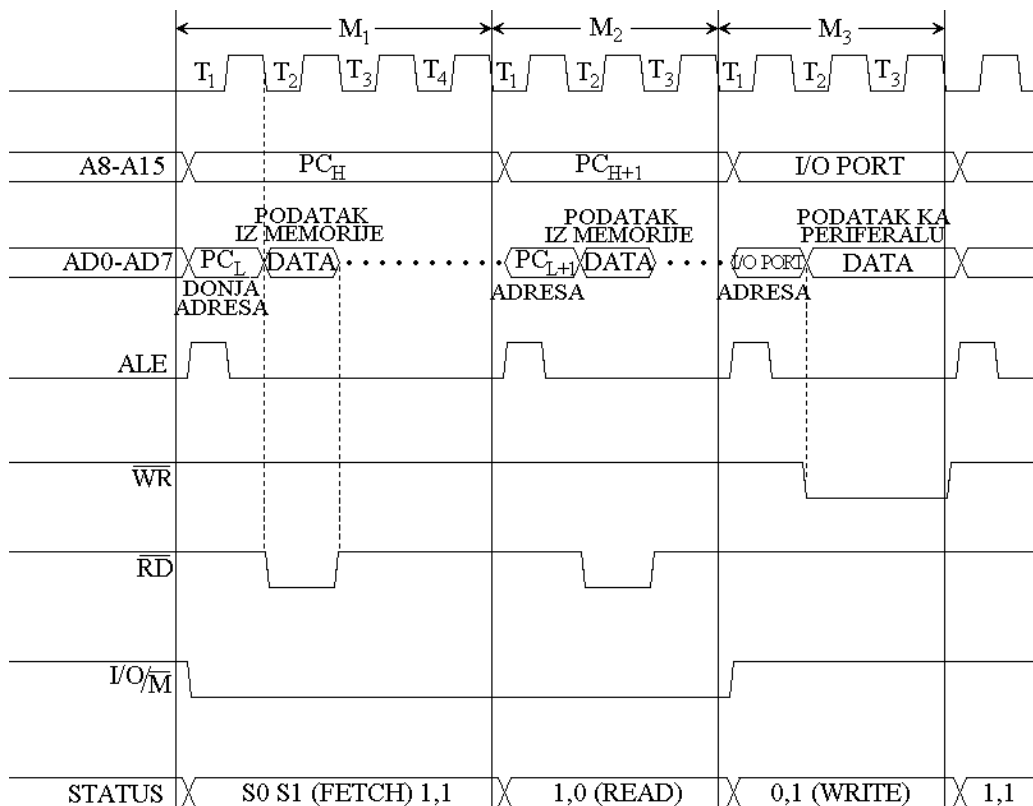
Slika 4.4: Dijagram toka izvršenja naredbe

➤ Ciklus naredbe, mašinski ciklus i takt ciklus

Vrijeme jednog mašinskog ciklusa je 3-6 takt ciklusa kod mikroprocesora. Tipični mašinski ciklusi koji se javljaju u toku ciklusa jedne naredbe su:

1. Poziv naredbe (OPCODE-Fetch)
2. Čitanje memorije (Memory Read)
3. Pisanje u memoriju (Memory Write)
4. Čitanje stack-a (Stack pop)
5. Pisanje na stack (Stack push)
6. Ulaz (Input)
7. Izlaz (Output)

➤ Osnovni tajming sistema i vremenski dijagrami



Slika 4.5: Osnovni tajming sistema

Kao što je već spomenuto mikroprocesor 8085 ima multipleksiranu sabirnicu podataka, tako da se koristi signal ALE za pamćenje, odnosno zadržavanje donjih 8 bita. Mašinski ciklus se sastoji od 3 T-stanja sa izuzetkom fečovanja (prihvatanja) operacionog koda, koji obično ima 4-6 taktova. Svako od T stanja mora biti jedno od deset mogućih prikazanih u tabeli 4.4.

Tabela 4.4

MACHINE CYCLE	STATUS			CONTROL		
	IO/M	S1	S0	RD	WR	INTA
OPCODE FETCH (OF)	0	1	1	0	1	1
MEMORY READ (MR)	0	1	0	0	1	1
MEMORY WRITE (MW)	0	0	1	1	0	1
I/O READ (IOR)	1	1	0	0	1	1
I/O WRITE (IOW)	1	0	1	1	0	1
ACKNOWLEDGE OF INTR (INA)	1	1	1	1	1	0
BUS IDLE (BI): DAD	0	1	0	1	1	1
ACK. OF RST,TRAP	1	1	1	1	1	1
HALT	TS	0	0	TS	TS	1

ALU prima podatke od privremenog registra, akumulatora i od flagova. Rezultat operacije se prenosi ili na internu sabirnicu podataka ili nazad u akumulator. ALU takođe diktira stanje statusnog registra (flip-flova). Svaka operacija koju mikroprocesor može da izvrši definisana je instrukcijskim kodom. Tako za vrijeme fetchovanja ili M_1 (mašinski ciklus 1) procesor čita operacioni kod iz programske memorije i na bazi sadržaja (analizira operacioni kod) zna se tačno kojeg je tipa instrukcija koju treba izvršiti. Instrukcijski kod se smješta (stori) u registar instrukcija, a slijedeći bit u pomoćni registar, nakon čega mikroprocesor pristupa izvršenju adekvatnih operacija. U te svrhe upravljačka logika šalje neophodne signale strukturama unutar i izvan mikroprocesora kojima inicira zahtjevane operacije. Upravljačka logika često mora da reaguje na vanjske signale INTR i WAIT. WAIT se često pojavljuje kada memorija ili U/I elementi rade znatno sporije od mikroprocesora. Tada, npr. logika prisiljava mikroprocesor (preko signala READY) da sačeka dok memorija ili U/I element ne budu spremni sa svojim signalima. Aplikacioni program, građen na skupu neophodnih instrukcija, smješten je u programskoj memoriji (najčešće EPROM) koju adresira mikroprocesor koristeći programski brojač (PC). Kako aplikacioni program predstavlja niz različitih instrukcija koje mikroprocesor mora da izvrši očito je da nakon izvršenja jedne instrukcije mikroprocesor mora znati adresu slijedeće što se postiže pod sredstvom programskog brojača koji ukazuje na memorijsku lokaciju gdje se nalazi instrukcija koju treba izvršiti u slijedećem koraku. Dio RAM-a koristi se kao STACK oblast memorije. Stack nam služi da se zapamti memorijska lokacija na koju se program vraća nakon izvršenja neke rutine ili prekida, kao i parametarsko izvršavanje neke funkcija.

Na slici 4.5 je prikazana instrukcija dobavljanja, ciklus čitanja i I/O pisanja u memoriju. Prilikom I/O ciklusa čitanja i pisanja, I/O adresa porta je kopirana i u gornju i u donju polovinu adrese.

Tabela 4.5:

STANJE MAŠINE	Statusni bitovi i bitovi na sabirnici				Kontrolni bitovi		
	S_1, S_0	IO/\overline{M}	$A_{15}-A_8$	AD_7-AD_0	$\overline{RD}, \overline{WR}$	\overline{INTA}	ALE
T_1	X	X	X	X	1	1	1 ¹⁾
T_2	X	X	X	X	X	X	0
T_{WAIT}	X	X	X	X	X	X	0
T_3	X	X	X	X	X	X	0
T_4, T_5, T_6	1	0 ²⁾	X	TS	1	1	0
T_{RESET}, T_{HOLD}	X	TS	TS	TS	TS	1	0
T_{HALT}	0	TS	TS	TS	TS	1	0

Napomene:

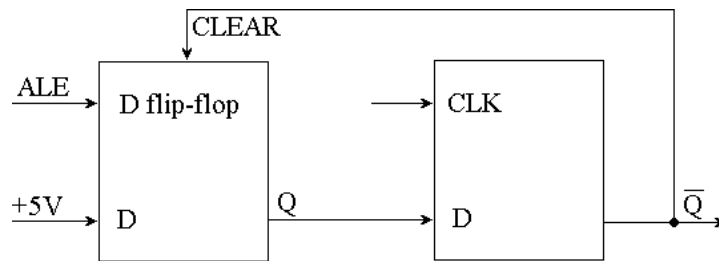
- 1) ALE nije generisan za vrijeme drugog i tećeg mašinskog ciklusa instrukcije DAD (decimalno podesi).
- 2) $IO/\overline{M} = 0$ za vrijeme T_4-T_6 instrukcije INTA.

Definisano je sedam tipova mogućih mašinskih ciklusa. Koji će mašinski ciklus nastupiti zavisi od kombinacija tri statusne linije IO/\overline{M} , S_1 i S_0 i tri kontrolna signala \overline{RD} , \overline{WR} i

\overline{INTA} , (tabela 4.5). Statusne linije mogu biti korištene kao dodatne kontrole (npr. za selekciju uređaja), kada \overline{RD} i \overline{WR} postanu aktivne na T_1 stanju, na početku svakog mašinskog ciklusa. Kontrolne linije \overline{RD} i \overline{WR} postanu aktivne kasnije, kada je vrijeme da počne prenos podatka, kada se koriste kao komandne linije.

➤ Generisanje 8085a stanja čekanja

Ako su zahtjevi sistema takvi da usporavaju memoriju ili se koriste periferalni uređaji, kolo prikazano na slici može biti korišteno da ubaci WAIT stanje u svaki 8085 mašinski ciklus.



Slika 4.6: Kolo za generisanje WAIT stanja

READY linija se koristi da produži trajanje read i write impulsa tako da se može koristiti i kod sporih memorija. HOLD prouzrokuje da CPU napusti sabirnicu stavljajući adrese i sabirnicu podataka u treće stanje.

➤ Naredbe mikroprocesora 8085

Tabela naredbi

OZNAKA	DUŽINA	TRAJANJE	FUNKCIJA
PREBACIVANJE SADRŽAJA IZ REGISTRA U REGISTAR			
MOV r1,r2	1	4	r1,r2 = A,B,C,D,E,H ili L puni registar r1 sadržajem registra r2.
XCHG	1	4	Zamijeniti sadržaj para registara (D,E) i (H,L)
XTHL	1	16	Zamijeniti sadržaj para (H,L) i sadržaj riječi koja je adresirana preko SP.
SPHL	1	6	Puni SP sa sadržajem para registara (H,L).
PREBACIVANJE SADRŽAJA IZ PERIFERALA U REGISTAR			
MOV r1,M	1	7	Puni registar r1 sa sadržajem memorijskog bajta M koji je adresiran sadržajem para registara (H,L).
LDA adr	3	13	Puni akumulator sadržajem adrese adr.
LDAX rp	1	7	rp= B,D Puni akumulator sadržajem memorijske ćelije koja je adresirana sadržajem registarskog para rp.
LHLD adr	3	16	Puni registarski par sadržajem adrese adr i adr+1.
POP rp	1	10	rp= B,D,H,PSW U registarski par rp se smještata riječ , koja je adresirana preko stack pintera.
IN nr	2	10	Akumulator se puni sadržajem ulaznog kanala (broj nr<256).
KONSTANTA U REGISTARSKI PAR			
LXI rp,adr	3	10	rp=B,D,H,SP Puni registarski par rp sa vrijednošću adr.
PREBACIVANJE SADRŽAJA IZ REGISTARA U PERIFERAL			
MOV M,r1	1	7	r1 =A;B;C;D;E;H ili L Sadržaj registra r1 staviti na mjesto adresirano preko registarskog para (H,L).
STA adr	3	13	Sadržaj akumulatora spasiti na adresu adr.
STAX rp	1	7	rp=B,D

			Akumulator memorisati na bajtu koji je adresiranim sadržajem registarskog para rp.
SHLD adr	3	16	Registarski par (H,L) spasiti na adresi adr i adr+1;
PUSH rp	1	12	rp=B,D,H,PSW Sadržaj registarskog para rp bit će prenesen u riječ na koju pokazuje stack pointer;
OUT nr	2	10	Akumulator će biti "izdat" izlaznom kanalu (broj<256)
SMJEŠTANJE KONSTANTE U REGISTRAR			
MVI M,konst	2	10	Puni memoriju koja je adresirana sadržajem registarskog para (H,L) sa konstantom; (konst= Konstanta< 256)
MVI r1,konst	2	7	r1=A,B,C,D,E,G,H puni registar konstantama (konst < 256)
ARITMETIČKE OPERACIJE			
INR r1	1	4	r1=A,B,C,D,E,F,H ili L Sadržaju registra r1 dodaje se 1;
INR M	1	10	Sadržaju bajta adresiranog preko registarskog para dodaje se 1;
DCR r1	1	4	r1=A,B,C,D,E,F,H ili L Od sadržaja registra r1 se oduzima 1;
DCR M	1	10	Sadržaju bajta adresiranog preko registarskog para oduzima se 1;
INX rp	1	6	rp=B,D,H,SP Sadržaj registarskog para rp se uvećava za 1;
DCX rp	1	6	rp=B,D,H,SP Sadržaj registarskog para rp se smanjuje za 1;
ADD r1	1	4	r1=A,B,C,D,E,F,H ili L Sadržaj r1 se dodaje sadržaju akumulatora.
ADD M	1	7	Sadržaj memorijskog bajta (adresiranog preko registarskog para) se dodaje akumulatoru.
ADC r1	1	4	r1=A,B,C,D,E,F,H ili L Sadržaj registra r1 i Carry bita se dodaju akumulatoru.

ADC M	1	7	Sadržaj memorijskog bajta i i Carry bita se dodaju sadržaju akumulatora.
DAD rp	1	10	rp=B,D,H,SP: Sadržaj registarskog para rp i registarskog para (H,L) se sabiraju. Rezultat je u (H,L).
SUB r1	1	4	r1=A,B,C,D,E,H ili L: Sadržaj registarskog para i akumulatora se oduzimaju.
SUB M	1	7	Sadržaj memorijskog bajta se oduzima od akumulatora.
SBB r1	1	4	r1=A,B,C,D,E,H ili L Sadržaj registra r1 i carry bita se oduzimaju od sadržaja akumulatora.
SBB M	1	7	Sadržaj memorijskog bajta i sadžaj Carry-Bita oduzimaju se od akumulatora.
ADI konst	2	7	Konstanta se dodaje sadržaju akumulatora.
ACI konst	2	7	Sadržaju akumulatora se dodaje konstanta i Carry-bit.
SUI konst	2	7	Konstanta se oduzima od sadržaja akumulatora.
SBI konst	2	7	Od sadržaja akumulatora se oduzima konstanta i Carry-bit.
DAA	1	7	Sadržaj akumulatora se pretvara u dvocifren broj.
LOGIČKE OPERACIJE			
CMA	1	4	Negacija akumulatora.
ANA r1	1	4	r1=A,B,C,D,E,H ili L Sadržaj akumulatora i registra r1 se povezuju I vezom.
ANA M	1	7	Sadržaj bajta adresiranog preko para (H,L) i sadržaj akumulatora se povezuju I vezom.
ANI konst	2	7	Sadržaj akumulatora i konstanta se povezuju I vezom.
ORA r1	1	4	r1=A,B,C,D,E,H ili L Sadržaj akumulatora i registra r1 se povezuju ILI vezom.
ORA M	1	7	Sadržaj bajta adresiranog preko para (H,L) i sadržaj akumulatora se povezuju ILI

			vezom.
ORI konst	2	7	Sadržaj akumulatora i konstanta se povezuju ILI vezom.
XRA r1	1	4	r1=A,B,C,D,E,H ili L Sadržaj akumulatora i registra r1 se povezuju ekskluzivnom ILI vezom.
XRA M	1	7	Sadržaj bajta adresiranog preko para (H,L) i sadržaj akumulatora se povezuju ekskluzivnom ILI vezom.
XRI konst	2	7	Sadržaj akumulatora i konstanta se povezuju ekskluzivnom ILI vezom.
CMP r1	1	4	Akumulator se poredi sa sadržajem registra. Ako su jednaki, postavlja se Zero flag.
CMP M	1	7	Akumulator se poredi sa sadržajem bajta adresiranog preko para (H,L).
CPI konst	2	7	Akumulator se poredi sa konstantom (const<256).
ROTACIJA AKUMULATORA			
RLC	1	4	Sadržaj akumulatora se ciklično pomjera na lijevo za jedan bit. Bit 2^7 upisuje se u Carry-Bit. Bit $2^0 = \text{Bit } 2^7$
RRC	1	4	Sadržaj akumulatora se ciklično pomjera na desno za jedan bit.Bit 2^0 upisuje se u Carry-Bit. Bit $2^7 = \text{Bit } 2^0$
RAL	1	4	Sadržaj akumulatora se ciklično pomjera na desno za jedan bit.Bit 2^0 upisuje se u Carry-Bit, a Carry-Bit u 2^7 .
RAR	1	4	Sadržaj akumulatora se ciklično pomjera na lijevo za jedan bit. Bit 2^0 upisuje se u Carry-Bit, a Carry-Bit u bit 2^0 .
POSTAVKE CARRY-BITA			
CMC	1	4	Carry-Bit se negira.
STC	1	4	Carry-Bit se postavlja.
BEZUSLOVNI SKOKOVI			
PCHL	1	6	Program se nastavlja od adrese koja stoji u registarskom paru (H,L).
JMP adr	3	10	Program se nastavlja od adrese adr.

USLOVNI SKOKOVI			
JC adr	3	7 / 10	Kod Carry-Bit = 1 program se nastavlja od adrese adr.
JNC adr	3	7 / 10	Kod Carry-Bit = 0 program se nastavlja od adrese adr.
JZ adr	3	7 / 10	Kod Zero-Bit = 1 program se nastavlja od adrese adr.
JNZ adr	3	7 / 10	Kod Zero-Bit = 0 program se nastavlja od adrese adr.
JM adr	3	7 / 10	Kod Sign-Bit = 1 program se nastavlja od adrese adr.
JP adr	3	7 / 10	Kod Sign-Bit = 0 program se nastavlja od adrese adr.
JPE adr	3	7 / 10	Kod Parity-Bit= 1 program se nastavlja od adrese adr.
JPO adr	3	7 / 10	Kod Parity-Bit= 0 program se nastavlja od adrese adr.
POZIVI UNUTAR PROGRAMA			
CALL adr	3	18	Program se nastavlja od adrese adr.
CC adr	3	9 / 18	Kod Carry-Bit = 1 program se nastavlja od adrese adr.
CNC adr	3	9 / 18	Kod Carry-Bit = 0 program se nastavlja od adrese adr.
CZ adr	3	9 / 18	Kod Zero-Bit = 1 program se nastavlja od adrese adr.
CNZ adr	3	9 / 18	Kod Zero-Bit = 0 program se nastavlja od adrese adr.
CM adr	3	9 / 18	Kod Sign-Bit = 1 program se nastavlja od adrese adr.
CP adr	3	9 / 18	Kod Sign-Bit = 0 program se nastavlja od adrese adr.
CPE adr	3	9 / 18	Kod Parity-Bit= 1 program se nastavlja od adrese adr.
CPO adr	3	9 / 18	Kod Parity-Bit= 0 program se nastavlja od adrese adr.
RST konst	1	9 / 18	Program se nastavlja kod adrese 8 x const (0-7).

SKOK UNUTAR PROGRAMA			
RET	1	10	Program se nastavlja na adresi koja je u riječi na koju pokazuje stack pointer.
RC	1	6 / 12	Carry-Bit = 1 Program se nastavlja na adresi koja je u riječi na koju pokazuje stack pointer.
RNC	1	7 / 12	Carry-Bit = 0 Program se nastavlja na adresi koja je u riječi na koju pokazuje stack pointer.
RZ	1	8 / 12	Zero -Bit = 1 Program se nastavlja na adresi koja je u riječi na koju pokazuje stack pointer.
RNZ	1	9 / 12	Zero -Bit = 0 Program se nastavlja na adresi koja je u riječi na koju pokazuje stack pointer..
RM	1	10 / 12	Sign -Bit = 1 Program se nastavlja na adresi koja je u riječi na koju pokazuje stack pointer.
RP	1	11 / 12	Sign -Bit = 0 Program se nastavlja na adresi koja je u riječi na koju pokazuje stack pointer.
RPE	1	12 / 12	Parity-Bit= 1 Program se nastavlja na adresi koja je u riječi na koju pokazuje stack pointer.
RPO	1	13 / 12	Parity-Bit= 0 Program se nastavlja na adresi koja je u riječi na koju pokazuje stack pointer.
PREKID PROGRAMA			
EI	1	4	Interrupt-Flipflop se postavlja; Mikroprocesor može preuzeti jedan zahtjev za prekidom.
DI	1	4	Interrupt-Flipflop se ukida. Mikroprocesor ignorišeignoriert zahtjev za prekidom.
NAREBE ZA MASKIRAJUĆI BIT			
RIM	1	4	Maska za prekid za čitanje i serijski ulaz za akumulator.
SIM	1	4	Postavlja masku za prekid i serijski izlaz.

OSTALE NAREDBE			
HLT	1	5	Program čeka dok ne dođe zahtjev za prekidom.
NOP	1	4	Prazna naredba. (Nema operacije.)