

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku

GRAĐA RAČUNALA
(predavanja u ak. god. 2005./2006.)

doc.dr.sc. Goran Martinović
www.etfos.hr/~martin
goran.martinovic@etfos.hr

Tel: 031 224-766

Soba: 0-15

Osijek, 2006.

1

MIKROPROCESOR Z80

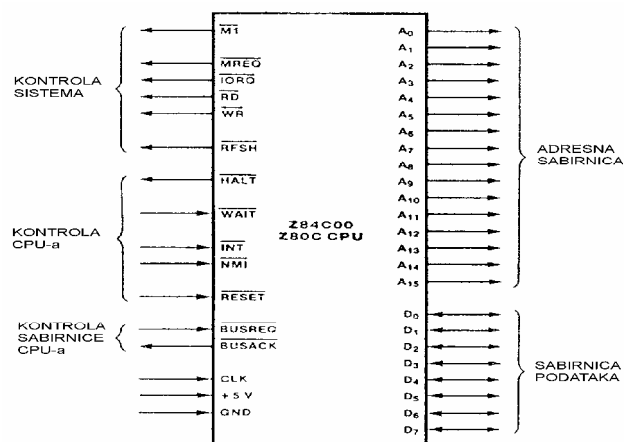
2

ZNAČAJKE :

- Niska potrošnja energije
- Izrađuje se za dvije brzine (do 2.5 MHz i do 4 MHz)
- Potrebno jedno napajanje + 5 V
- Ima dvostruki set registara opće namjene
- Ima statusne registre
- Ima dva 16-bitna indeksna registra
- Mikroprocesor i pridružena obitelj periferije može biti povezana vektorskim prekidnim sastavom
- Postoje 3 načina rukovanja maskirajućim prekidom
- Ima logiku za dinamičko osvježavanje memorije

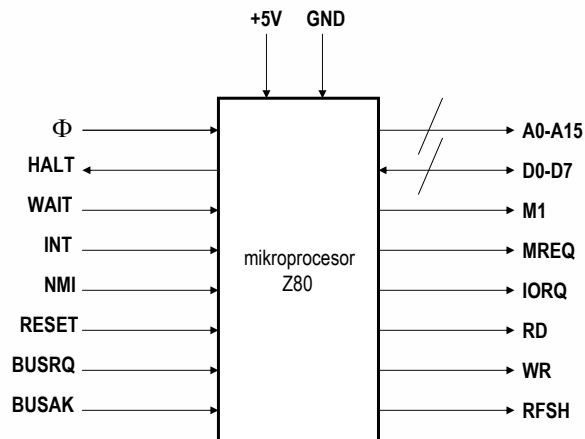
3

FUNKCIJE PINOVA NA MIKROPROCESORU



4

Vanjski priključci mikroprocesora Z80



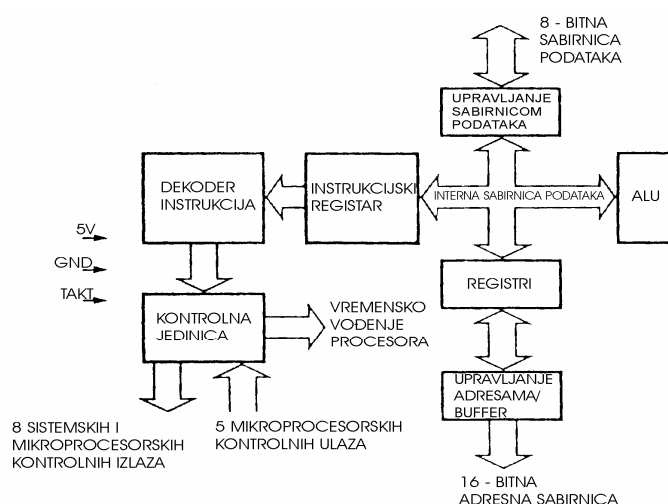
OPIS PINOVA

- A0 - A15 - Address Bus - adresna sabirnica (izlazi, aktivni u "1", mogu imati 3 stanja: "0", "1" i stanje visoke impedancije). A0 - A15 obrazuju 16-bitnu adresnu sabirnicu. Ona adresira u memoriji (do 64 Kbajta) ili ulazno/izlaznim jedinicama podatke za prijenos preko sabirnice podataka.
- BUSACK - Bus Acknowledge - potvrda sabirnice (izlaz, aktivan u "0"). Ovaj pin se aktivira odlaskom adresne sabirnice, sabirnice podataka i kontrolnih signala MREQ, IORQ, RD i WR u stanje visoke impedancije.
- BUSREQ - Bus Request - zahtjev za sabirnicom (ulaz, aktivan u "0"). Dolaskom "0" na ovaj pin zahtjeva se da adresna sabirnica, sabirnica podataka i kontrolni signali MREQ, IORQ, RD i WR odu u stanje visoke impedancije da bi ih mogli kontrolirati vanjski krugovi.
- D0 - D7 - Data Bus - sabirnica podataka (ulaz/izlaz, aktivan u "1", može imati 3 stanja). Ovi pinovi obrazuju 8-bitnu dvosmjernu sabirnicu podataka, koja se koristi za izmjenu podataka s memorijom i U/I sklopovima.
- HALT - Halt state - stanje zaustavljanja (izlaz, aktivan u "0"). Pin se aktivira izvršavanjem instrukcije zaustavljanja.
- INT - Interrupt Request - zahtjev za prekidom (ulaz, aktivan u "0"). Generiraju ga ulazno/izlazni krugovi.
- IORQ - Input/Output Request - zahtjev za ulaz/izlaz (izlaz, aktivan u "0", može imati 3 stanja). Aktiviranjem se pokazuje da donja polovica adresne sabirnice sadrži U/I adresu za operaciju čitanja ili upisivanja podataka.

- M1 - Machine Cycle One - strojni ciklus jedan (izlaz, aktivan u "0"). M1 zajedno s IORQ potvrđuje prekid, a M1 zajedno s MREQ pokazuje da je u toku izvršavanje faze pribavljanja.
- MREQ - Memory Request - zahtjev za memorijom (izlaz, aktivan u "0", može imati 3 stanja). Aktiviranjem se pokazuje da adresna sabirnica sadrži adresu memorijske lokacije za operaciju čitanja ili upisivanja.
- NMI - Non - Maskable Interrupt - nemaskirajući prekid (ulaz, okidan padajućim bridom). Automatski se prisiljava CPU na ponovni start na lokaciji 0066H.
- RD - Read - čitaj (izlaz, aktivan u "0", može imati 3 stanja). Pojavljuje se kada CPU želi da očita podatke ili iz memorije ili iz ulazno/izlaznih krugova.
- RESET - Reset (ulaz, aktivan u stanju "0"). Resetira CPU.
- RFSH - Refresh - osvježavanje (izlaz, aktivan u "0"). RFSH zajedno s MREQ pokazuje da se 7 nižih bitova adresne sabirnice mogu koristiti kao adrese osvježavanja za sistemsku dinamičku memoriju.
- WAIT - Wait - čekaj (ulaz, aktivan je u "0"). Sve dok je signal aktivan pokazuje se CPU-u da adresirana memorija ili U/I krugovi nisu spremni za prijenos podataka.
- WR - Write - piši (izlaz, aktivan u "0", može imati 3 stanja). Aktiviranjem se pokazuje da sabirnica podataka CPU -a sadrži podatak za spremanje na adresiranu memoriju ili U/I lokaciju.

7

BLOK DIJAGRAM MIKROPROCESORA Z 80 C CPU



8

NAČIN RADA MIKROPROCESORA Z 80 C CPU

- Program koji želimo da CPU obavi nalazi se spremljen u memoriji koja nije dio CPU-a. Cijela memorija je adresibilna tako da je svakoj 8-bitnoj memorijskoj lokaciji (od 65536 lokacija) pridružena neka adresa. Mikroprocesor je napajan s 5 V i dovodi mu se takt. Program se počinje obavljati postavljanjem sadržaja 16-bitnog programskog brojila na adresnu sabirnicu. Ujedno se šalje i upravljački signal RD (očitaj). Memorijski sklop dekodira postavljenu adresu i omogućava pristup mikroprocesoru sadržaju memorijske lokacije preko 8-bitne sabirnice podataka. Taj se sadržaj sprema u CPU, u instrukcijski registar, i to je operacijski kod instrukcije.

9

NAČIN RADA MIKROPROCESORA Z 80 C CPU

- Nakon što je programsko brojilo poslalo na adresnu sabirnicu adresu memorijske lokacije prve instrukcije brojilo se inkrementira i čeka da prenese adresu druge instrukcije na adresnu sabirnicu. Spremljeni operacijski kod instrukcije u instrukcijskom registru se dekodira i kontrolna jedinica generiranjem upravljačkih signala aktivira pojedine sklopove da bi se izvršila instrukcija (npr. prijenos podataka u akumulator ili registre opće namjene ili upisivanje u memoriju ili prijenos podataka na U/I jedinicu, ili npr. aktiviranje ALU - aritmetičke - logičke jedinice za obavljanje zbrajanja, oduzimanja, inkrementiranja, izvršavanja logičkog I, ILI, EX ILI, itd.). Isto tako kontrolna jedinica može upravljati perifernim jedinicama u ovisnosti o programu, a isto tako i one mogu upravljati CPU-om preko kontrolnih ulaza i izlaza.
- Nakon obavljanja prve instrukcije programsko brojilo ponovno postavlja sadržaj (koji je inkrementiran u odnosu na prvi put) na sabirnicu podataka. Iz memorije se uzima slijedeća instrukcija i program se nastavlja.

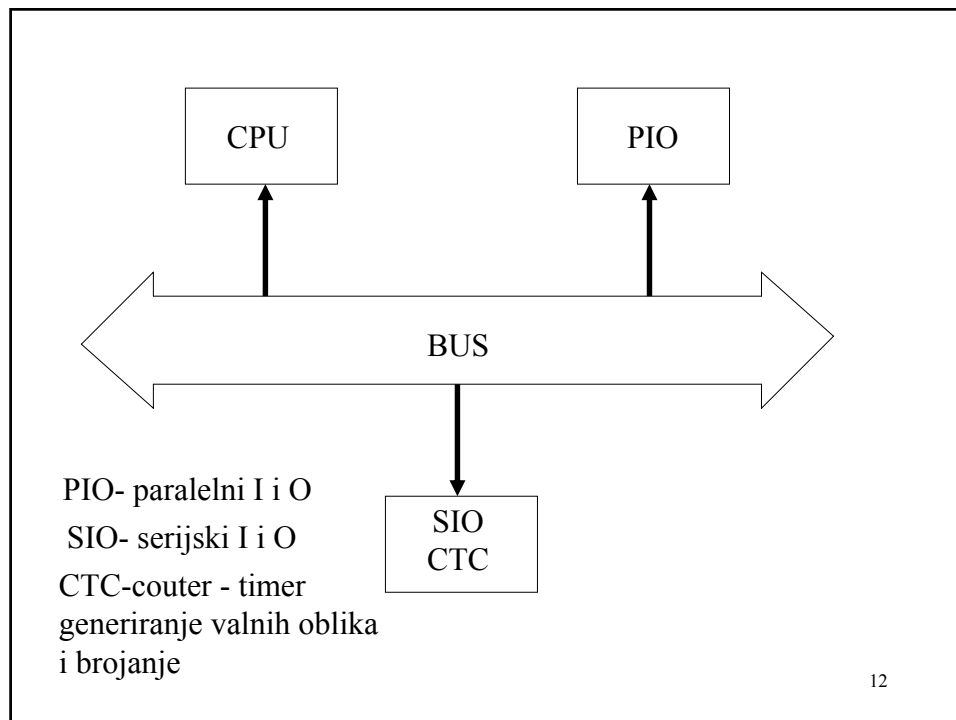
10

**Z 80 C CPU JE 8-BITNI MIKROPROCESOR KOJI IMA
NAJKOMPLETNIJU OBITELJ POPRATNIH
SKLOPOVA.**

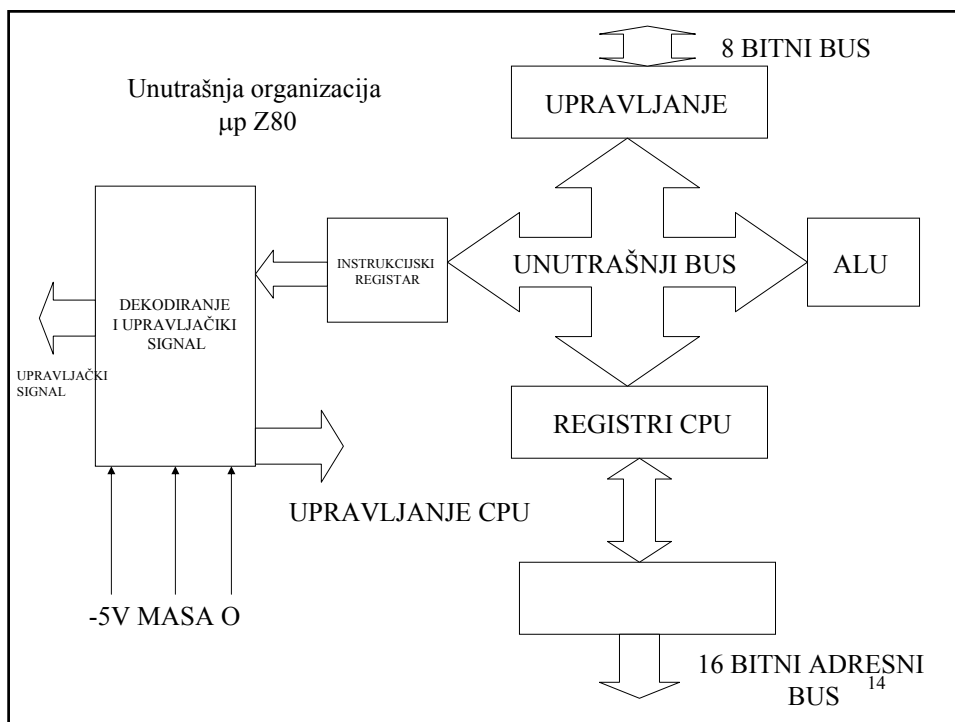
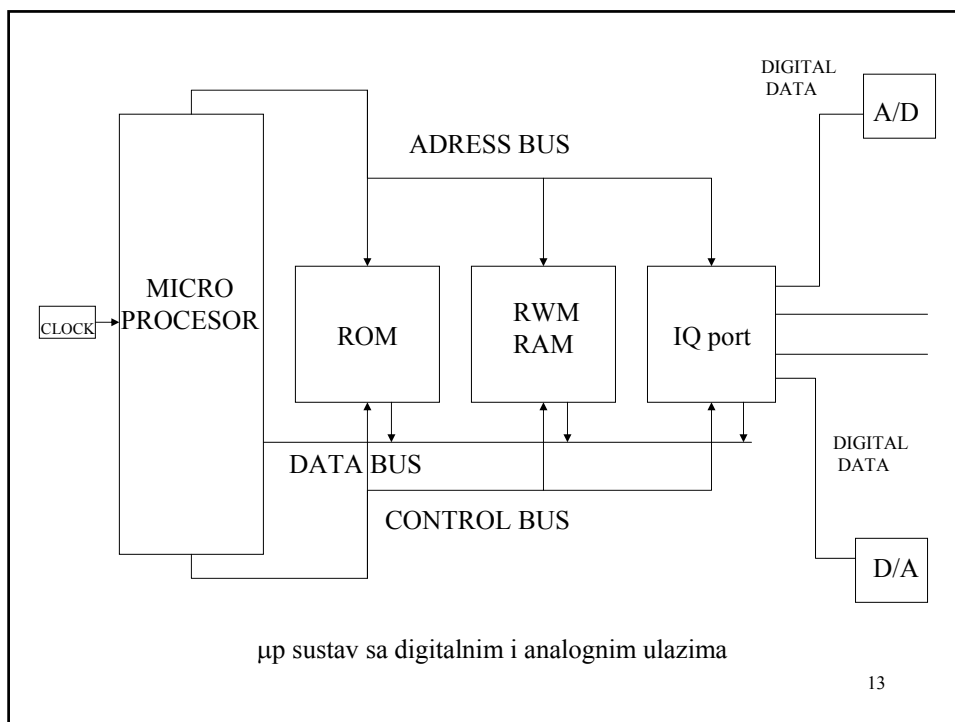
za potporu mikroprocesoru razvijeni su:

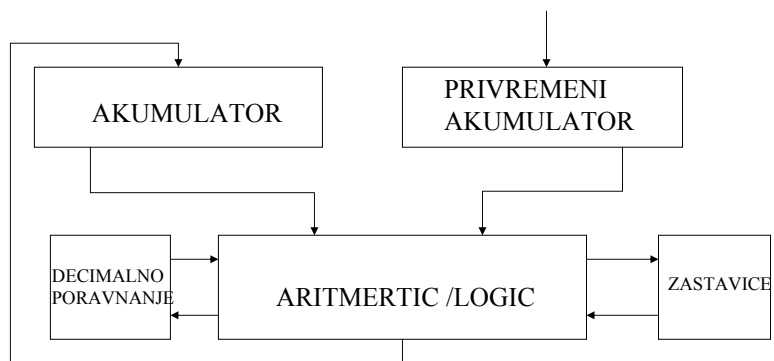
- PIO (Parallel Input / Output) - paralelni ulaz / izlaz
- CTC (Counter / Timer Circuit) - brojač / vremenski krug
- DMA (Direct Memory Access) - izravan pristup memoriji
- SIO (Serial Input / Output) - serijski ulaz / izlaz
- DART (Dual Asynchronous Receiver / Transmitter) - dualni asinkroni prijemnik / prijenosnik

11



12





ARITMETRIČKO LOGIČKA JEDINICA

15

PREGLED REGISTRARA U Z 80 C CPU

OSNOVNI SKUP

POMOĆNI SKUP

A AKUMULATOR	F STATUS REGISTRAR	A' AKUMULATOR	F' STATUS REGISTRAR
B OPĆE NAMJENE	C OPĆE NAMJENE	B' OPĆE NAMJENE	C' OPĆE NAMJENE
D OPĆE NAMJENE	E OPĆE NAMJENE	D' OPĆE NAMJENE	E' OPĆE NAMJENE
H OPĆE NAMJENE	L OPĆE NAMJENE	H' OPĆE NAMJENE	L' OPĆE NAMJENE

← 8 BITA

→

SKUP POSEBNE
REGISTARA NAMJENE

INDEKSNI	REGISTAR IX
INDEKSNI	REGISTAR IY
POKAZIVAČ	STOGA SP
PROGRAMSKO	BROJLO PC
PREKIDNI VEKTOR I	OSVJEŽAVANJE MEMORIJE R

← 16 BITA →

16

OPIS REGISTARA

- Prethodna slika prikazuje 2 grupe registara:
Prva grupa se sastoji od glavnog i pomoćnog seta registara, a druga od 6 registara s pridruženim funkcijama.
- A i A' su akumulatori svaki dužine 8 bita. Spremaju operanda ili rezultat operacije. Omogućavaju brz prijenos podataka između mikroprocesora i U/I međusklopova.
- B, B', C, C', D, D', E, E', H, H', L, L' su registri opće namjene svaki dužine 8 bita. Služe kao skup registara za prijevremeno pohranjivanje podataka, međurezultata i rezultata. Parovi registara BC, DE, HL, B'C', D'E', H'L' se mogu upotrebljavati kao 16 bitna kazala podataka.
- I je registar prekidnog vektora, a upotrebljava se za pohranjivanje adrese stranice u kojoj se nalazi adresa prekidne rutine.
- R je registar za osvježavanje dinamičke memorije. 7 manje značajnih bitova registra predstavlja brojilo osvježavanja koje se automatski inkrementira nakon svakog pribavljanja instrukcije i omogućava osvježavanje sadržaja slijedeće memorijske lokacije.
- IX i IY su indeksni registri za indeksirani način adresiranja.
- SP je pokazivač stoga koje omogućava izvedbu stoga u bilo kojem dijelu adresibilne memorije.
- PC je programsko brojilo koje sadrži adresu slijedeće instrukcije.
- F i F' su statusni registri. Oni sadržavaju zastavice.

17

Registar uvjeta F (i F') se sastoji od slijedećih zastavica F (Flag zastavica):

S	Z		P/V		H	N	C
---	---	--	-----	--	---	---	---

- S - Sign flag - zastavica predznaka.
Sadrži vrijednost najznačajnijeg bita rezultata aritmetičke ili logičke operacije (rezultat pozitivan s= "0").
- Z - Zero flag - zastavica nule
Postavlja se u "1" kada neka aritmetička ili logička operacija daje rezultat "0".
- P/V - Parity or overflow flag - zastavica parnosti ili preliva.
Kod aritmetičkih operacija ima funkciju zastavice preliva.
Za ulaznu instrukciju IN r, (C), kružni posmak i logičke operacije ima funkciju zastavice parnosti (p = "1" ako je broj jedinica u rezultatu paran).
- H - Half - carry flag - zastavica polovičnog (pomoćnog) prijenosa.
Pohranjuje bit prijenosa iz b3 u b4 bita rezultata prilikom izvršavanja aritmetičke instrukcije. Polovični prijenos se upotrebljava u aritmetici BCD.
- N - Add / Subtract flag - zastavica oduzimanja.
Postavlja se u "1" za sve instrukcije oduzimanja, a sve instrukcije zbrajanja je resetiraju.
- C - Carry / Link flag - zastavica prijenosa.
Pohranjuje bit kod aritmetičkog prijenosa (prijenos sa najznačajnijeg mjesta) i upotrebljava se kod operacija posmaka.

Četvrti i šesti bit se ne koriste u statusnom registru.

18

Programski model μp Z-80

Akumulator A	Registar zastavica F	Akumulator A'	Registar zastavica F'	registri opće namjene
B	C	B'	C'	
D	E	D'	E'	
H	L	H'	L'	

19

vektor prekida programa	osvježavanje memorije R	Registri posebne namjene
indeks registar X		
indeks registar Y		
pokazivač adrese stoga SP		
programsko brojilo PC		
registar sa zastavicama		

20

- A-akumulator-pomoću njega izvodi se većina operacija i u njemu ostaju rezultati 8 bitnih aritmetičkih i logičkih operacija
- F-registar sa zastavicama-gdje su zastavice pokazivači svojstava rezultata dobivenog izvođenjem instrukcije
- 6 zastavica pokazuje stanja aritmetičke logičke jedinice (ALU)
 - S-predznak rezultata-0 ako je pozitivan
1 ako je negativan
 - Z-pokazuje kada je rezultat jednak 0 i u tome se slučaju postavlja stanje 1
 - H-poluprijenos i koristi se kod instrukcija DAA ako se pojavi prijenos sa nižeg na viši polubit
 - P/V-pokazuje kada aritm. Instrukcija ima značenje preljeva ili prijenosa, a preljev se pojavljuje kada rezultat izlazi između +127 i -127. Kod logičkih instrukcija ovaj bit označava prioritet. Za paran broj zastavica ima stanje 1
 - N- označava dali je izvršena operacija oduzimanja pri čemu je postavljena u stanje 1
 - C-sluzi za spremanje prijenosa s najznačajnijeg bita ALU koji se može pojaviti pri +, -, log, operacijskom pomaku i rotiranju

21

- Izbor između glavnog i pomoćnog skupa registara izvodi se jednostavno instrukcijama za razmjenu a koriste se kod prekidnog načina rada.
- Registri B,C,D,E,H,L su registri opće namjene
- Mogu se koristiti kao 6 nezavisnih registara ili kao 3 registarska para od 16 bita (BC,DE,HL)
- Služe za adresiranje vanjskih jedinica (C)
- za pohranjivanje 16-bitne adrese u memoriju (HL)
- za razmjenu 16-bitnih podataka (DL,HL)

22

Registri posebne namjene su PC, SP, IX, IY, I, R

- PC (program counter)- programsko brojilo i sadrži adresu memorijske lokacije na kojoj se nalazi naredba koja će se sljedeća dohvatiti i izvršiti
- SP (stack pointer)- 16 bitni registar. služi za spremanje podataka na stog. Stog predstavlja dio memorije u koji se podaci pohranjuju kao u pomični registar tako da novi podatak potiskuje stari podatak dublje u stog. Radi po LIFO principu
- IX, IY- se koriste za indeksno adresiranje. U registar se upisuju početne adrese memorije bloka u koji se želi pohranjivati podatak ili ih dohvati. Dodatni bajt opisuje odstupanje od početne adrese.
- I-adresni registar stranice prekida programa. Prilikom vektorskog prekidnog rada omogućava adresiranje bilo koje memorijske lokacije na kojoj se nalazi prekidna rutina
- R (refresh)- registar služi za zadržavanje pri osvježavanju dinamičke memorije

23

- Flags(zastavice) pokazuju da je procesor u određenom stanju
- Procesor je električni uređaj čiji temeljni rad da na nožicama registrira i mijenja 0 i 1
- μ p razumije (može) samo binarni oblik programa
- Ako se program vrši u višem programskom jeziku od strojnog i izvršni oblik programa se također prevodi u binarni
- Što je programski jezik prihvatljiviji za korisnika udaljeniji je od strojnog jezika koji je brži

24

ZASTAVICE(FLAGS)



↑
prazne

↑
prazne

Z-zero

S-sign

C-carry

N-add/ subtract

P/V-parity/overflow

25

+5V,GND – (ulazni) napajanje mikroprocesora

Φ – (ulazni) signal vremenskog vođenja sustava – takt (uobičajeno 4 MHz, engl. clock)

SABIRNICA ADRESA (engl. ADDRESS BUSS)

A0-A15 – (izlazni) adresiranje memorije i vanjskih jedinica

SABIRNICA PODATAKA (engl. DATA BUS)

D0-D7 – (dvosmjerni) podatak koji se čita ili piše u memoriju ili vanjsku jedinicu

KONTROLA SUSTAVA (engl. SYSTEM CONTROL)

M1 – (izlazni) označava da je trenutni strojni ciklus dohvat operacijskog koda iz memorije, kada je M1 aktivan zajedno sa IORQ, označava ciklus potvrde zahtjeva za prekid

MREQ – (izlazni) označava da se na sabirnici adresa nalazi adresa za čitanje ili pisanje u memoriju

IORQ – (izlazni) označava da se na nižih 8 bita sabirnice adresa nalazi adresa vanjske jedinice

RD – (izlazni) označava ciklus čitanja iz memorije ili vanjske jedinice

WR – (izlazni) označava ciklus pisanja iz memorije ili vanjske jedinice

RFSH – (izlazni) označava da 7 nižih bita sabirnice adresa sadrže adresu osvježavanja za dinamičke memorije. Bit A7 je u nuli, a na gornjih 8 bita upisan je sadržaj registra I.

26

KONTROLA MIKROPROCESORSKE JEDINICE (engl. CPU CONTROL)

HALT - (izlazni) označava da je Z80 izveo naredbu HALT i da čeka prekid (obični ili nemaskirani). Dok je procesor u stanju HALT, izvode se naredbe NOP, kako bi se osvježavale dinamičke memorije.

WAIT - (ulazni) dojava procesoru da adresirana memorija ili vanjska jedinica nije spremna za prijenos podataka. Dok je taj signal aktivan, procesor umeće stanja čekanja. Taj signal omogućava upotrebu memorijskih i vanjskih uređaja bilo koje brzine i asinkronu komunikaciju s procesorom.

INT - (ulazni) zahtjev za prekid. Postavlja ga neka vanjska jedinica. Zahtjev će biti obrađen na kraju tekuće naredbe ako je prekid omogućen.

NMI - (ulazni, aktivan na padajući brid) nemaskirani zahtjev za prekid. Slično kao i običan zahtjev za prekid, ali ima viši prioritet i ne može se maskirati (onemogućiti). Zahtjev će biti uvijek obrađen na kraju tekuće naredbe, a obrada se izvodi uvijek na isti način.

RESET - (ulazni) postavljanje mikroprocesora u početno stanje. Registri PC, I i R postavljaju se u nulu, onemogućava se prekid i postavlja se prekidni način rada O. Sabirnice adresa i podataka postavljaju se u visoku impedanciju, a kontrolni signali u naktivno stanje. Memorija se ne osvježava.

KONTROLA SABIRNICE PROCESORA (engl. CPU BUS CONTROL)

BUSRQ - (ULAZNI) zahtjev procesoru da preda upravljanje nad sabirnicama adresa, podataka i kontrolnim priključcima. Kada je zahtjev prihvaćen, svi ti priključci prelaze u stanje visoke impedancije i vanjskoj jedinici koja je postavila zahtjev omogućuju preuzimanje upravljanja nad njima. Zahtjev za sabirnicom je prioritetniji od običnog i nemaskiranog zahtjeva za prekid. Zahtjev za sabirnicom bit će prihvaćen čim završi trenutni strojni ciklus.

BUSAQ - (izlazni) dojava vanjskoj jedinici koja je zahtijevala sabirnicu da je procesor postavio sve priključke u stanje visoke impedancije

27

Programiranje mikroprocesora Z80

Program pišemo u simboličkom strojnom jeziku, odnosno mnemoničkom obliku. To znači da se instrukcije ne prikazuju u binarnom obliku nego u obliku korijene engleske riječi koja opisuje njihovu namjenu (npr. ADD A,B za zbrajanje).

Postupak razvoja programa: 1. PISANJE : EDITOR – IME.ASM

- assembler je ime za strojni jezik ali i za program

prevoditelj napisanog ASM programa

2. PREVOĐENJE : X80 (tzv. cross assembler) – IME.OBJ

3. POVEZIVANJE : LINK1 – IME.TSH ili IME.HEX

4. IZVOĐENJE

Popis instrukcija po abecednom redu

Za svaku instrukciju naveden je pripadajući operacijski kod u obliku heksadecimalnog zapisa, koji je kod nekih jednostbitni, a kod nekih višebajtni. Kod višebajtnih je redoslijed bajtova od prvog prema višim adresama prikazan s lijeva na desno bez razmaka.

Broj perioda osnovnog signala takta potreban za izvođenje pojedine instrukcije je označen sa oznakom "Clocks".

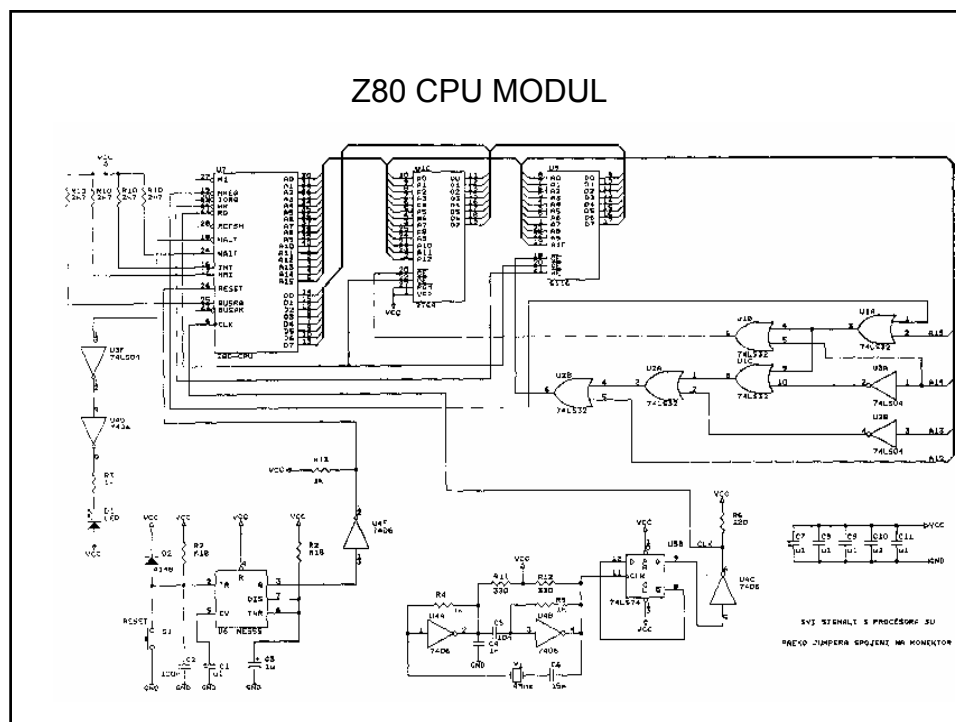
U rubrici "Comment" nalazi se simboličko objašnjenje operacije koja se izvodi pojedinom instrukcijom. Kod logičkih operacija upotrebljen je simbol "v" za logičku operaciju ILI, simbol "A" za logičku operaciju I te simbol "V" za operaciju ISKLJUČIVO ILI (Exor).

28

Stanja zastavica imaju ova značenja:

- ? - znači da nije definirano stanje
- + - znači da se zastavica mijenja izvođenjem
- 0 – postavlja stanje 0
- 1 – postavlja stanje 1
- “ ” – praznina znači da se zastavica ne mijenja
- P – označava paritet kod P/V zastavice
- V –označava preliv kod P/V zastavice

"LD A,(HL)", gdje se u HL nalazi 16-bitna adresa memorijske lokacije, čiji sadržaj se prebacuje u akumulator.



Instruction	Opcode	Clocks	Comment	Flags						
				S	Z	H	PV	N	C	
ADC A,(HL)	8E	7	$A = A + (HL) - Cy$	↑	↑	↑	V	0	↑	
ADC A,(IX+dd)	DD8Edd	19	$A = A + (IX+dd) + Cy$	↑	↑	↑	V	0	↑	
ADC A,(IY+dd)	FD8Edd	19	$A = A + (IY+dd) + Cy$	↑	↑	↑	V	0	↑	
ADC A,B	88	4	$A = A + B + Cy$	↑	↑	↑	V	0	↑	
ADC A,C	89	4	$A = A + C + Cy$	↑	↑	↑	V	0	↑	
ADC A,D	8A	4	$A = A + D + Cy$	↑	↑	↑	V	0	↑	
ADC A,E	8B	4	$A = A + E + Cy$	↑	↑	↑	V	0	↑	
ADC A,H	8C	4	$A = A + H + Cy$	↑	↑	↑	V	0	↑	
ADC A,L	8D	4	$A = A + L + Cy$	↑	↑	↑	V	0	↑	
ADC A,A	8F	4	$A = A + A + Cy$	↑	↑	↑	V	0	↑	
ADC A,d8	CEd8	7	$A = A + d8 + Cy$	↑	↑	↑	V	0	↑	
ADC HL,BC	ED4A	15	$HL = HL + BC - Cy$	↑	↑	?	V	0	↑	
ADC HL,DE	ED5A	15	$HL = HL + DE - Cy$	↑	↑	?	V	0	↑	
ADC HL,HL	ED6A	15	$HL = HL + HL - Cy$	↑	↑	?	V	0	↑	
ADC HL,SP	ED7A	15	$HL = HL + SP - Cy$	↑	↑	?	V	0	↑	
ADD A,(HL)	86	7	$A = A + (HL)$	↑	↑	↑	V	0	↑	
ADD A,(IX+dd)	DD86dd	19	$A = A + (IX+dd)$	↑	↑	↑	V	0	↑	
ADD A,(IY+dd)	FD86dd	19	$A = A + (IY+dd)$	↑	↑	↑	V	0	↑	
ADD A,B	80	4	$A = A + B$	↑	↑	↑	V	0	↑	
ADD A,C	81	4	$A = A + C$	↑	↑	↑	V	0	↑	
ADD A,D	82	4	$A = A + D$	↑	↑	↑	V	0	↑	
ADD A,E	83	4	$A = A + E$	↑	↑	↑	V	0	↑	
ADD A,H	84	4	$A = A + H$	↑	↑	↑	V	0	↑	

Instruction	Opcode	Clocks	Comment	Flags						
				S	Z	H	PV	N	C	
ADD A,L	S5	4	$A = A + L$	↑	↑	↑	V	0	↑	
ADD A,A	S7	4	$A = A + A$	↑	↑	↑	V	0	↑	
ADD A,d8	C6d8	7	$A = A + d8$	↑	↑	↑	V	0	↑	
ADD HL,BC	09	11	$HL = HL + BC$	↑	↑	?	V	0	↑	
ADD HL,DE	19	11	$HL = HL + DE$	↑	↑	?	V	0	↑	
ADD HL,HL	29	11	$HL = HL + HL$	↑	↑	?	V	0	↑	
ADD HL,SP	39	11	$HL = HL + SP$	↑	↑	?	V	0	↑	
ADD IX,BC	DD09	15	$IX = IX + BC$	↑	↑	?	V	0	↑	
ADD IX,DE	DD19	15	$IX = IX + DE$	↑	↑	?	V	0	↑	
ADD IX,IX	DD29	15	$IX = IX + IX$	↑	↑	?	V	0	↑	
ADD IX,SP	DD39	15	$IX = IX + SP$	↑	↑	?	V	0	↑	
ADD IY,BC	FD09	15	$IY = IY + BC$	↑	↑	?	V	0	↑	
ADD IY,DE	FD19	15	$IY = IY + DE$	↑	↑	?	V	0	↑	
ADD IY,IY	FD29	15	$IY = IY + IY$	↑	↑	?	V	0	↑	
ADD IY,SP	FD39	15	$IY = IY + SP$	↑	↑	?	V	0	↑	
AND (HL)	A6	7	$A = A \wedge (HL)$	↑	↑	↑	P	0	0	
AND (IX+dd)	DDA6dd	19	$A = A \wedge (IX+dd)$	↑	↑	↑	P	0	0	
AND (IY+dd)	FDA6dd	19	$A = A \wedge (IY+dd)$	↑	↑	↑	P	0	0	
AND B	A0	4	$A = A \wedge B$	↑	↑	↑	P	0	0	
AND C	A1	4	$A = A \wedge C$	↑	↑	↑	P	0	0	
AND D	A2	4	$A = A \wedge D$	↑	↑	↑	P	0	0	
AND E	A3	4	$A = A \wedge E$	↑	↑	↑	P	0	0	
AND H	A4	4	$A = A \wedge H$	↑	↑	↑	P	0	0	
AND L	A5	4	$A = A \wedge L$	↑	↑	↑	P	0	0	
AND A	A7	4	$A = A \wedge A$	↑	↑	↑	P	0	0	
AND DB	EDd8	7	$A = A \wedge d8$	↑	↑	↑	P	0	0	
BIT 0,(HL)	CB46	12	test bit 0 of (HL)	?	↑	↑	?	?	?	
BIT 0,(IX+dd)	DDCBdd46	20	test bit 0 of (IX+dd)	?	↑	↑	?	?	?	
BIT 0,(IY+dd)	FDCCdd46	20	test bit 0 of (IY+dd)	?	↑	↑	?	?	?	
BIT 0,B	CB40	8	test bit 0 of B	?	↑	↑	?	?	?	
BIT 0,C	CB41	8	test bit 0 of C	?	↑	↑	?	?	?	
BIT 0,D	CB42	8	test bit 0 of D	?	↑	↑	?	?	?	
BIT 0,E	CB43	8	test bit 0 of E	?	↑	↑	?	?	?	
BIT 0,H	CB44	8	test bit 0 of H	?	↑	↑	?	?	?	
BIT 0,L	CB45	8	test bit 0 of L	?	↑	↑	?	?	?	
BIT 0,A	CB47	8	test bit 0 of A	?	↑	↑	?	?	?	
BIT 1,(HL)	CB4E	12	test bit 1 of (HL)	?	↑	↑	?	?	?	
BIT 1,(IX+dd)	DDCBdd4E	20	test bit 1 of (IX+dd)	?	↑	↑	?	?	?	
BIT 1,(IY+dd)	FDCCdd4E	20	test bit 1 of (IY+dd)	?	↑	↑	?	?	?	
BIT 1,B	CB48	8	test bit 1 of B	?	↑	↑	?	?	?	
BIT 1,C	CB49	8	test bit 1 of C	?	↑	↑	?	?	?	
BIT 1,D	CB4A	8	test bit 1 of D	?	↑	↑	?	?	?	
BIT 1,E	CB4B	8	test bit 1 of E	?	↑	↑	?	?	?	
BIT 1,H	CB4C	8	test bit 1 of H	?	↑	↑	?	?	?	
BIT 1,L	CB4D	8	test bit 1 of L	?	↑	↑	?	?	?	

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
BIT 1,A	CB4F	8	test bit 1 of A	?	?	?	?	?	?
BIT 2,(HL)	C656	12	test bit 2 of (HL)	?	?	?	?	?	?
BIT 2,(IX+dd)	DDCBdd56	20	test bit 2 of (IX+dd)	?	?	?	?	?	?
BIT 2,(IY+dd)	FDCBdd56	20	test bit 2 of (IY+dd)	?	?	?	?	?	?
BIT 2,B	CB50	8	test bit 2 of B	?	?	?	?	?	?
BIT 2,C	CB51	8	test bit 2 of C	?	?	?	?	?	?
BIT 2,D	CB52	8	test bit 2 of D	?	?	?	?	?	?
BIT 2,E	CB53	8	test bit 2 of E	?	?	?	?	?	?
BIT 2,H	CB54	8	test bit 2 of H	?	?	?	?	?	?
BIT 2,L	CB55	8	test bit 2 of L	?	?	?	?	?	?
BIT 2,A	CB57	8	test bit 2 of A	?	?	?	?	?	?
BIT 3,(HL)	CB5E	12	test bit 3 of (HL)	?	?	?	?	?	?
BIT 3,(IX+dd)	DDCBdd5E	20	test bit 3 of (IX+dd)	?	?	?	?	?	?
BIT 3,(IY+dd)	FDCBdd5E	20	test bit 3 of (IY+dd)	?	?	?	?	?	?
BIT 3,B	CB58	8	test bit 3 of B	?	?	?	?	?	?
BIT 3,C	CB59	8	test bit 3 of C	?	?	?	?	?	?
BIT 3,D	CB5A	8	test bit 3 of D	?	?	?	?	?	?
BIT 3,E	CB5B	8	test bit 3 of E	?	?	?	?	?	?
BIT 3,H	CB5C	8	test bit 3 of H	?	?	?	?	?	?
BIT 3,L	CB5D	8	test bit 3 of L	?	?	?	?	?	?
BIT 3,A	CB5F	8	test bit 3 of A	?	?	?	?	?	?
BIT 4,(HL)	CB66	12	test bit 4 of (HL)	?	?	?	?	?	?
BIT 4,(IX+dd)	DDCBdd66	20	test bit 4 of (IX+dd)	?	?	?	?	?	?
BIT 4,(IY+dd)	FDCBdd66	20	test bit 4 of (IY+dd)	?	?	?	?	?	?
BIT 4,B	CB60	8	test bit 4 of B	?	?	?	?	?	?
BIT 4,C	CB61	8	test bit 4 of C	?	?	?	?	?	?
BIT 4,D	CB62	8	test bit 4 of D	?	?	?	?	?	?
BIT 4,E	CB63	8	test bit 4 of E	?	?	?	?	?	?
BIT 4,H	CB64	8	test bit 4 of H	?	?	?	?	?	?
BIT 4,L	CB65	8	test bit 4 of L	?	?	?	?	?	?
BIT 4,A	CB67	8	test bit 4 of A	?	?	?	?	?	?
BIT 5,(HL)	CB6E	12	test bit 5 of (HL)	?	?	?	?	?	?
BIT 5,(IX+dd)	DDCBdd6E	20	test bit 5 of (IX+dd)	?	?	?	?	?	?
BIT 5,(IY+dd)	FDCBdd6E	20	test bit 5 of (IY+dd)	?	?	?	?	?	?
BIT 5,B	CB60	8	test bit 5 of B	?	?	?	?	?	?
BIT 5,C	CB61	8	test bit 5 of C	?	?	?	?	?	?
BIT 5,D	CB62	8	test bit 5 of D	?	?	?	?	?	?
BIT 5,E	CB63	8	test bit 5 of E	?	?	?	?	?	?
BIT 5,H	CB64	8	test bit 5 of H	?	?	?	?	?	?
BIT 5,L	CB65	8	test bit 5 of L	?	?	?	?	?	?
BIT 5,A	CB67	8	test bit 5 of A	?	?	?	?	?	?
BIT 6,(HL)	CB76	12	test bit 6 of (HL)	?	?	?	?	?	?
BIT 6,(IX+dd)	DDCBdd76	20	test bit 6 of (IX+dd)	?	?	?	?	?	?
BIT 6,(IY+dd)	FDCBdd76	20	test bit 6 of (IY+dd)	?	?	?	?	?	?
BIT 6,B	CB70	8	test bit 6 of B	?	?	?	?	?	?

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
BIT 6,C	CB71	8	test bit 6 of C	?	?	?	?	?	?
BIT 6,D	CB72	8	test bit 6 of D	?	?	?	?	?	?
BIT 6,E	CB73	8	test bit 6 of E	?	?	?	?	?	?
BIT 6,H	CB74	8	test bit 6 of H	?	?	?	?	?	?
BIT 6,L	CB75	8	test bit 6 of L	?	?	?	?	?	?
BIT 6,A	CB77	8	test bit 6 of A	?	?	?	?	?	?
BIT 7,(HL)	CB7E	12	test bit 7 of (HL)	?	?	?	?	?	?
BIT 7,(IX+dd)	DDCBdd7E	20	test bit 7 of (IX+dd)	?	?	?	?	?	?
BIT 7,(IY+dd)	FDCBdd7E	20	test bit 7 of (IY+dd)	?	?	?	?	?	?
BIT 7,B	CB78	8	test bit 7 of B	?	?	?	?	?	?
BIT 7,C	CB79	8	test bit 7 of C	?	?	?	?	?	?
BIT 7,D	CB7A	8	test bit 7 of D	?	?	?	?	?	?
BIT 7,E	CB7B	8	test bit 7 of E	?	?	?	?	?	?
BIT 7,H	CB7C	8	test bit 7 of H	?	?	?	?	?	?
BIT 7,L	CB7D	8	test bit 7 of L	?	?	?	?	?	?
BIT 7,A	CB7F	8	test bit 7 of A	?	?	?	?	?	?
CALL C,adr	DCIhh	10/17	CALL if carry	?	?	?	?	?	?
CALL M,adr	FCIhh	10/17	CALL if minus	?	?	?	?	?	?
CALL NC,adr	D4Ihh	10/17	CALL if no carry	?	?	?	?	?	?
CALL adr	CDIhh	17	CALL subroutine	?	?	?	?	?	?
CALL NZ,adr	C4Ihh	10/17	CALL if not zero	?	?	?	?	?	?
CALL P,adr	F4Ihh	10/17	CALL if positive	?	?	?	?	?	?
CALL PE,adr	ECIhh	10/17	CALL if parity even	?	?	?	?	?	?
CALL PO,adr	E4Ihh	10/17	CALL if parity odd	?	?	?	?	?	?
CALL Z,adr	CCIhh	10/17	CALL if zero	?	?	?	?	?	?
CCF	3F	4	complement carry	?	?	?	?	?	?
CP (HL)	BE	7	compare A with (HL)	?	?	?	?	?	?
CP (IX+dd)	DDBEdd	19	compare A with (IX+dd)	?	?	?	?	?	?
CP (IY+dd)	FDBEd	19	compare A with (IY+dd)	?	?	?	?	?	?
CP B	BB	4	compare A with B	?	?	?	?	?	?
CP C	BB	4	compare A with C	?	?	?	?	?	?
CP D	BA	4	compare A with D	?	?	?	?	?	?
CP E	BB	4	compare A with E	?	?	?	?	?	?
CP H	BC	4	compare A with H	?	?	?	?	?	?
CP L	BD	4	compare A with L	?	?	?	?	?	?
CP A	BF	4	compare A with A	?	?	?	?	?	?
CP d8	FED8	7	compare A with d8	?	?	?	?	?	?
CPD	EDA9	16	compare A with (HL) then decrement HL and BC	?	?	?	?	?	?
CPDR	EDB9	21/16	compare A with (HL) then decrement HL and BC. Repeat until BC = 0 or A = (HL).	?	?	?	?	?	?
CPI	EDA1	16	compare A with (HL) then increment HL and decrement BC	?	?	?	?	?	?

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
CPIR	EDB1	21/16	compare A with (HL) then decrement BC and increment HL. Repeat until BC = 0 or A = (HL).	1	1	1	1	1	1
CPL	2F	4	complement A				1		1
DAA	27	4	decimal adjust A	1	1	1	P		1
DEC (HL)	35	11	(HL) = (HL) - 1	1	1	1	V	1	1
DEC (IX+dd)	DD35dd	23	(IX+dd) = (IX+dd) - 1	1	1	1	V	1	1
DEC (IY+dd)	FD35dd	23	(IY+dd) = (IY+dd) - 1	1	1	1	V	1	1
DEC B	05	4	B = B - 1	1	1	1	V	1	1
DEC C	0D	4	C = C - 1	1	1	1	V	1	1
DEC D	15	4	D = D - 1	1	1	1	V	1	1
DEC E	1D	4	E = E - 1	1	1	1	V	1	1
DEC H	25	4	H = H - 1	1	1	1	V	1	1
DEC L	2D	4	L = L - 1	1	1	1	V	1	1
DEC A	3D	4	A = A - 1	1	1	1	V	1	1
DEC BC	0B	6	BC = BC - 1						
DEC DE	1B	6	DE = DE - 1						
DEC HL	2B	6	HL = HL - 1						
DEC SP	3B	6	SP = SP - 1						
DEC IX	DD2B	10	IX = IX - 1						
DEC IY	FD2B	10	IY = IY - 1						
DI	F3	4	disable interrupts						
DJNZ	10dd	8/13	decrement B and jump if B is not zero						
EI	F8	4	enable interrupts						
EX (SP),HL	E3	19	exchange HL with (SP)						
EX (SP),IX	DOE3	23	exchange IX with (SP)						
EX (SP),IY	FDE3	23	exchange IY with (SP)						
EX AF,AF'	08	4	exchange AF with AF'						
EX DE,HL	E8	4	exchange DE with HL						
EXX	D9	4	exchange BC, DE, and HL with BC', DE', and HL'						
HALT	76	4	halt for interrupt of reset						
IM 0	ED46	8	select interrupt mode 0						
IM 1	ED56	8	select interrupt mode 1						
IM 2	ED66	8	select interrupt mode 2						
IN A,(C)	ED76	12	input data to A from I/O port (C)	1	1	1	P	0	0
IN A,(p8)	DBp8	11	input data to A from I/O port (p8)	1	1	1	P	0	0
IN B,(C)	ED40	12	input data to B from I/O port (C)	1	1	1	P	0	0
IN C,(C)	ED48	12	input data to C from I/O port (C)	1	1	1	P	0	0

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
IN D,(C)	ED50	12	input data to E from I/O port (C)	1	1	1	P	0	0
IN E,(C)	ED58	12	input data to E from I/O port (C)	1	1	1	P	0	0
IN H,(C)	ED60	12	input data to H from I/O port (C)	1	1	1	P	0	0
IN L,(C)	ED68	12	input data to L from I/O port (C)	1	1	1	P	0	0
INC (HL)	34	11	(HL) = (HL) + 1	1	1	1	V	0	0
INC (IX+dd)	DD34dd	23	(IX+dd) = (IX+dd) + 1	1	1	1	V	0	0
INC (IY+dd)	FD34dd	23	(IY+dd) = (IY+dd) + 1	1	1	1	V	0	0
INC B	04	4	B = B + 1	1	1	1	V	0	0
INC C	0C	4	C = C + 1	1	1	1	V	0	0
INC D	14	4	D = D + 1	1	1	1	V	0	0
INC E	1C	4	E = E + 1	1	1	1	V	0	0
INC H	24	4	H = H + 1	1	1	1	V	0	0
INC L	2C	4	L = L + 1	1	1	1	V	0	0
INC A	3C	4	A = A + 1	1	1	1	V	0	0
INC BC	03	6	BC = BC + 1						
INC DE	13	6	DE = DE + 1						
INC HL	23	6	HL = HL + 1						
INC SP	33	6	SP = SP + 1						
INC IX	DD23	10	IX = IX + 1						
INC IY	FD23	10	IY = IY + 1						
IND	EDA4	16	input data to (HL) from I/O port (C). Decrement B and HL. Repeat until B = 0.	?	1	?	?	1	?
INDR	EDBA	21/16	input data to (HL) from I/O port (C). Decrement B and HL. Repeat until B = 0.	?	1	?	?	1	?
INI	EDA2	16	input data to (HL) from I/O port (C). Decrement B and increment HL. Repeat until B = 0.	?	1	?	?	1	?
INIR	EDB2	21/16	input data to (HL) from I/O port (C). Decrement B and increment HL. Repeat until B = 0.	?	1	?	?	1	?
JP (HL)	E9	4	jump to address (HL)						
JP (IX)	DDE9	8	jump to address (IX)						
JP (IY)	FDE9	8	jump to address (IY)						
JP C,adr	DA11hh	10	jump if carry						
JP M,adr	FA11hh	10	jump if minus						
JP NC,adr	D211hh	10	jump if no carry						
JP adr	C311hh	10	jump						
JP NZ,adr	C211hh	10	jump if not zero						
JP P,adr	F211hh	10	jump if positive						

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
JP PE,adr	EAllhh	10	jump if parity even						
JP PO,adr	E2llhh	10	jump if parity odd						
JP Z,adr	CAllhh	10	jump if zero						
JR C,dd	38dd	7/12	jump relative if carry						
JR dd	18dd	12	jump relative						
JR NC,dd	30dd	7/12	jump relative if no carry						
JR NZ,dd	20dd	7/12	jump relative if not zero						
JR Z,dd	28dd	7/12	jump relative if zero						
LD (BC),A	02	7	load (BC) from A						
LD (DE),A	12	7	load (DE) from A						
LD (HL),B	70	7	load (HL) from B						
LD (HL),C	71	7	load (HL) from C						
LD (HL),D	72	7	load (HL) from D						
LD (HL),E	73	7	load (HL) from E						
LD (HL),H	74	7	load (HL) from H						
LD (HL),L	75	7	load (HL) from L						
LD (HL),A	77	7	load (HL) from A						
LD (HL),d8	36d8	7	load (HL) with d8						
LD (IX+dd),B	DD70dd	19	load (IX+dd) from B						
LD (IX+dd),C	DD71dd	19	load (IX+dd) from C						
LD (IX+dd),D	DD72dd	19	load (IX+dd) from D						
LD (IX+dd),E	DD73dd	19	load (IX+dd) from E						
LD (IX+dd),H	DD74dd	19	load (IX+dd) from H						
LD (IX+dd),L	DD75dd	19	load (IX+dd) from L						
LD (IX+dd),A	DD77dd	19	load (IX+dd) from A						
LD (IX+dd),d8	DD36ddd8	19	load (IX+dd) with d8						
LD (IY+dd),B	FD70dd	19	load (IY+dd) from B						
LD (IY+dd),C	FD71dd	19	load (IY+dd) from C						
LD (IY+dd),D	FD72dd	19	load (IY+dd) from D						
LD (IY+dd),E	FD73dd	19	load (IY+dd) from E						
LD (IY+dd),H	FD74dd	19	load (IY+dd) from H						
LD (IY+dd),A	FD77dd	19	load (IY+dd) from A						
LD (IY+dd),d8	FD36ddd8	19	load (IY+dd) with d8						
LD (adr),A	32llhh	13	load (adr) from A						
LD (adr),BC	ED43llhh	20	load (adr) from BC						
LD (adr),DE	ED52llhh	20	load (adr) from DE						
LD (adr),HL	22llhh	16	load (adr) from HL						
LD (adr),IX	DD22llhh	20	load (adr) from IX						
LD (adr),IY	FD22llhh	20	load (adr) from IY						
LD (adr),SP	ED73llhh	20	load (adr) from SP						
LD A,(BC)	0A	7	load A from (BC)						
LD A,(DE)	1A	7	load A from (DE)						
LD A,(HL)	7E	7	load A from (HL)						
LD A,(IX+dd)	DD7Edd	19	load A from (IX+dd)						
LD A,(IY+dd)	FD7Edd	19	load A from (IY+dd)						

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
LD A,(adr)	3A1llhh	13	load A from (adr)						
LD A,A	7F	4	load A from A						
LD A,B	7B	4	load A from B						
LD A,C	79	4	load A from C						
LD A,D	7A	4	load A from D						
LD A,E	78	4	load A from E						
LD A,H	7C	4	load A from H						
LD A,L	7D	4	load A from L						
LD A,d8	35d8	7	load A with d8						
LD A,I	ED57	9	load A from I						
LD A,R	ED5F	9	load A from R						
LD B,(HL)	46	7	load B from (HL)						
LD B,(IX+dd)	DD46nn	19	load B from (IX+dd)						
LD B,(IY+dd)	FD46dd	19	load B from (IY+dd)						
LD B,A	47	4	load B from A						
LD B,C	48	4	load B from C						
LD B,D	49	4	load B from D						
LD B,E	4A	4	load B from E						
LD B,H	43	4	load B from H						
LD B,L	4C	4	load B from L						
LD B,d8	05d8	7	load B with d8						
LD BC,(adr)	ED4Bllhh	20	load BC from (adr)						
LD BC,d16	01llhh	10	load BC with d16						
LD C,(HL)	4E	7	load C from (HL)						
LD C,(IX+dd)	DD4Edd	19	load C from (IX+dd)						
LD C,(IY+dd)	FD4Edd	19	load C from (IY+dd)						
LD C,A	4F	4	load C from A						
LD C,B	48	4	load C from B						
LD C,C	49	4	load C from C						
LD C,D	4A	4	load C from D						
LD C,E	4B	4	load C from E						
LD C,H	4C	4	load C from H						
LD C,L	4D	4	load C from L						
LD C,d8	0E d8	7	load C with d8						
LD D,(HL)	56	7	load D from (HL)						
LD D,(IX+dd)	DD56dd	19	load D from (IX+dd)						
LD D,(IY+dd)	FD56dd	19	load D from (IY+dd)						
LD D,A	57	7	load D from A						
LD D,B	50	4	load D from B						
LD D,C	51	4	load D from C						
LD D,D	52	4	load D from D						
LD D,E	53	4	load D from E						
LD D,H	54	4	load D from H						
LD D,L	55	4	load D from L						
LD D,d8	16d8	7	load D with d8						

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
LD DE,(adr)	ED5Bllhh	20	load DE from (adr)						
LD DE,d16	11llhh	10	load DE with d16						
LD E,(HL)	5E	7	load E from (HL)						
LD E,(IX + dd)	DD5Edd	19	load E from (IX + dd)						
LD E,(IY + dd)	FD5Edd	19	load E from (IY + dd)						
LD E,A	5F	4	load E from A						
LD E,B	58	4	load E from B						
LD E,C	59	4	load E from C						
LD E,D	5A	4	load E from D						
LD E,E	5B	4	load E from E						
LD E,H	5C	4	load E from H						
LD E,L	5D	4	load E from L						
LD E,d8	1Ed8	7	load E with d8						
LD H,(HL)	66	7	load H from (HL)						
LD H,(IX + dd)	DD66dd	19	load H from (IX + dd)						
LD H,(IY + dd)	FD66dd	19	load H from (IY + dd)						
LD H,A	67	4	load H from A						
LD H,B	60	4	load H from B						
LD H,C	61	4	load H from C						
LD H,D	62	4	load H from D						
LD H,E	63	4	load H from E						
LD H,H	64	4	load H from H						
LD H,L	65	4	load H from L						
LD H,d8	26d8	4	load H with d8						
LD HL,(adr)	2Allhh	16	load HL from (HL)						
LD HL,d16	21llhh	10	load HL with d16						
LD I,A	ED47	9	load I from A						
LD IX,(adr)	DD2Allhh	20	load IX from (adr)						
LD IX,d16	DD21llhh	14	load IX with d16						
LD IY,(adr)	FD2Allhh	20	load IY from (adr)						
LD IY,d16	FD21llhh	14	load IY with d16						
LD L,(HL)	CE	7	load L from (HL)						
LD L,(IX + dd)	DD6Edd	19	load L from (IX + dd)						
LD L,(IY + dd)	FD6Edd	19	load L from (IY + dd)						
LD L,A	6F	4	load L from A						
LD L,B	68	4	load L from B						
LD L,C	69	4	load L from C						
LD L,D	6A	4	load L from D						
LD L,E	6B	4	load L from E						
LD L,H	6C	4	load L from H						
LD L,L	6D	4	load L from L						
LD L,d8	2Ed8	4	load L with d8						
LD R,A	ED4F	4	load R from A						
LD SP,(adr)	ED7Bllhh	20	load SP from (adr)						
LD SP,HL	F9	6	load SP from HL						

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
LD SP,IX	DDF9	10	load SP from IX						
LD SP,IY	DDF9	10	load SP from IY						
LD SP,d16	21llhh	10	load SP with d16						
LDD	EDA3	16	load (HL) from (DE) then decrement BC, DE, and HL.	0	1	0			
LDDR	EDB8	21/16	load (HL) from (DE) then decrement BC, DE, and HL. Repeat until BC = 0.	0	0	0			
LDI	EDA3	16	load (HL) from (DE) then decrement BC and increment DE and HL.	0	1	0			
LDIR	EDB0	21/16	load (HL) from (DE) then decrement BC and increment DE and HL. Repeat until BC = 0.	0	1	0			
NEG	ED44	8	2's complement A	1	1	1	V	1	1
OR (HL)	B6	7	A = A V (HL)	1	1	0	P	0	0
OR (IX + dd)	DDB6dd	19	A = A V (IX + dd)	1	1	0	P	0	0
OR (IY + dd)	FDB6dd	19	A = A V (IY + dd)	1	1	0	P	0	0
OR A	B7	4	A = A V A	1	1	0	P	0	0
OR B	B0	4	A = A V B	1	1	0	P	0	0
OR C	B1	4	A = A V C	1	1	0	P	0	0
OR D	B2	4	A = A V D	1	1	0	P	0	0
OR E	B3	4	A = A V E	1	1	0	P	0	0
OR H	B4	4	A = A V H	1	1	0	P	0	0
OR L	B5	4	A = A V L	1	1	0	P	0	0
OR d8	F6d8	7	A = A V d8	1	1	0	P	0	0
OTDR	E0B8	21/16	output (HL) to I/O port (C). Decrement B and HL. Repeat until B = 0.						
OTIR	EDB3	21/16	output (HL) to I/O port (C). Decrement B and increment HL. Repeat until B = 0.						
OUT (C),A	ED75	12	output A to I/O port (C).						
OUT (C),B	ED41	12	output B to I/O port (C).						
OUT (C),C	ED49	12	output C to I/O port (C).						
OUT (C),D	ED51	12	output D to I/O port (C).						
OUT (C),E	ED59	12	output E to I/O port (C).						
OUT (C),H	ED61	12	output H to I/O port (C).						
OUT (C),L	ED69	12	output L to I/O port (C).						
OUT (p8), A	D3p8	11	output A to I/O port (p8)						
OUTD	EDA8	16	output (HL) to I/O port (C) then decrement B and HL.	?	1	?	?	1	?
OUTI	EDA3	16	output (HL) to I/O	?	1	?	?	1	?
POP AF	F1	10	load AF from stack	1	1	1	1	1	1
POP BC	C1	10	load BC from stack						

Instruction	Opcode	Clocks	Comment	Flags				
				S	Z	H	P/V	N C
POP DE	D1	10	load DE from stack					
POP HL	E1	10	load HL from stack					
POP IX	DDE1	14	load IX from stack					
POP IY	FDE1	14	load IY from stack					
PUSH AF	F5	11	load stack from AF					
PUSH BC	C5	11	load stack from BC					
PUSH DE	D5	11	load stack from DE					
PUSH HL	E5	11	load stack from HL					
PUSH IX	DDE5	15	load stack from IX					
PUSH IY	FDE5	15	load stack from IY					
RES 0,(HL)	CB86	15	bit 0 of (HL) = 0					
RES 0,(IX + dd)	DDCBdd86	23	bit 0 of (IX + dd) = 0					
RES 0,(IY + dd)	FDCBdd86	23	bit 0 of (IY + dd) = 0					
RES 0,A	CB87	8	bit 0 of A = 0					
RES 0,B	CB80	8	bit 0 of B = 0					
RES 0,C	CB81	8	bit 0 of C = 0					
RES 0,D	CB82	8	bit 0 of D = 0					
RES 0,E	CB83	8	bit 0 of E = 0					
RES 0,H	CB84	8	bit 0 of H = 0					
RES 0,L	CB85	8	bit 0 of L = 0					
RES 1,(HL)	CB8E	15	bit 1 of (HL) = 0					
RES 1,(IX + dd)	DDCBdd8E	23	bit 1 of (IX + dd) = 0					
RES 1,(IY + dd)	FDCBdd8E	23	bit 1 of (IY + dd) = 0					
RES 1,A	CB8F	8	bit 1 of A = 0					
RES 1,B	CB88	8	bit 1 of B = 0					
RES 1,C	CB89	8	bit 1 of C = 0					
RES 1,D	CB8A	8	bit 1 of D = 0					
RES 1,E	CB8B	8	bit 1 of E = 0					
RES 1,H	CB8C	8	bit 1 of H = 0					
RES 1,L	CB8D	8	bit 1 of L = 0					
RES 2,(HL)	CB96	15	bit 2 of (HL) = 0					
RES 2,(IX + dd)	DDCBdd96	23	bit 2 of (IX + dd) = 0					
RES 2,(IY + dd)	FDCBdd96	23	bit 2 of (IY + dd) = 0					
RES 2,A	CB97	8	bit 2 of A = 0					
RES 2,B	CB90	8	bit 2 of B = 0					
RES 2,C	CB91	8	bit 2 of C = 0					
RES 2,D	CB92	8	bit 2 of D = 0					
RES 2,E	CB93	8	bit 2 of E = 0					
RES 2,H	CB94	8	bit 2 of H = 0					
RES 2,L	CB95	8	bit 2 of L = 0					
RES 3,(HL)	CB9E	15	bit 3 of (HL) = 0					
RES 3,(IX + dd)	DDCBdd9E	23	bit 3 of (IX + dd) = 0					
RES 3,(IY + dd)	FDCBdd9E	23	bit 3 of (IY + dd) = 0					
RES 3,A	CB9F	8	bit 3 of A = 0					
RES 3,B	CB98	8	bit 3 of B = 0					

Instruction	Opcode	Clocks	Comment	Flags				
				S	Z	H	P/V	N C
RES 3,C	CB95	8	bit 3 of C = 0					
RES 3,D	CB9A	8	bit 3 of D = 0					
RES 3,E	CB9E	8	bit 3 of E = 0					
RES 3,H	CB9C	8	bit 3 of H = 0					
RES 3,L	CB9D	8	bit 3 of L = 0					
RES 4,(HL)	CBA5	15	bit 4 of (HL) = 0					
RES 4,(IX + dd)	DDCBddA5	23	bit 4 of (IX + dd) = 0					
RES 4,(IY + dd)	FDCBddA5	23	bit 4 of (IY + dd) = 0					
RES 4,A	CBA7	8	bit 4 of A = 0					
RES 4,B	CBA0	8	bit 4 of B = 0					
RES 4,C	CBA1	8	bit 4 of C = 0					
RES 4,D	CBA2	8	bit 4 of D = 0					
RES 4,E	CBA3	8	bit 4 of E = 0					
RES 4,H	CBA4	8	bit 4 of H = 0					
RES 4,L	CBA5	8	bit 4 of L = 0					
RES 5,(HL)	CBA5	15	bit 5 of (HL) = 0					
RES 5,(IX + dd)	DDCBddA5	23	bit 5 of (IX + dd) = 0					
RES 5,(IY + dd)	FDCBddA5	23	bit 5 of (IY + dd) = 0					
RES 5,A	CBAF	8	bit 5 of A = 0					
RES 5,B	CBA8	8	bit 5 of B = 0					
RES 5,C	CBA9	8	bit 5 of C = 0					
RES 5,D	CBAA	8	bit 5 of D = 0					
RES 5,E	CBAB	8	bit 5 of E = 0					
RES 5,H	CBAC	8	bit 5 of H = 0					
RES 5,L	CBAD	8	bit 5 of L = 0					
RES 6,(HL)	CB8E	15	bit 6 of (HL) = 0					
RES 6,(IX + dd)	DDCBdd8E	23	bit 6 of (IX + dd) = 0					
RES 6,(IY + dd)	FDCBdd8E	23	bit 6 of (IY + dd) = 0					
RES 6,A	CB87	8	bit 6 of A = 0					
RES 6,B	CB80	8	bit 6 of B = 0					
RES 6,C	CB81	8	bit 6 of C = 0					
RES 6,D	CB82	8	bit 6 of D = 0					
RES 6,E	CB83	8	bit 6 of E = 0					
RES 6,H	CB84	8	bit 6 of H = 0					
RES 6,L	CB85	8	bit 6 of L = 0					
RES 7,(HL)	CB8E	15	bit 7 of (HL) = 0					
RES 7,(IX + dd)	DDCBdd8E	23	bit 7 of (IX + dd) = 0					
RES 7,(IY + dd)	FDCBdd8E	23	bit 7 of (IY + dd) = 0					
RES 7,A	CB8F	8	bit 7 of A = 0					
RES 7,B	CB88	8	bit 7 of B = 0					
RES 7,C	CB89	8	bit 7 of C = 0					
RES 7,D	CB8A	8	bit 7 of D = 0					
RES 7,E	CB8B	8	bit 7 of E = 0					
RES 7,H	CB8C	8	bit 7 of H = 0					
RES 7,L	CB8D	8	bit 7 of L = 0					

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
RET	C9	10	return						
RET C	D8	5/11	return if carry						
RET M	F8	5/11	return if minus						
RET NC	D0	5/11	return if no carry						
RET NZ	C0	5/11	return if not zero						
RET P	F0	5/11	return if positive						
RET PE	E8	5/11	return if parity even						
RET PO	E0	5/11	return if parity odd						
RET Z	C8	5/11	return if zero						
RETI	ED4D	14	return from interrupt						
RETN	ED45	14	return from nonmaskable interrupt						
RL (HL)	CB16	15	rotate (HL) left through carry	1	1	0	P	0	1
RL (IX + dd)	DDCBdd16	23	rotate (IX + dd) left through carry	1	1	0	P	0	1
RL (IY + dd)	FDCBdd16	23	rotate (IY + dd) left through carry	1	1	0	P	0	1
RL A	CB17	8	rotate A left through carry	1	1	0	P	0	1
RL B	CB10	8	rotate B left through carry	1	1	0	P	0	1
RL C	CB11	8	rotate C left through carry	1	1	0	P	0	1
RL D	CB12	8	rotate D left through carry	1	1	0	P	0	1
RL E	CB13	8	rotate E left through carry	1	1	0	P	0	1
RL H	CB14	8	rotate H left through carry	1	1	0	P	0	1
RL L	CB15	8	rotate L left through carry	1	1	0	P	0	1
RLA	17	4	rotate A left through carry						
RLC (HL)	CB06	15	rotate (HL) left	1	1	0	P	0	1
RLC (IX + dd)	DDCBdd06	23	rotate (IX + dd) left	1	1	0	P	0	1
RLC (IY + dd)	FDCBdd06	23	rotate (IY + dd) left	1	1	0	P	0	1
RLC A	CB07	8	rotate A left	1	1	0	P	0	1
RLC B	CB00	8	rotate B left	1	1	0	P	0	1
RLC C	CB01	8	rotate C left	1	1	0	P	0	1
RLC D	CB02	8	rotate D left	1	1	0	P	0	1
RLC E	CB03	8	rotate E left	1	1	0	P	0	1
RLC H	CB04	8	rotate H left	1	1	0	P	0	1
RLC L	CB05	8	rotate L left	1	1	0	P	0	1
RLCA	07	4	rotate A left						
RLD	ED6F	8	rotate left digit	1	1	0	P	0	1
RR (HL)	CB1E	15	rotate (HL) right through carry	1	1	0	P	0	1
RR (IX + dd)	DDCBdd1E	23	rotate (IX + dd) right through carry	1	1	0	P	0	1
RR (IY + dd)	FDCBdd1E	23	rotate (IY + dd) right through carry	1	1	0	P	0	1
RR A	CB1F	8	rotate A right through carry	1	1	0	P	0	1
RR B	CB18	8	rotate B right through carry	1	1	0	P	0	1
RR C	CB19	8	rotate C right through carry	1	1	0	P	0	1

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
RR D	CB1A	8	rotate D right through carry	1	1	0	P	0	1
RR E	CB1C	8	rotate E right through carry	1	1	0	P	0	1
RR H	CB1D	8	rotate H right through carry	1	1	0	P	0	1
RR L	CB10	8	rotate L right through carry	1	1	0	P	0	1
RR A	1F	4	rotate A right through carry						
RRC (HL)	CB0E	15	rotate (HL) right	1	1	0	P	0	1
RRC (IX + dd)	DDCBdd0E	23	rotate (IX + dd) right	1	1	0	P	0	1
RRC (IY + dd)	FDCBdd0E	23	rotate (IY + dd) right	1	1	0	P	0	1
RRC A	CB0F	8	rotate A right	1	1	0	P	0	1
RRC B	CB08	8	rotate B right	1	1	0	P	0	1
RRC C	CB09	8	rotate C right	1	1	0	P	0	1
RRC D	CB0A	8	rotate D right	1	1	0	P	0	1
RRC E	CB0B	8	rotate E right	1	1	0	P	0	1
RRC H	CB0C	8	rotate H right	1	1	0	P	0	1
RRC L	CB0D	8	rotate L right	1	1	0	P	0	1
RRCA	0F	4	rotate A right						
RRO	ED57	18	rotate digit right	1	1	0	P	0	1
RST 00H	C7	11	CALL 0000H						
RST 08H	CF	11	CALL 0008H						
RST 10H	D7	11	CALL 0010H						
RST 18H	DF	11	CALL 0018H						
RST 20H	E7	11	CALL 0020H						
RST 28H	EF	11	CALL 0028H						
RST 30H	F7	11	CALL 0030H						
RST 38H	FF	11	CALL 0038H						
SBC A,(HL)	9E	7	A = A - (HL) - Cy	1	1	1	V	1	1
SBC A,(IX + dd)	DD9Edd	19	A = A - (IX + dd) - Cy	1	1	1	V	1	1
SBC A,(IY + dd)	FD9Edd	19	A = A - (IY + dd) - Cy	1	1	1	V	1	1
SBC A,A	9F	4	A = A - A - Cy	1	1	1	V	1	1
SBC A,B	98	4	A = A - B - Cy	1	1	1	V	1	1
SBC A,C	99	4	A = A - C - Cy	1	1	1	V	1	1
SBC A,D	9A	4	A = A - D - Cy	1	1	1	V	1	1
SBC A,E	9B	4	A = A - E - Cy	1	1	1	V	1	1
SBC A,H	9C	4	A = A - H - Cy	1	1	1	V	1	1
SBC A,L	9D	4	A = A - L - Cy	1	1	1	V	1	1
SBC A,d8	DEd8	7	A = A - d8 - Cy	1	1	1	V	1	1
SBC HL,BC	E042	15	HL = HL - BC - Cy	1	1	1	V	1	1
SBC HL,DE	E052	15	HL = HL - DE - Cy	1	1	1	V	1	1
SBC HL,HL	E062	15	HL = HL - HL - Cy	1	1	1	V	1	1
SBC HL,SP	E072	15	HL = HL - SP - Cy	1	1	1	V	1	1
SCF	37	4	set carry				0		1
SET 0,(HL)	CB06	15	bit 0 of (HL) = 1						
SET 0,(IX + dd)	DDCBdd06	23	bit 0 of (IX + dd) = 1						
SET 0,(IY + dd)	FDCBdd06	23	bit 0 of (IY + dd) = 1						
SET 0,A	CB07	8	bit 0 of A = 1						

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
SET 0,B	CBC0	8	bit 0 of B = 1						
SET 0,C	CBC1	8	bit 0 of C = 1						
SET 0,D	CBC2	8	bit 0 of D = 1						
SET 0,E	CBC3	8	bit 0 of E = 1						
SET 0,H	CBC4	8	bit 0 of H = 1						
SET 0,L	CBC5	8	bit 0 of L = 1						
SET 1,(HL)	CBCE	15	bit 1 of (HL) = 1						
SET 1,(IX+dd)	DDCBddCE	23	bit 1 of (IX+dd) = 1						
SET 1,(IY+dd)	FDCBddCE	23	bit 1 of (IY+dd) = 1						
SET 1,A	CBCE	8	bit 1 of A = 1						
SET 1,B	CBC8	8	bit 1 of B = 1						
SET 1,C	CBC9	8	bit 1 of C = 1						
SET 1,D	CBCA	8	bit 1 of D = 1						
SET 1,E	CBCB	8	bit 1 of E = 1						
SET 1,H	CBCC	8	bit 1 of H = 1						
SET 1,L	CBCD	8	bit 1 of L = 1						
SET 2,(HL)	CBDE	15	bit 2 of (HL) = 1						
SET 2,(IX+dd)	DDCBddDE	23	bit 2 of (IX+dd) = 1						
SET 2,(IY+dd)	FDCBddDE	23	bit 2 of (IY+dd) = 1						
SET 2,A	CBDE	8	bit 2 of A = 1						
SET 2,B	CBDE	8	bit 2 of B = 1						
SET 2,C	CBDE	8	bit 2 of C = 1						
SET 2,D	CBDE	8	bit 2 of D = 1						
SET 2,E	CBDE	8	bit 2 of E = 1						
SET 2,H	CBDE	8	bit 2 of H = 1						
SET 2,L	CBDE	8	bit 2 of L = 1						
SET 3,(HL)	CBDE	15	bit 3 of (HL) = 1						
SET 3,(IX+dd)	DDCBddDE	23	bit 3 of (IX+dd) = 1						
SET 3,(IY+dd)	FDCBddDE	23	bit 3 of (IY+dd) = 1						
SET 3,A	CBDE	8	bit 3 of A = 1						
SET 3,B	CBDE	8	bit 3 of B = 1						
SET 3,C	CBDE	8	bit 3 of C = 1						
SET 3,D	CBDE	8	bit 3 of D = 1						
SET 3,E	CBDE	8	bit 3 of E = 1						
SET 3,H	CBDE	8	bit 3 of H = 1						
SET 3,L	CBDE	8	bit 3 of L = 1						
SET 4,(HL)	CBDE	15	bit 4 of (HL) = 1						
SET 4,(IX+dd)	DDCBddDE	23	bit 4 of (IX+dd) = 1						
SET 4,(IY+dd)	FDCBddDE	23	bit 4 of (IY+dd) = 1						
SET 4,A	CBDE	8	bit 4 of A = 1						
SET 4,B	CBDE	8	bit 4 of B = 1						
SET 4,C	CBDE	8	bit 4 of C = 1						
SET 4,D	CBDE	8	bit 4 of D = 1						
SET 4,E	CBDE	8	bit 4 of E = 1						
SET 4,H	CBDE	8	bit 4 of H = 1						

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
SET 4,L	CBDE	8	bit 4 of L = 1						
SET 5,(HL)	CBDE	15	bit 5 of (HL) = 1						
SET 5,(IX+dd)	DDCBddDE	23	bit 5 of (IX+dd) = 1						
SET 5,(IY+dd)	FDCBddDE	23	bit 5 of (IY+dd) = 1						
SET 5,A	CBDE	8	bit 5 of A = 1						
SET 5,B	CBDE	8	bit 5 of B = 1						
SET 5,C	CBDE	8	bit 5 of C = 1						
SET 5,D	CBDE	8	bit 5 of D = 1						
SET 5,E	CBDE	8	bit 5 of E = 1						
SET 5,H	CBDE	8	bit 5 of H = 1						
SET 5,L	CBDE	8	bit 5 of L = 1						
SET 6,(HL)	CBDE	15	bit 6 of (HL) = 1						
SET 6,(IX+dd)	DDCBddDE	23	bit 6 of (IX+dd) = 1						
SET 6,(IY+dd)	FDCBddDE	23	bit 6 of (IY+dd) = 1						
SET 6,A	CBDE	8	bit 6 of A = 1						
SET 6,B	CBDE	8	bit 6 of B = 1						
SET 6,C	CBDE	8	bit 6 of C = 1						
SET 6,D	CBDE	8	bit 6 of D = 1						
SET 6,E	CBDE	8	bit 6 of E = 1						
SET 6,H	CBDE	8	bit 6 of H = 1						
SET 6,L	CBDE	8	bit 6 of L = 1						
SET 7,(HL)	CBDE	15	bit 7 of (HL) = 1						
SET 7,(IX+dd)	DDCBddDE	23	bit 7 of (IX+dd) = 1						
SET 7,(IY+dd)	FDCBddDE	23	bit 7 of (IY+dd) = 1						
SET 7,A	CBDE	8	bit 7 of A = 1						
SET 7,B	CBDE	8	bit 7 of B = 1						
SET 7,C	CBDE	8	bit 7 of C = 1						
SET 7,D	CBDE	8	bit 7 of D = 1						
SET 7,E	CBDE	8	bit 7 of E = 1						
SET 7,H	CBDE	8	bit 7 of H = 1						
SET 7,L	CBDE	8	bit 7 of L = 1						
SLA (HL)	CB26	15	shift (HL) left	1	1	0	P	0	1
SLA (IX+dd)	DDCBdd26	23	shift (IX+dd) left	1	1	0	P	0	1
SLA (IY+dd)	FDCBdd26	23	shift (IY+dd) left	1	1	0	P	0	1
SLA A	CB26	8	shift A left	1	1	0	P	0	1
SLA B	CB20	8	shift B left	1	1	0	P	0	1
SLA C	CB21	8	shift C left	1	1	0	P	0	1
SLA D	CB22	8	shift D left	1	1	0	P	0	1
SLA E	CB23	8	shift E left	1	1	0	P	0	1
SLA H	CB24	8	shift H left	1	1	0	P	0	1
SLA L	CB25	8	shift L left	1	1	0	P	0	1
SRA (HL)	CB2E	15	arithmetic shift (HL) right	1	1	0	P	0	1
SRA (IX+dd)	DDCBdd2E	23	arithmetic shift (IX+dd) right	1	1	0	P	0	1
SRA (IY+dd)	FDCBdd2E	23	arithmetic shift (IY+dd) right	1	1	0	P	0	1
SRA A	CB2F	8	arithmetic shift A right	1	1	0	P	0	1

Instruction	Opcode	Clocks	Comment	Flags					
				S	Z	H	P/V	N	C
SRA B	CB28	8	arithmetic shift B right	1	1	0	P	0	1
SRA C	CB29	8	arithmetic shift C right	1	1	0	P	0	1
SRA D	CB2A	8	arithmetic shift D right	1	1	0	P	0	1
SRA E	CB2B	8	arithmetic shift E right	1	1	0	P	0	1
SRA H	CB2C	8	arithmetic shift H right	1	1	0	P	0	1
SRA L	CB2D	8	arithmetic shift	1	1	0	P	0	1
SRL (HL)	CB3E	15	shift (HL) right	1	1	0	P	0	1
SLR (IX+dd)	DDCBdd3E	23	shift (IX+dd) right	1	1	0	P	0	1
SRL (IY+dd)	FDCBdd3E	23	shift (IY+dd) right	1	1	0	P	0	1
SRL A	CB3F	6	shift A right	1	1	0	P	0	1
SRL B	CB38	8	shift B right	1	1	0	P	0	1
SRL C	CB39	8	shift C right	1	1	0	P	0	1
SRL D	CB3A	8	shift D right	1	1	0	P	0	1
SRL E	CB3B	8	shift E right	1	1	0	P	0	1
SRL H	CB3C	8	shift H right	1	1	0	P	0	1
SRL L	CB3D	8	shift L right	1	1	0	P	0	1
SUB (HL)	96	7	A = A - (HL)	1	1	1	V	1	1
SUB (IX+dd)	DD96dd	19	A = A - (IX+dd)	1	1	1	V	1	1
SUB (IY+dd)	FD96dd	19	A = A - (IY+dd)	1	1	1	V	1	1
SUB A	97	4	A = A - A	0	1	0	1	1	0
SUB B	90	4	A = A - B	1	1	1	V	1	1
SUB C	91	4	A = A - C	1	1	1	V	1	1
SUB D	92	4	A = A - D	1	1	1	V	1	1
SUB E	93	4	A = A - E	1	1	1	V	1	1
SUB H	94	4	A = A - H	1	1	1	V	1	1
SUB L	95	4	A = A - L	1	1	1	V	1	1
SUB d8	D6d8	7	A = A - d8	1	1	1	V	1	1
XOR (HL)	AE	7	A = A V (HL)	1	1	0	P	0	0
XOR (IX+dd)	DDAEdd	19	A = A V (IX+dd)	1	1	0	P	0	0
XOR (IY+dd)	FDAddd	19	A = A V (IY+dd)	1	1	0	P	0	0
XOR A	AF	4	A = A V A	0	1	0	1	0	0
XOR B	A8	4	A = A V B	1	1	0	P	0	0
XOR C	A9	4	A = A V C	1	1	0	P	0	0
XOR D	AA	4	A = A V D	1	1	0	P	0	0
XOR E	AB	4	A = A V E	1	1	0	P	0	0
XOR H	AC	4	A = A V H	1	1	0	P	0	0
XOR L	AD	4	A = A V L	1	1	0	P	0	0
XOR d8	EEd8	7	A = A V d8	1	1	0	P	0	0

Primjeri instrukcija

- LD A,B - iz B registra u A (akumulator)
- LD (BC),A - na adresu u memoriji koja je navedena u BC paru registara smjesti podatak iz akumulatora A
- PUSH BC - stavi na stog
- POP BC - skini sa stoga
- EX AF,AF' - razmjena podataka
- EXX - sva tri izmjenjivanja odjednom
- LDI i LDIR - spremi u (HL) sadržaj iz (DE) tada dekrementiraj BC i inkrementiraj DE i HL
 - prije upisivanja LDI treba definirati – adresu izvora podataka u HL
 - adresu odredišta podataka u DE
 - duljinu bloka (broj prenesenih bajta u BC)
- CPI i CPIR - usporedi A sa (HL), inkrementiraj HL i dekrementiraj BC
- ADD A,H - zbroji A sa H
- SUB A,B - odzmi B od A
- DAA - pretvara A u BCD oblik
- NOP - no operation (troši jedan ciklus takta)
- HALT - zaustavlja CPU
- RLC E - rotiranje u lijevo
- RRC D - rotiranje u desno
- BIT 2,H - ispitaj 2 bit registra H
- SET 2,H - postavi 2 bit registra H

RES 4,0 - resetiraj 4 bit na vrijednost 0
 IN A,ime - učitavanje s vanjske jedinice definirane adrese pod
 imenom ime (isključivo preko akumulatora)
 OUT ime,A - ispis na vanjsku jedinicu (isključivo preko akumulatora)
 JP nn - apsolutni bezuvjetni skok
 JP cc,nn - uvjetni skok
 JR c - relativni skok
 DJNZ e - skok u petlji (Decrement and Jump if Not Zero)

Primjeri programa

; Program za 16-bitno oduzimanje ako je prvi operand na adresi 1000H, drugi na 1002H,
 ; a rezultat spremiti na 1004H.

```
ORG 0000H      ; početak programa koji odgovara adresi u EPROM-u
AND A          ; zbog brisanja zastavice Cy
LD HL,1000H    ; u HL stavi adresu prvog operanda
LD BC,1002H    ; u BC stavi adresu drugog operanda
SBC (HL),(BC)
LD (1004H),(HL)
END
```

49

; Primjer za skok u petlji.

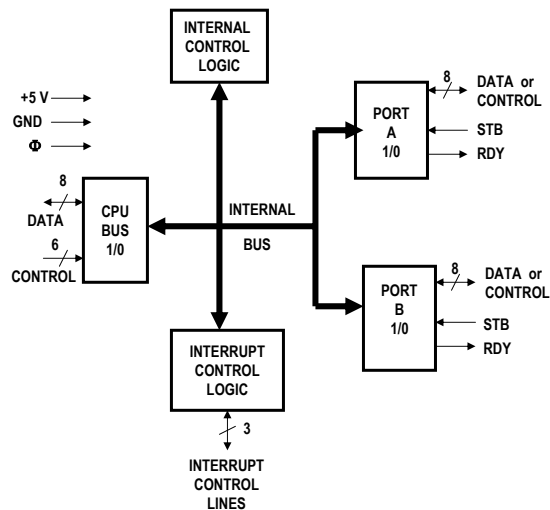
```
ORG 0000H
LD B,6
LOOP: ADD HL,HL
      DJNZ LOOP
      JP ENDP
      NOP
END P: HALT
      END
```

; Broj u registru D pomnožiti sa 7.

```
ORG 0000H
LD A,D
SLA A          ; *2
SLA A          ; *2
ADD A,D        ; +1
ADD A,D        ; +1
ADD A,D        ; +1
LD D,A
END
```

50

Ulazno-izlazni PIO sklop



51

Načini rada

Ulazi-izlazi (portovi) unutar sklopa PIO mogu raditi u jednom od četiri osnovna načina:

- način 0 – Izlazni način
- način 1 – Ulazni način
- način 2 – Dvosmjerni način (u ovom način može raditi samo Port A)
- način 3 – Kontrola bitova

Način 0: Izlazni način rada

U ovom načinu rada izabrani port služiti će za izlaz podataka. Osam-bitni podatak upisan u registar Podataka pojaviti će se na izlazu PIO sklopa i aktiviranjem linije RDY javiti vanjskoj jedinici da su Podatci za nju spremni. Nakon što vanjska jedinica pročita podatak, obavještava o tome PIO aktiviranjem linije STB. Pošto PIO prepozna taj povratni signal, generirat će prekid, čime se procesor obavještava da može ispisati (poslati) sljedeći podatak.

Način 1: Ulazni način rada

U ovom načinu rada izabrani port služiti će za ulaz podataka. Nakon što vanjska jedinica postavi Podatak na ulaz aktivirat će ulaznu liniju STB, čime upisuje taj podatak u registar podataka PIO sklopa. PIO sklop odmah nakon prepoznavanja tog signala generira prekid, čime obavještava Z80 da ima podatak za njega. Tek pošto Z80 pročita podatak iz registra podataka, na PIO portu aktivira se izlazna linija RDY, čime se javlja vanjskoj jedinici da može poslati sljedeći podatak.

52

Način 2. Dvosmjerni način

Taj način rada kombinacija je načina 0 i načina 1 i port može služiti za izlaz i za ulaz podataka. Kako kontrola takvog načina rada zahtjeva 4 kontrolne linije, ovaj način rada može se izvesti samo na portu A.

Način 3. Kontrola bitova

Ovim se načinom rada ne upotrebljavaju linije za upravljanje, pa je učitavanje ili ispisivanje podataka moguće u svakom trenutku. Osim toga, ovim načinom rada može se za svaki pojedini bit definirati je li ulazni ili izlazni, po čemu je taj način dobio ime. Dodatno se može definirati uz koje će uvjete PIO generirati prekid, što taj način rada čini izuzetno praktičnim za upotrebu.

U primjerima će se upotrebljavati isključivo način 3, jer pri njegovj upotrebi nema nikakvih zahtjeva na vanjski sklop (nema signala upravljanja) i ima praktičan način generiranja prekida.

Programiranje

ADRESA_PIO = PODACI A

ADRESA_PIO+1 = PODACI B

ADRESA_PIO+2 = KONTROLNI REGISTAR A

ADRESA_PIO+3 = KONTROLNI REGISTAR B

Kontrolne riječi

PIO sklop ima 4 različite kontrolne riječi: za izbor načina rada, za postavljanje kontrole prekida, za brzo omogućavanje-onemogućavanje prekida i prekidni vektor. PIO automatski prepoznaje o kojoj se kontrolnoj riječi radi iako se sve upisuju u isti registar. Riječ se prepoznaje na temelju njenih najnižih bitova (svaka kontrolna riječ ima drugu kombinaciju bitova).

53

Kontrolna riječ za izbor načina rada

Prilikom inicijalizacije porta najprije se mora upisati kontrolna riječ za izbor načina rada. Ta riječ uvijek mora imati u najniža četiri bita upisano 1111.

D7	D6	D5	D4	D3	D2	D1	D0
M1	M2	X	X	1	1	1	1

D7	D6	način rada
0	0	0-izlazni način
0	1	1-ulazni način
1	0	2-dvosmjerni način
1	1	3-kontrola bitova

Kontrolna riječ za postavljanje za postavljanje kontrole prekida

D7	D6	D5	D4	D3	D2	D1	D0
INTER- UPT ENABLE	AND/ OR	HIGH LOW	MASKE FOLL- OWS	0	1	1	1

bit 7: 1: Omogućiti generiranje prekida
0: Onemogućiti generiranje prekida

54

(Slijedeća 3 bita (bit 6, bit 5 i bit 4) upotrebljavaju se samo kod načina 3)

bit 6: Ovaj bit određuje hoće li prekid biti generiran kada su svi bitovi koji se ispisuju postavljeni u aktivno stanje ili kada je bilo koji od njih postavljen u aktivno stanje.

1: Svi izabrani bitovi moraju biti u zadanom stanju (logički I).

0: Barem jedan izabrani bit mora biti u zadanom stanju (logički ILI).

bit 5: 1: Aktivno stanje bita predstavlja logička jedinica.

0: Aktivno stanje bita predstavlja logička nula.

bit 4: 1: Slijedeća upisana riječ bit će riječ maske u kojoj logička nula upisana na mjesto nekog bita znači da će se njegovo stanje ispitivati prije generiranja prekida.

0: Nakon te riječi ne očekuje se maska.

Kontrolna riječ za brzo omogućavanje-onemogućavanje prekida

	D7	D6	D5	D4	D3	D2	D1	D0
INTER- UPT ENABLE		X	X	X	0	0	1	1

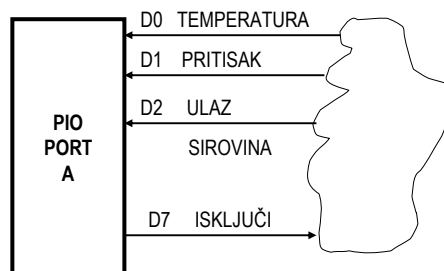
55

Prekidni vektor

D7	D6	D5	D4	D3	D2	D1	D0
V7	V6	V5	V4	V3	V2	V1	0

Primjer programiranja

Na PIO sklop, koji se nalazi na adresi 10(16), spojen je uređaj koji kontrolira rad nekog stroja. Napišite program za kontrolu procesa tako da se stroj automatski isključuje kada neka od kontrolnih vrijednosti (temperatura, pritisak, ulaz sirovina) prijeđe dozvoljenu granicu. U trenutku pojave nedozvoljenog stanja Detektor na ulaz u PIO postavlja vrijednost logičke jedinice. Povezivanje PIO sklopa i procesa prikazano je na slici.



56

Ulazi: D0 = 0 TEMPERATURA ISPRAVNA
 D0 = 1 TEMPERATURA PREKORAČENA

 D1 = 0 PRITISAK ISPRAVAN
 D1 = 1 PRITISAK PREKORAČEN

 D2 = 0 SIROVINE POSTOJE NA ULAZU U STROJ
 D2 = 1 SIROVINA NEMA

 Izlaz: D7 = 0 STROJ ISKLJUČEN
 D7 = 1 STROJ UKLJUČEN

Rješenje:

ORG 0
 PIOAD EQU 10
 PIOBD EQU 11
 PIOAC EQU 12
 PIOBC EQU 13

INIT DI ;DI za vrijeme inicijalizacije
 LD A,FF ;Inicijalizacija PIO sklopa
 OUT (PIOAC),A ;PIOA kontrola bitova
 LD A,7F

57

 OUT (PIOAC),A ;Bitovi 0-6 ulazni, 7 izlazni
 LD A,3
 OUT (PIOAC),A ;Onemogući prekid

 GLAVNI: LD A,80
 OUT (PIOAD),A ;Uključenje stroja
 PETLJA: IN A,(PIOAD) ;Učitavanje stroja
 AND A,7
 JR Z,PETLJA ;Ako je sve u redu ,
 ;ponovi petlju
 XOR A
 OUT (PIOAD),A ;Isključivanje stroja
 HALT
 END

58

```

; Program uzima podatak s ulaznog porta PIO sklopa i
; smanjuje ga za 1 dok ne bude 1, onda stane s prikazom
; ***** POCETNE DEFINICIJE *****
;
; Postavljanje stoga:
STOG EQU 67FFH ; dno stoga = vrh RAM-a
;
; Adrese I/O jedinice PIO:
PIOAC EQU 11H ; upravljacki
PIOBC EQU 13H ; registri
PIOAD EQU 10H ; registri
PIOBD EQU 12H ; podataka
;
; Konstante:
IZPORT EQU 0FH ; upravljacka rijec za izlazni port PIO
ULPORT EQU 4FH ; upravljacka rijec za ulazni port
SEKUND EQU 1350H ; broj ponavljanja petlje za cekanje
ADRESS EQU 6222H ; pomocna adresa
;
; ***** PROGRAM *****
;
ORG 0000H ; pocetak progama = pocetak EPROM-a
DI ; onemoguci prekid
LD SP,STOG ; postavi dno stoga u registar SP
CALL INIC ; poziv potprograma za inicijalizaciju PIO sklopa
;
;
; Glavni program:
; =====
;
XOR A ; brise C zastavicu
IN A,(PIOAD) ; ucitava podatak iz akumulatora
JP JEDAN ; skok na labelu JEDAN
;
KRAJ: CALL CEKAJ ; poziv potprograma CEKAJ
POP AF ; vraca podatak sa stoga
OUT (PIOBD),A ; salje podatak na izlazni port
LD B,A ; stavi podatak u B zbog DJNZ
DJNZ JEDAN ; smanjui B do 0 i
HALT ; stani
;
JEDAN: DEC A ; smanji podatak za 1
PUSH AF ; stavlja podatak na stog
JP KRAJ ; skok na labelu KRAJ

```

```

;
;
; Inicijalizacija PIO sklopa:
; =====
;
INIC: LD A,ULPORT ; inicijalizacija porta A kao ulaznog
OUT (PIOAC),A ; - || -
IN A,(PIOAD) ; probno citanje
LD A,IZPORT ; probno pisanje
OUT (PIOBC),A ; - || -
OUT (PIOBD),A ; - || -
RET ; povratak iz potprograma
;
;
; Vremenska petlja:
; =====
;
;
CEKAJ: LD BC,0000H ; upisuje nulu u registarski par BC
PETLJA: LD HL,SEKUND ; upisuje u HL broj ponavljanja petlje
SBC HL,BC ; oduzima BC od HL
INC BC ; povecaj BC za jedan
LD IX,(ADRESS) ; naredbe za trosenje vremena
LD IY,(ADRESS) ; - || -
NOP ; - || -
NOP ; - || -
JR NZ,PETLJA ; vrati se na PETLJA ako nije BC=HL
RET ; povratak iz potprograma ako je BC=HL
;
END ; zavrsetak programa

```