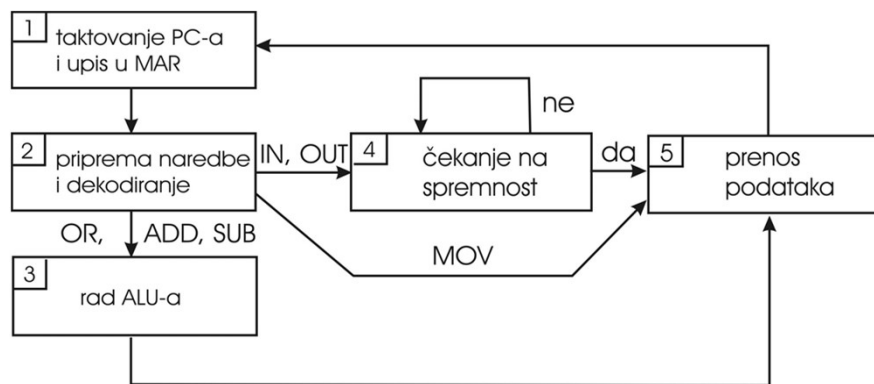


## Stanje čekanja

- neuskладjenost brzine rada procesora i događaja u spoljnom svetu. Brzina rada procesora je za nekoliko redova veličine veća od brzine kojom se dešavaju događaji u spoljnom svetu

- najjednostavnije rešenje za problem sinhronizacije leži u uvođenju **linije spremnosti** (ready line). Stanjem na ovoj liniji spoljni uređaj, kome se procesor obraća ( U/I uređaj, memorija, ...), ukazuje da je spreman za prenos podataka. Procesor testira stanje na ovoj liniji i završava naredbu koju je započeo tek kada utvrdi da je spoljni uređaj spreman za razmenu.

- period čekanja je celobrojni umnožak taktnog perioda.



Slika 15. Mašina kod koje postoji stanje čekanja na spremnost

S obzirom da dijagram prikazuje 5 različitih stanja, **brojač stanja** se realizuje kao trostepeni brojač po modulu 5. Kod starog dijagrama “load” je bio aktivan samo kada se prelazilo iz 2→4 za naredbe koje nisu koristile ALU. Ovde “load” ima složenu funkciju. Postavlja brojač stanja:

- iz 2→4 (IN, OUT)
- iz 2→5 (MOV, NOP)
- iz 3→5 (ADD, SUB, OR, ...).

### **naredbe koje koriste obraćanje memoriji**

Veliki broj procesora poseduje naredbe koje koriste neposredno obraćanje memoriji.

Tipična je naredba za neposredno kopiranje MVI

MVI A, 65 - dvobajtna naredba

- prvi bajt – operacioni kod naredbe
- drugi bajt – neposredni podatak
- Zbog ovoga, dekođer naredbi se mora tako projektovati da se drugi bajt naredbe smesti iz programske memorije u registar

(heksadekadno) lokacije	(binarni zapis) sadržaj	mnemonik	komentar
0000:	10010001	IN 1	; (A) ← (prek.)
0001:	10110100	MOV B,A	; (B) ← (A)
0002:	10010001	IN 1	; (A) ← (prek.)
0003:	00010001	ADD A,B	; (A) ← (A)+(B)
0004:	10110100	MOV B,A	; (B) ← (A)
0005:	11000000	MVI A,65	; (A) ← 65H
0006:	01100101		;
0007:	00010001	ADD A,B	; (A) ← (A)+(B)
0008:	10101000	OUT 2	; LED ← (A)
0009:	????????		

Šta bi mogla biti sledeća naredba?

- HLT nakon čijeg izvršenja se zaustavlja rad
- možemo obezbediti da se programska sekvenca neprekidno ponavlja (JMP 0000) koristeći naredbu bezuslovnog grananja

0009:	11110000	JMP 0000	trobajtna
000A:	00000000		naredba
000B:	00000000		

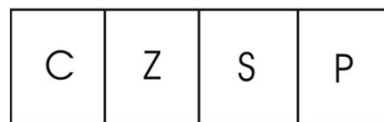
## naredbe za izmenu toka programa

- - bezuslovne
- - uslovne

Uslovi se postavljaju nakon izvršenja aritmetičkih i logičkih operacija. Npr. prilikom sabiranja 2 broja može se javiti prenos.

Prenos se detektuje, i ako postoji, postavlja se marker (fleg – flag) za prenos (c- carry). Postoje i drugi uslovi koji se sagledavaju (nula, znak, parnost) i u čije se markere upisuje stanje nakon izvršenja neke aritmetičke ili logičke operacije.

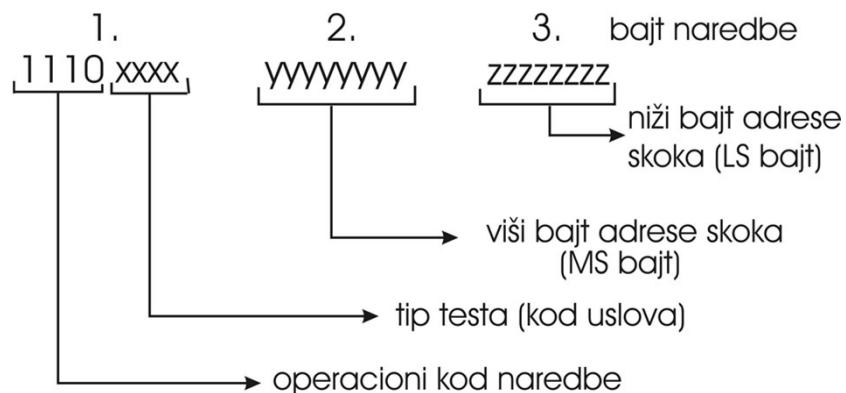
Markeri su objedinjeni u registru uslovnih markera. Svaki bit ima svoje značenje.



C- carry  
Z- zero  
S- sign  
P- parity

Slika 16. Registar markera uslova

- Registar markera uslova pridružen je ALU.
- Stanje markera uslova može se testirati naredbama za uslovnu promenu toka programa.

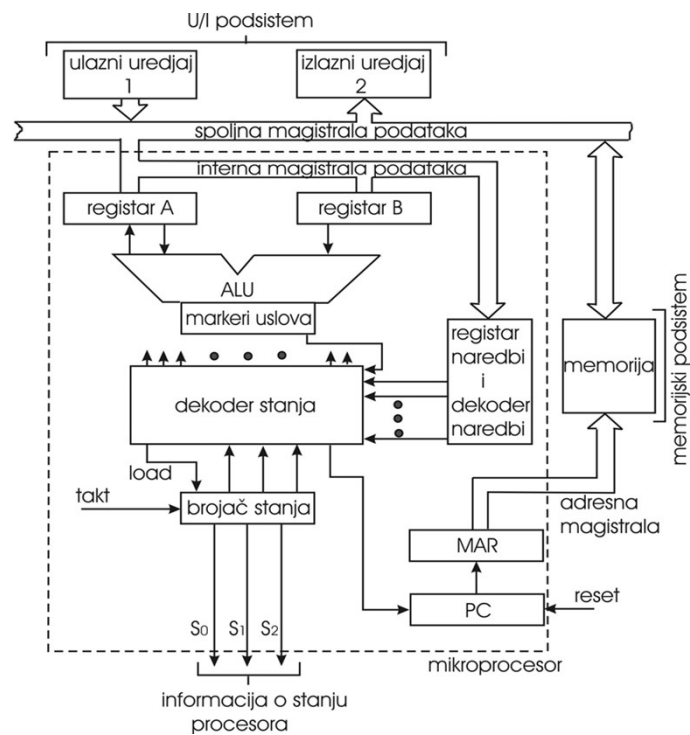


Slika 17. Format naredbe uslovnog skoka Jxx

kod uslova	mnemonik	komentar
0000	JC	; grananje ako ima prenosa
0001	JNC	; grananje ako nema prenosa
0010	JZ	; gran. ako je rezultat nula
0011	JNZ	; gran. ako rezultat nije nula
0100	JS	; gran. ako je rezultat pozitivan
0101	JNS	; gran. ako je rez. negativan
0110	JP	; gran. ako je parna parnost
0111	JNP	; gran. ako je neparna parnost

0000:	IN 1
0001:	MOV B, A
0002:	IN 1
0003:	ADD A, B
0004, 0005, 0006:	JC 000B
0007:	MOV B, A
0008, 0009:	MVI A, 65
000A:	ADD A, B
000B:	OUT 2
000C, 000D, 000E:	JMP 0000

Slika 18. Blok šema mikroračunarskog sistema



- Uočavamo spoljnu magistralu podataka i internu magistralu podataka.
- Preko spoljne magistrale prenose se :
  - a) podaci iz memorije uključujući i naredbe
  - b) podaci za U/I uređaje.
- Upravljački signali koji se generišu od strane mikroprocesora određuju smer prenosa podataka na magistrali za podatke.
- Na slici se uočavaju tri celine koje čine mikroračunarski sistem:
  - mikroprocesor (CPU)
  - memorija (memorijski podsistem)
  - U/I podsistem

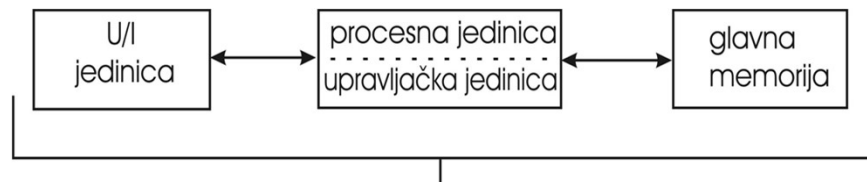
## **HARDVERSKA STRUKTURA MIKROPROCESORSKIH SISTEMA**

Tri hardverske komponente čine osnovnu strukturu mikroprocesorskog sistema:

mikroprocesor – primarni upravljački deo sistema ( CPU- Central Processing Unit)

glavna memorija – namenjena za čuvanje programa i podataka

U/I jedinica – preko koje sistem komunicira sa spoljnim okruženjem.



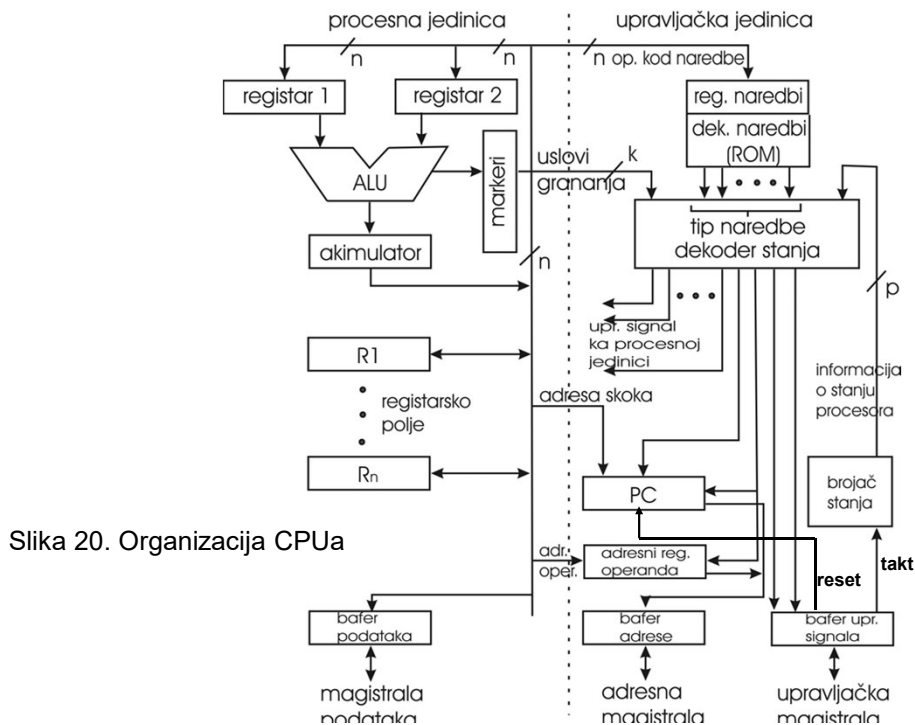
Slika 19. Mikroračunarski sistem

**CPU** je centralni deo sistema koji određuje ponašanje celog sistema. CPU čita naredbe iz glavne memorije, interpretira i preuzima određene akcije.

Većina naredbi ukazuje na vrlo jednostavne akcije:

- čitanje podataka iz U/I jedinice
- premeštanje iz jednog registra u drugi
- obavljanje aritmetičke ili logičke funkcije nad sadržajem nekog registra
- čitanje ili upis podataka u glavnu memoriju





Slika 20. Organizacija CPUa

CPU je organizovan oko jedinstvene interne magistrale. Procesnu jedinicu čine:

- **ALU** obavlja osnovne aritmetičko-logičke operacije nad operandima koji joj se dostavljaju
- **registarsko polje** – registri se mogu koristiti kao pokazivači adresa ili kao registri za čuvanje i prenos podataka
- **akumulator** – radni registar specijalne namene. Rezultat najvećeg broja aritmetičko-logičkih operacija smešta se u akumulator, a njegov sadržaj često se koristi kao jedan od operandi
- **registar 1 i registar 2** – registri za privremeni smeštaj operandi
- **markeri** – registar indikatora
- **bafer podataka** – razdvojni stepen za dvosmerni prenos podataka

**Upravljačka jedinica** je složeniji deo CPU- a.

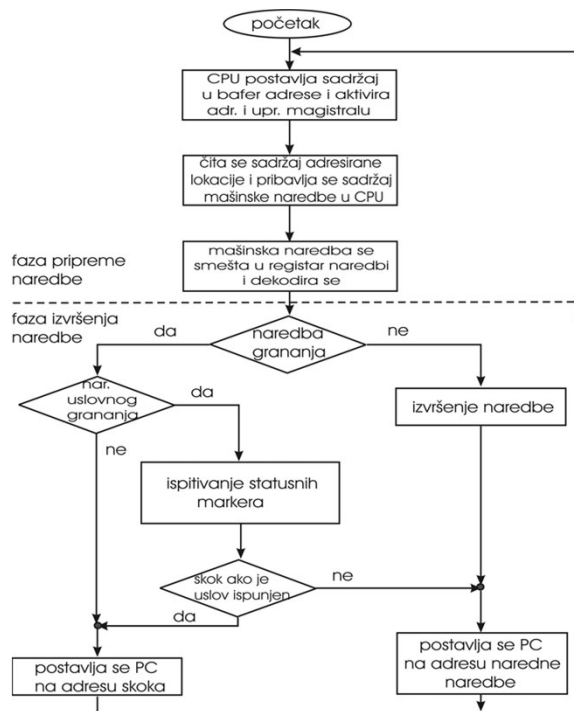
- Aktiviranjem **reset** signala, PC se postavlja u unapred definisano stanje ( ukazuje na lokaciju gde je smeštena prva naredba u programu).
- Deaktiviranjem signala **reset** (pripada signalima upravljačke magistrale), sadržaj PC- a se postavlja u bafer adrese i aktiviraju se signali adresne i upravljačke magistrale.
- Sa adresirane memorijske lokacije čita se op. kod prve naredbe i smešta se u registar naredbi.
- Upisom u registar naredbi aktivira se dekodер naredbi koji na izlazu generiše upravljačke signale pomoću kojih se ukazuje na tip naredbe ( MOV, LOAD,...).

Izlazi dekodera naredbi vode se na dekodер stanja koji na osnovu ulaznih signala (usl.grananja, tip naredbi, inf. o stanju) generiše upravljačke signale pomoću kojih se upravlja radom ostalih blokova CPU-a:

a) inkrementira ili postavlja na novu vrednost PC sa ciljem da ukaže na adresu naredne naredbe ili operanda u programu

b) prihvata, testira i generiše upravljačke signale pomoću kojih usmerava rad ostalih delova upravljačke jedinice sa ciljem da se započeta naredba uspešno završi, privremeno zaustavi ili prekine

c) generiše upravljačke signale za rad procesne jedinice



Slika 21. Dijagram toka aktivnosti CPU-a u toku izvršenja jedne naredbe

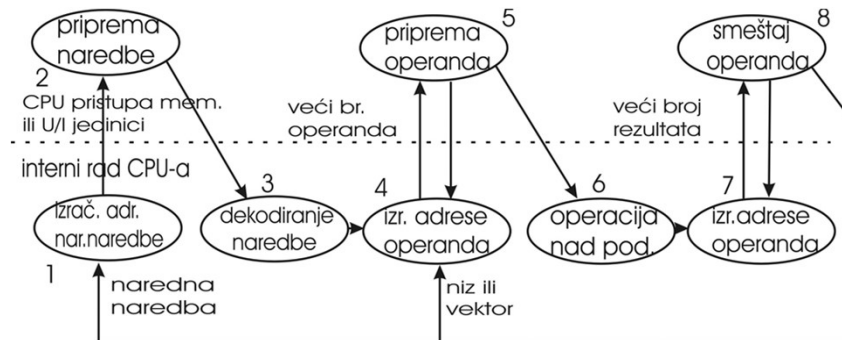
Kod svake naredbe razlikujemo dve faze:

1. faza pripreme, pribavljanja naredbe (op.fetch)
2. faza izvršenja (execution).

I priprema i izvršenje traju veći broj taktnih intervala, pri čemu za svaku naredbu prvo sledi faza pripreme, a zatim faza izvršenja.

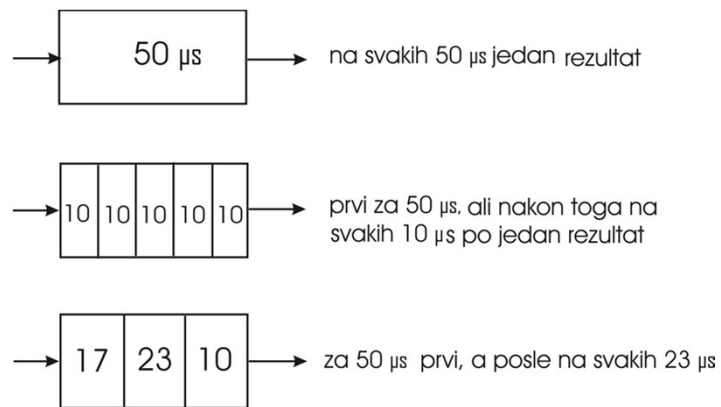
U toku faze izvršenja pojedinih naredbi moguće je da se obraćanje memoriji vrši više puta.

Detaljan pregled toka izvršenja naredbe dat je sledećim dijagramom stanja:



Slika 22. Dijagram stanja za tok izvršenja naredbe

- a) izračunavanje adrese naredbe – određuje se
- adresa naredne naredbe koja će se izvršiti
- b) priprema naredbe – CPU čita naredbu iz
- memorijske lokacije
- c) dekodiranje naredbe
- d) izračunavanje adrese operanda – ako se tipom
- operacije predviđa obraćanje operandu u
- memoriji ili ako je dostupan preko U/I
- određuje se adresa operanda
- e) priprema operanda – pribavlja se operand iz
- memorije ili se on čita iz U/I jedinice
- f) operacija nad podacima – obavlja se
- predviđena operacija
- g) smeštaj operanada – upisuje se rezultat u
- memoriju ili u U/I jedinicu



Slika 23. Princíp protočnosti

Kod mikroprocesora iAPx86 postoji vremensko preklapanje između faze pripreme i faze izvršenja čime se ubrzava rad procesora. To je omogućeno time što je CPU podeljen na BIU i EU.