


# Pogled na računar iznutra - Struktura i funkcija RS



# Binarna predstava informacija

- **Bit** - 0 ili 1
- **Nibble** - niz od 4 bita
- **Byte** - niz od 8 bitova
- **Word** - zavisi od mašine

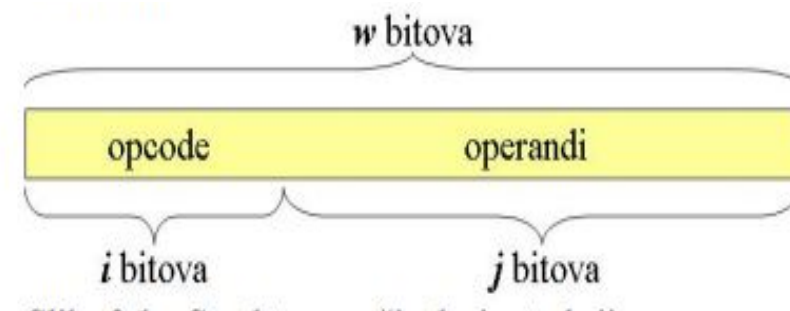
Instrukcije računara se takođe kodiraju binarno. Skup različitih mašinskih instrukcija koje mogu da se izvršavaju na nekom procesoru naziva se **skup instrukcija procesora**.



# Instrukcija


Instrukcije se sastoje iz nekoliko delova:

- šta treba uraditi, **opcode**
- gde se nalaze operandi
- gde smestiti rezultat
- gde se nalazi sledeća instrukcija

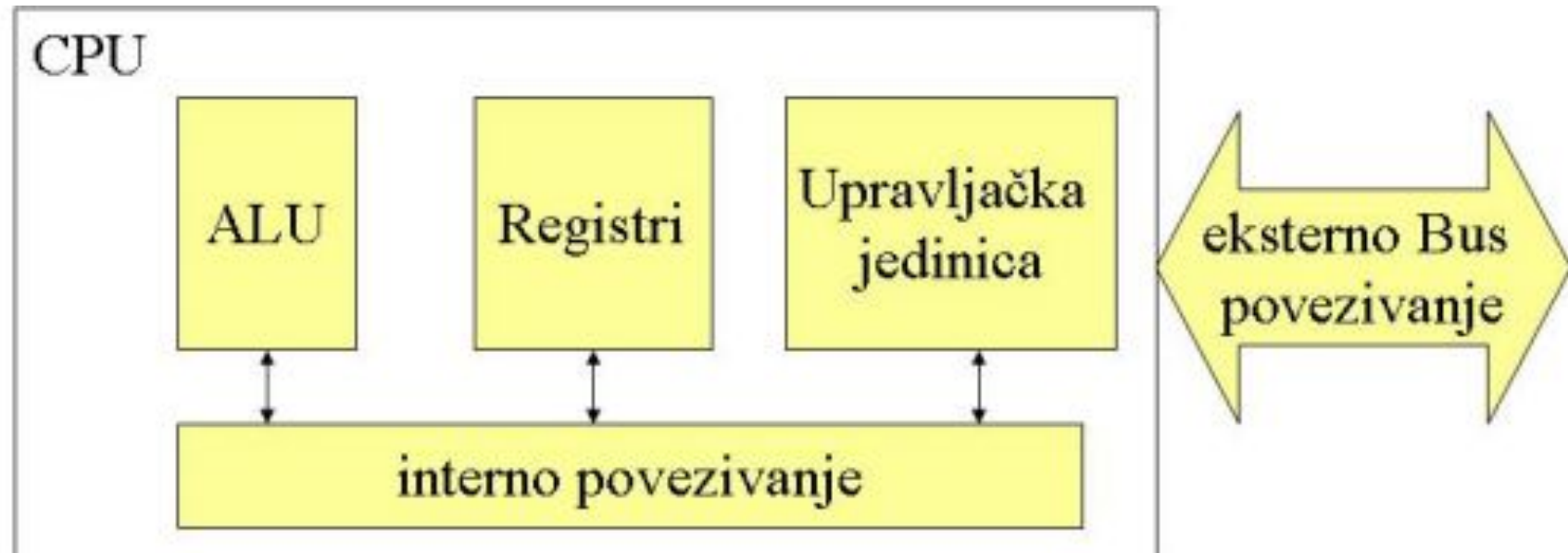


# Funkcije i struktura centralnog procesora (CPU)

U centralom procesoru osnovne računarske funkcije vrše sledeći delovi.

- OBRADA – Aritmetičko logička jedinica (ALU)
  - SKLADIŠTENJE - Registri
  - UPRAVLJANJE - Upravljačka jedinica
  - PREMEŠTANJE - Interna procesorska magistrala (povezuje upravljačku jedinicu, ALU i registre), a sistemska magistrala (povezuje CPU sa drugim komponentama).
- 

# Funkcije i struktura centralnog procesora (CPU)




# Funkcije i struktura centralnog procesora (CPU)

- Prilikom izvršavanja programa, naizmenično se menjaju **faza pripreme** i **faza izvršavanja**.



# Funkcije i struktura memorije


- Osnovna uloga **skladištenje**. Memorija (primarna, RAM) se sastoji od memorijskih registara fiksne dužine.
  - **PREMEŠTANJE** podataka u okviru same memorije vrši se preko internih veza, a prenošenje podataka između memorije i drugih uređaja obavlja se preko sistemske magistrale.
- 

# Funkcije i struktura memorije

- OBRADA koju vrši memorija je ograničena obrada u poređenju sa onom koju vrši CPU. Osvežavanje, pronalazak grešaka, ispravka grešaka.
- UPRAVLJANJE
  - write – kopira ulaznu vrednost kao novi sadržaj lokacije,
  - read – izdaje (ali ne menja) sadržaj lokacije.



# Funkcije i struktura ulazno/izlaznih uređaja

- Svaki uređaj ima **kontroler**.
  - Kontroler je povezan sa ostalim delovima sistema preko registara koji se zovu **portovi**.
  - Premeštanje, obrada i upravljanje slično kao kod RAM memorije.
  - Port podataka, statusni port, upravljački port
- 

# Elementi rada CPU (uprošćeni IAC)

uIAS Memorija

Memorija	
adresa	sadržaj
000	1 2 3 4
001	9 0 7 5
002	0 9 9 7
.	
.	
.	
997	3 1 8 0
998	6 7 2 4
999	9 9 3 2

Registri uIAS procesora

CPU Registri	
PC	0 8 2 5
IR	2 0 0 1
AC	0 0 0 0
MAR	0 8 2 4
MBR	2 0 0 1

# Elementi rada CPU (uprošćeni IAC)

Konvencija

$2 \rightarrow 4$

$(2) \rightarrow 4$


$((2)) \rightarrow 4$

# Instrukcijski ciklus uIAS

- Faza pripreme:
  - $(PC) \rightarrow MAR$  // postavlja adresu donošenja
  - $((MAR)) \rightarrow MBR$  // donosi instrukciju
  - $(MBR) \rightarrow IR$  // pamti instrukciju u IR
  - $(PC) + 1 \rightarrow PC$  // postavlja adresu sledećeg donošenja

# Instrukcijski cikula u IAS

Razlikujemo sledeće grupe instrukcija u fazi izvršavanja.

- Instrukcije za interno prenošenje podataka prenose podatke između procesora i glavne memorije
  - Instrukcije za eksterno prenošenje podataka prenose podatke između procesora i ulazno/izlaznih uređaja
  - Instrukcije obrade izvršavaju u ALU neke aritmetičke ili logičke operacije nad podacima
  - Instrukcije za predaju upravljanja uslovno ili bezuslovno menjaju sadržaj PC registra, npr. instrukcije skokova
- 

# Prelazak na asemblerski kod

- 1xxx - Puni AC vrednošću sa memorijske adrese xxx
- 2xxx - Upisuje vrednost AC na memorijsku adresu xxx
- 3xxx - Dodaje sadržaj sa memorijske adrese xxx na AC

Nepraktično za ljude. Koristimo simbolički zapis  
(mnemonike)

LDA xxx (load accumulator)

STA xxx (store accumulator)


ADD xxx (addition)

# Primer

CPU Registri	
PC	0 2 2 5
IR	1 8 2 6
AC	9 9 9 9
MAR	0 8 2 6
MBR	2 0 0 1

Memorija	
adresa	sadržaj
---	----
224	1 8 2 6
225	1 8 2 7
226	3 8 2 8
227	2 8 2 8
---	----
826.	9 9 9 9
827	0 0 0 1
828	0 0 0 9
829	0 0 0 0
---	----

# Još instrukcija

- 4-cifreni kod Operacija simbolički zapis
  - 4xxx Puni AC sadržajem porta sa adrese xxx IN xxx
  - 5xxx Ispisuje sadržaj AC na port sa adresom xxx. OUT xxx
  - 6xxx bezuslovno predaje upravljanje instrukciji na adresi xxx JMP xxx
- 



# Još instrukcija

- 7xxx Ako  $(AC) < 0$  predaje upravljanje instrukciji na adresi xxx, JL xxx
  - 8xxx Ako  $(AC) = 0$  predaje upravljanje instrukciji na adresi xxx, JE xxx
  - 9xxx Menja znak broju u AC (ne uzima se u obzir deo xxx), NEG
  - 0xxx Zaustavlja rad programa, HALT
- 