



# Mikroračunarski sistemi

## 20ER6004



# Terminal



# Terminal

- **Tastatura** – najčešći tip ulaznog uređaja
- Tastaturu čini veći broj označenih tastera, uređenih na određeni način
- Pritisnuti taster generiše jedinstveni binarni kod
- Tastatura se u praksi kombinuje sa nekom vizuelnom izlaznom jedinicom
- **Displej** – vizuelna izlazna jedinica (LED, LCD,...)
- **Tastatura + displej** čine jedinicu koju nazivamo - **terminal**



# Tastatura



# Tipovi tastature

Na osnovu funkcije tastera, tastature delimo na:

- tastature sa **fiksnom** funkcijom tastera i
- tastature sa **promenljivom** funkcijom tastera



# Fiksna funkcija tastera

## Prednosti:

- **Jednostavan rad** - za ostvarenje jedne funkcije pritiska se samo jedan taster i taj taster uvek obavlja istu funkciju
- Sve ponuđene (dostupne) funkcije se mogu odrediti **analizom tastature**
- Za rad tastature potrebna je relativno **mala programska podrška**
- Tasteri su uređeni u **logičke grupe**





# Fiksna funkcija tastera

## Nedostaci:

- Ako je broj funkcija veliki, biće neophodno ugraditi **veliki broj tastera**
- Ako ima mnogo tastera, potrebno je **često vizuelno pretraživanje** i pomeranje ruke i glave prema oblasti gde se nalazi taster
- **Mala promena** iziskuje **hardversku modifikaciju** (proširenje tastature za nekoliko dodatnih funkcija zahteva ugradnju novih tastera)
- Ponekad je **teško logički grupisati** tastere da bi se zadovoljili zahtevi koji važe za sve operativne procedure



# Fiksna funkcija tastera

Izbor tastature sa fiksnom funkcijom tastera je dobro rešenje:

- kada se određeni skup funkcija **često koristi**,
- kada funkcije treba da se **izvršavaju brzo**,
- kada je korektni **inicijalni izbor funkcija** kritičan da bi se ostvarila **operativnost sistema**.





# Promenljiva funkcija tastera

Mogu se realizovati na jedan od sledeća dva načina:

- Tasteri menjaju svoju funkciju, ali je pri tom broj alternativa mali ili
- „soft“ tasteri - funkcije tastera se definišu softverski i mogu se dinamički menjati (informacija o funkciji tastera obično se prikazuje na displeju).



# Promenljiva funkcija tastera

## Prednosti:

- Mali broj tastera,
- Ne zahteva mnogo vizuelnog pretraživanja,
- Ne zahteva mnogo pokreta rukom,
- Može se modifikovati promenom softvera.



# Promenljiva funkcija tastera

## Nedostaci:

- **Povećava se vreme izbora funkcija**
- **Smanjuje se jasnoća obeležavanja tastera**
- **Potreba za „promptom“ i povratnom reakcijom**
- **Duži period uvežbavanja**



# Promenljiva funkcija tastera

Izbor tastature sa promenljivom funkcijom tastera je dobro rešenje:

- kada postoji **nekoliko podskupova funkcija** koje se često korисти,
- kada **brzina unosa nije velika**,
- kada postoji odgovarajući „**prompt**“ i **povratna reakcija**.

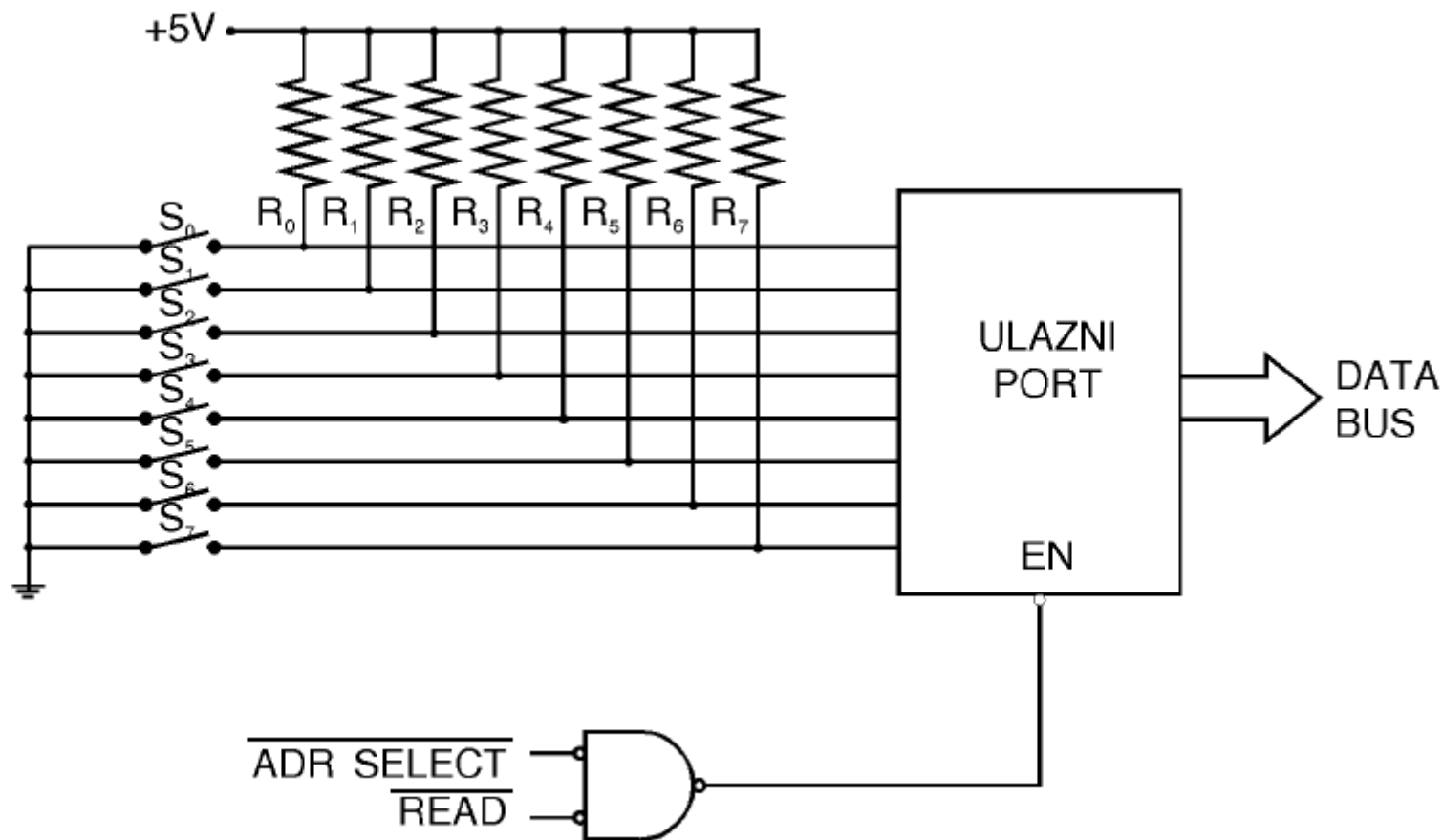


# Povezivanje tastature

Zavisno od broja tastera, najčešće se koriste sledeća dva tipa povezivanja:

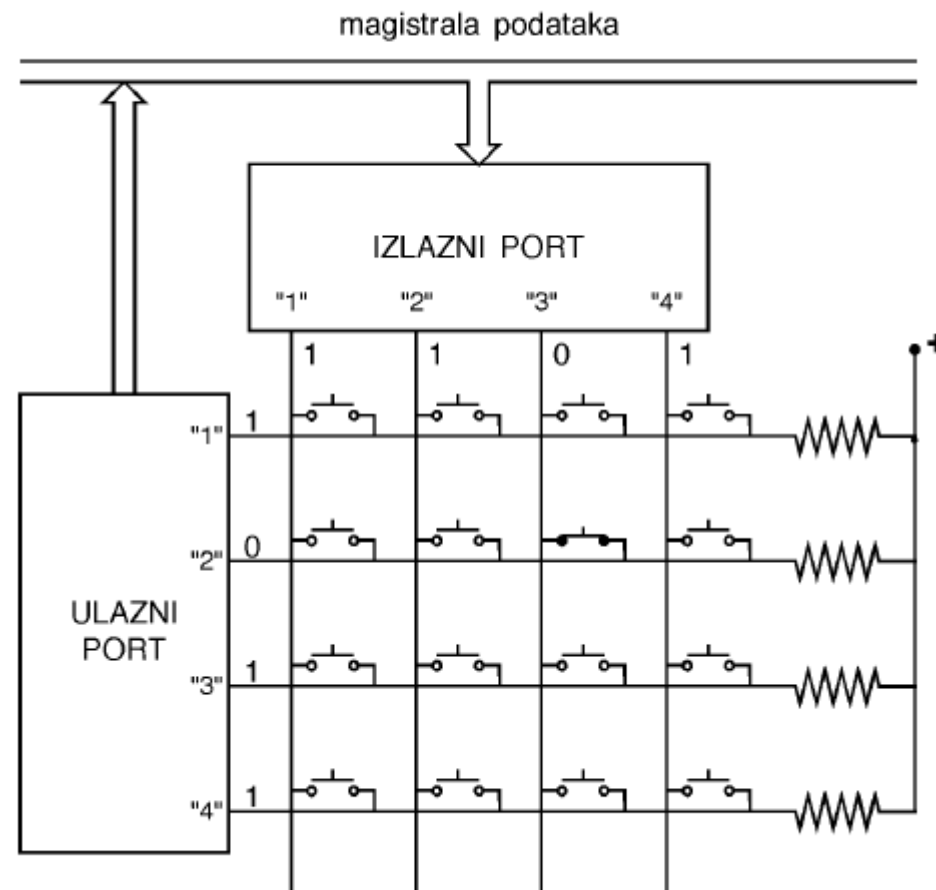
- **Nemultipleksirani** interfejs – pogodan za tastature sa malim brojem tastera ili za očitavanje postavljene konfiguracije (DIP prekidači ili kratkospajajući)
- **Multipleksirani** interfejs – pogodan za tastature sa većim brojem tastera

# Nemultiplexirani interfejs





# Multiplexirani interfejs

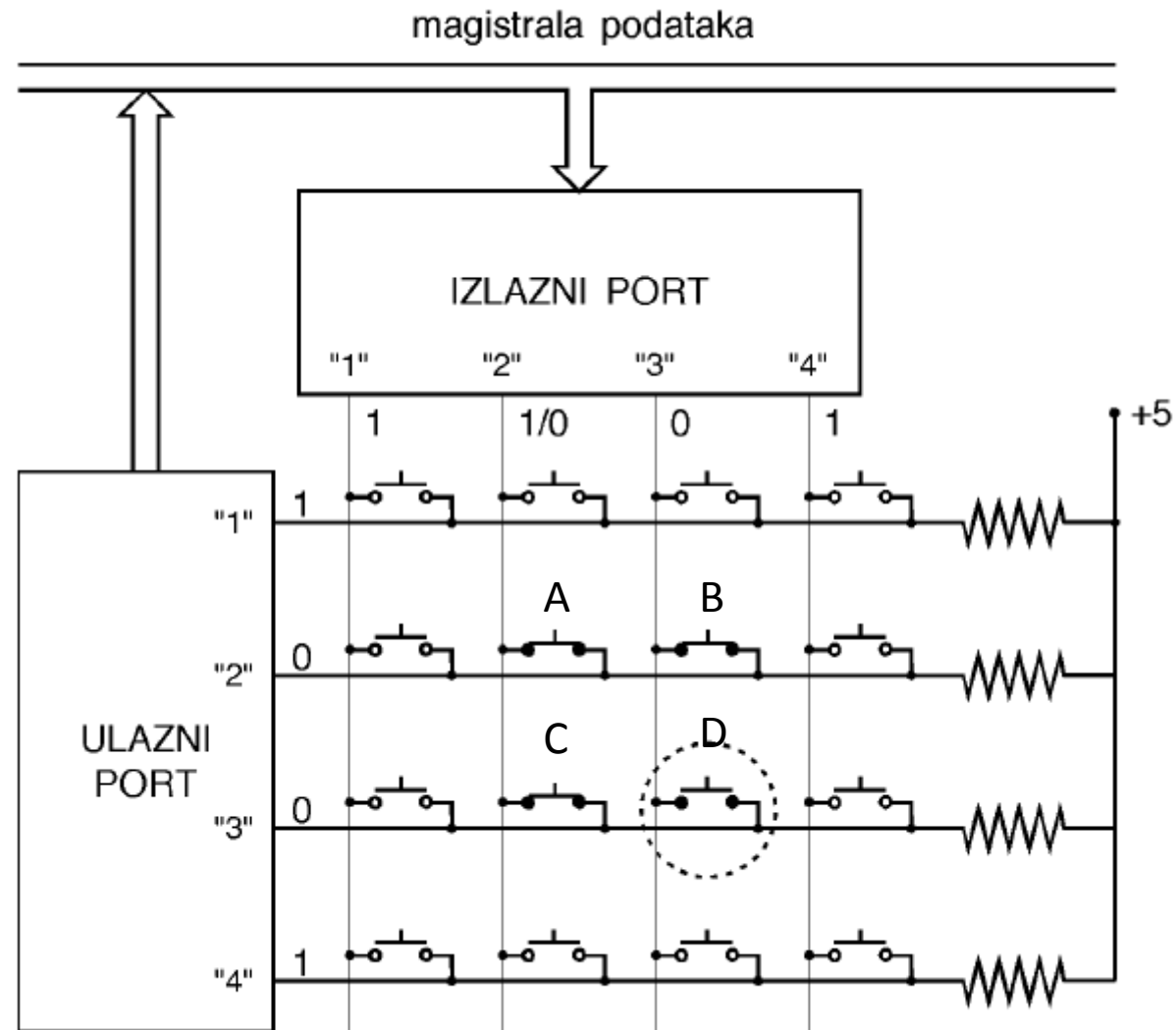




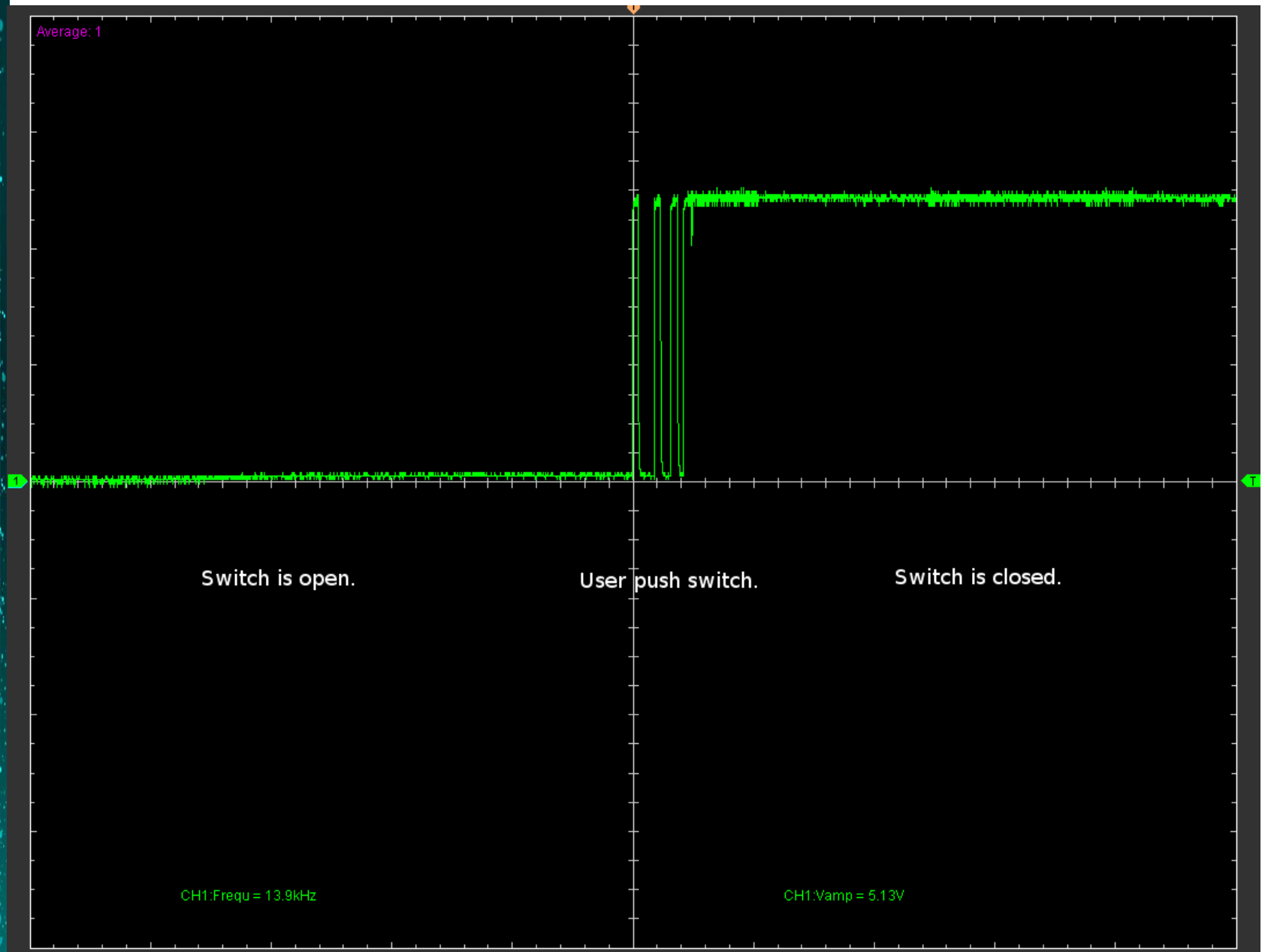
# Multiplexirani interfejs

- Potreban je manji broj U/I bitova
- Za 16 tastera treba 4-bitni ulazni i 4-bitni izlazni port (umesto 16-bitnog ulaznog porta kod nemultiplexiranog)
- Tehnika „šetajuće nule“
  - jedan pin izlaznog porta postavlja se na 0, a svi ostali na 1 (selektuje se jedna kolona)
  - pritiskom tastera u selektovanoj koloni, postavlja 0 na jednom od pinova ulaznog porta
  - vrednost na izlaznom portu se kružno pomera i time se omogućuje očitavanje tastature kolona po kolona
- Korišćenjem programabilnog tajmera, može se generisati zahtev za prekidom kojim se analizira jedna kolona tastature; ako postoji 0 na ulaznom portu, pritisnuti taster se dekodira na osnovu vrednosti na ulaznom i izlaznom portu

# „Ghost keys“ efekt



# Treperenje tastera

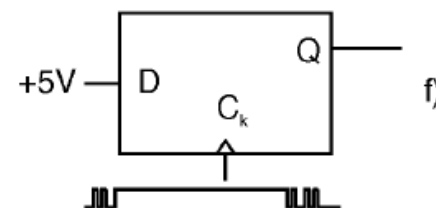
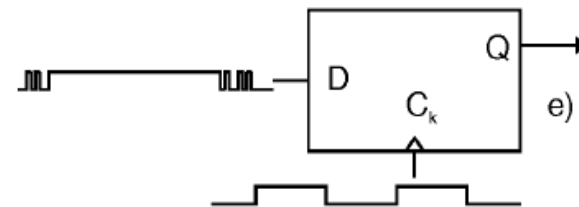
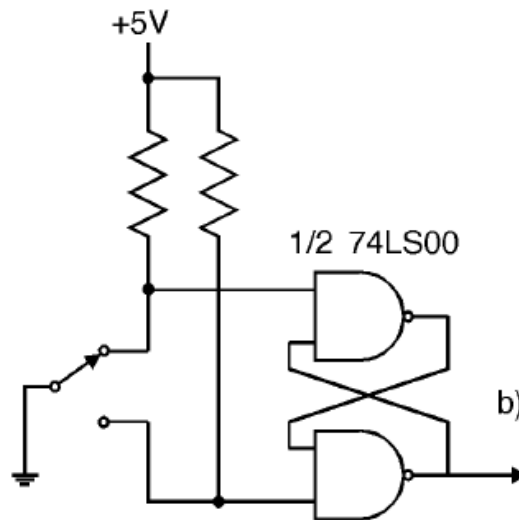
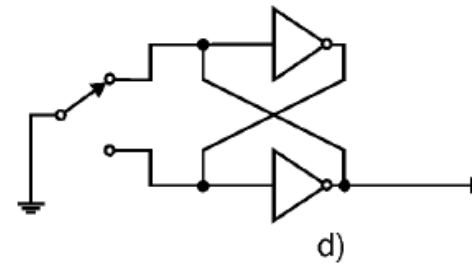
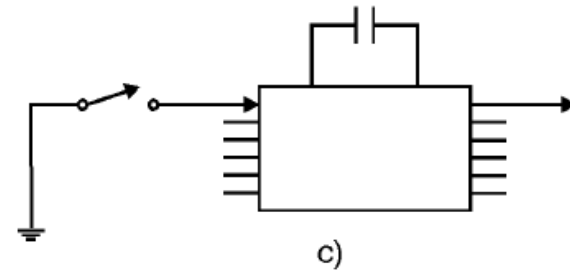
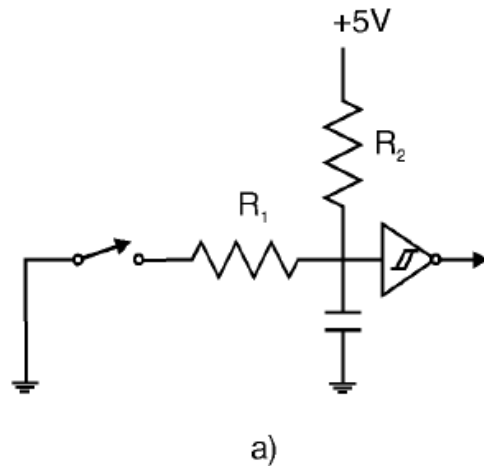




# Treperenje tastera

- Kada se pritisne ili otpusti taster, on „treperi“ (između stanja „otvoreno“ i „zatvoreno“) sve dok konačne ne ostane u odgovarajućem stanju.
- Period treperenja zavisi od konstrukcije.
- Kvalitetniji tasteri trepere od 1ms do 5ms, a manje kvalitetni od 20ms do čak 100ms.
- Problem se može rešiti hardverski ili softverski.

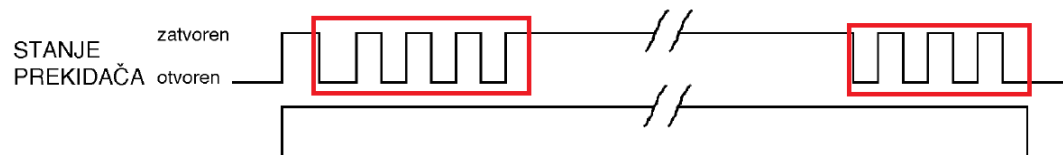
# Hardverske tehnike za eliminaciju treperenja tastera





# Softverske tehnike za eliminaciju treperenja tastera

- U potpunosti se obavlja programski
- Aktivira se brojač, čiji period odgovara trajanju treperenja
- Kada se detektuje promena stanja tastera, stanje (tj. kod tastera) se privremeno memoriše.
- Ako je po isteku brojača očitano stanje isto kao početno, smatra se da je taster u tom položaju, a kod tastera se predaje rutini za obradu.
- Može se uvesti eliminacija samo zatvaranja ili otvaranja prekidača (npr. čim prvi put pređe u zatvoreno stanje, smatra se da je zatvoren, ali se pri prelasku u otvoreno stanje odbrojava, pa ako nije otvoren nakon npr. 10ms, smatra se da je zatvoren).

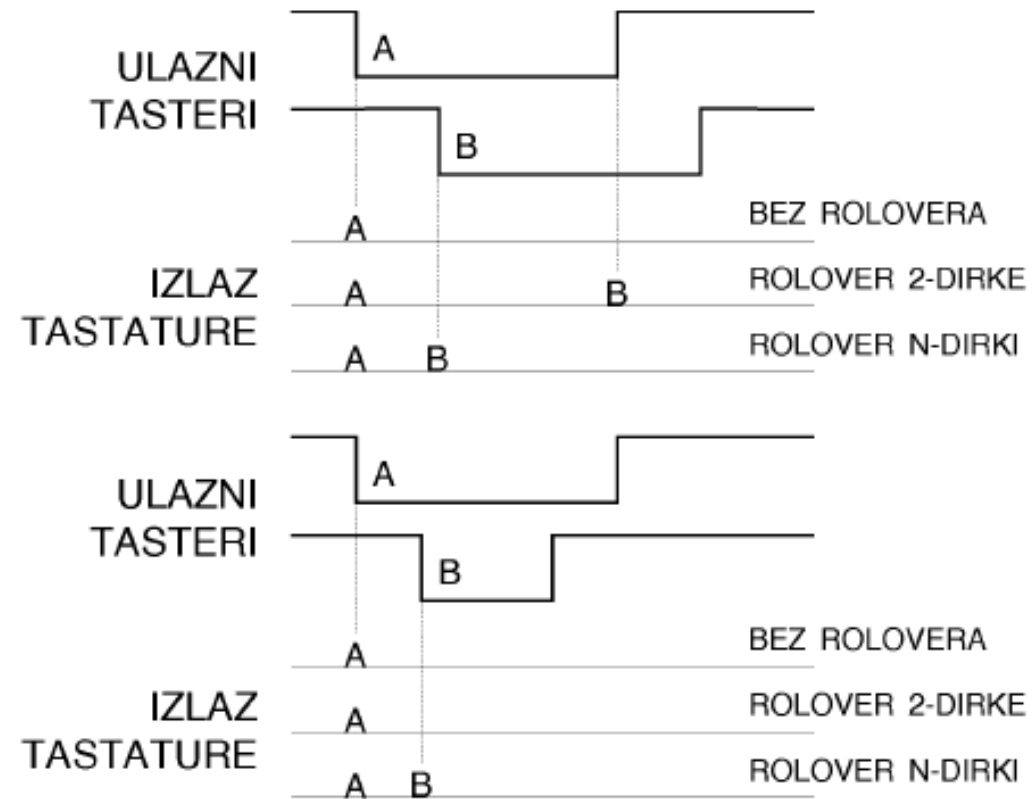




# Rollover i lockout

- Može se desiti da više tastera bude pritisnuto odjednom.
- Osobina tastature da se pritisnuti taster prihvati u trenutku kada je i neki drugi taster takođe pritisnut zove se „**rollover**“.
- Suprotno tome, tastatura kod koje nije ugrađen rollover (ova tehnika se još zove i „**two-key-lockout**“) ne dozvoljava prihvatanje narednog tastera sve dok se prethodni ne otpusti.

# „Rollover“ tehnika





# „Rollover“ tehnika

- **Two-key rollover** (poznata još i kao „**shadow rollover**“) dozvoljava da se i drugi taster pritisne, ako je prvi već pritisnut, ali drugi će generisati signal samo ako se prvi otpusti. Ako se drugi taster otpusti pre prvog, pritisak na drugi će se ignorisati. Ako se dva tastera istovremeno pritisnu, dolazi do blokiranja izlaza.
- **Three-key rollover** - obezbeđuje važeće podatke samo za dva sekrencijalna pritiska tastature. Pritisak trećeg tastera neće generisati znak na izlazu, sve dok se jedan od prva dva ne otpusti.
- **N-key rollover** - omogućava da tastatura prihvati sve pritisnute tastere i generiše važeće znakove nezavisno od broja istovremeno pritisnutih tastera. U ovom slučaju se koristi memorija u kojoj se pamti redosled pritisnutih tastera.



# Hardverski interfejs

- Sve prethodne tehnike su softverski vrlo intenzivne, jer se zahteva:
  - kontinualno analiziranje vrsta i kolona tastature,
  - dekodiranje podataka i utvrđivanje koji taster je pritisnut,
  - eliminisati treperenje tastera,
  - implementirati neku rollover ili lockout tehniku
- Potrebno je dosta vremena za realizaciju prethodnog algoritma, čak i kada nema pritisnutih tastera
- Zato se često implementira kontroler tastature
- Mi ćemo obraditi **Intel 8279** (posebna prezentacija)

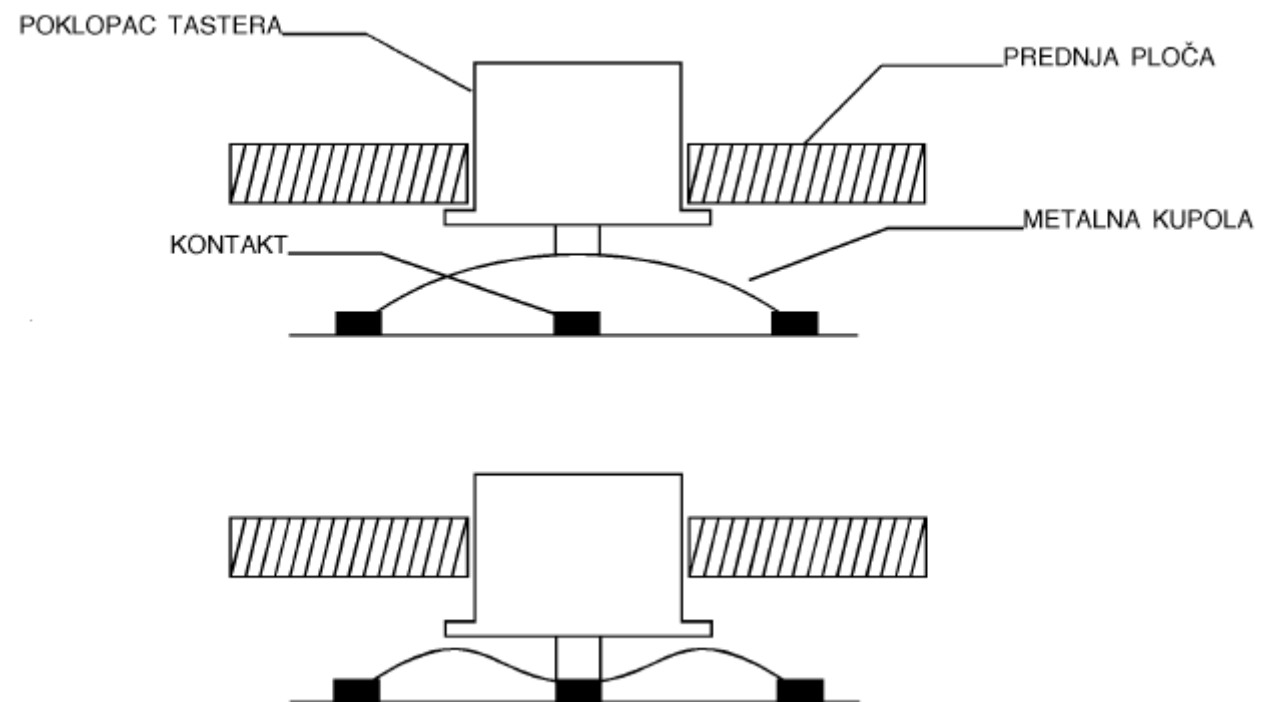


# Tehnologije prekidača

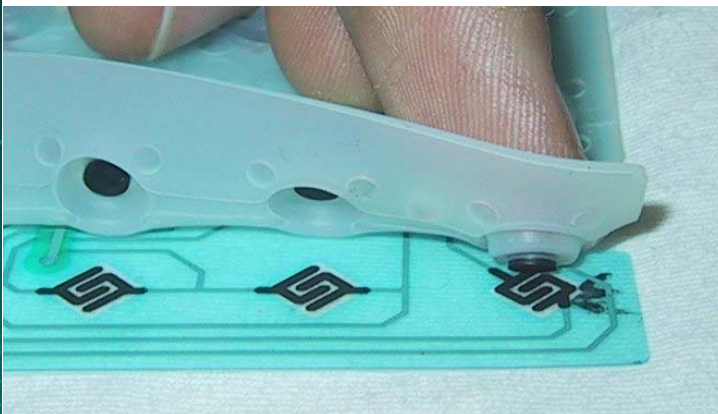
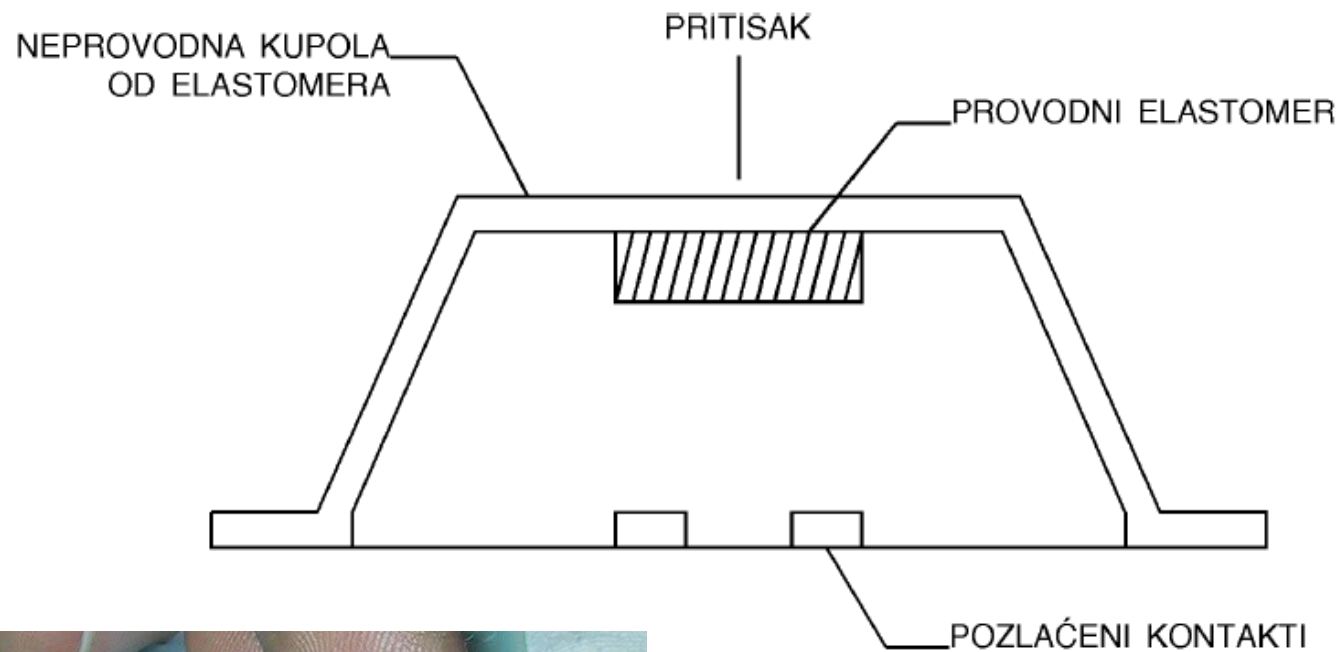
- Mehanički
- Optički
- sa Hologramnim efektom



# „Dome“ prekidač

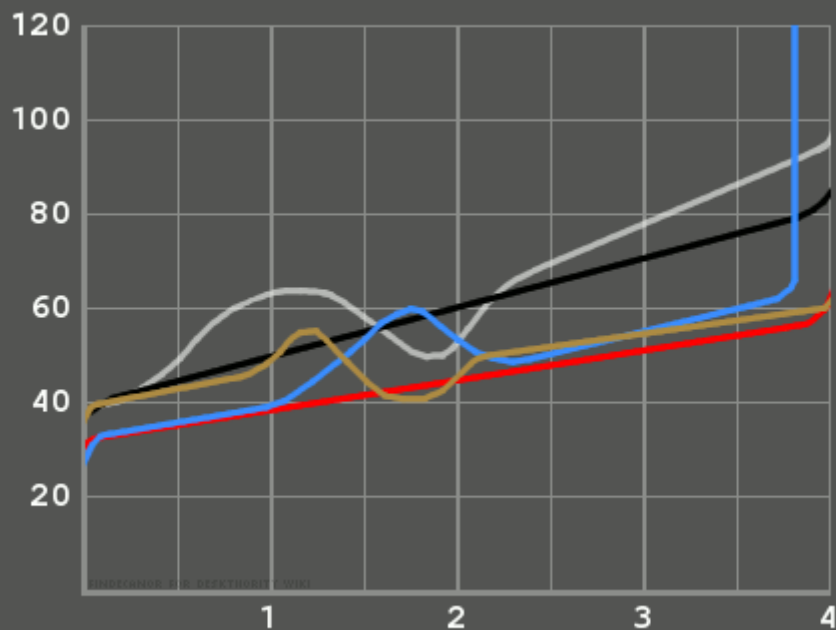


# Taster od provodne gume



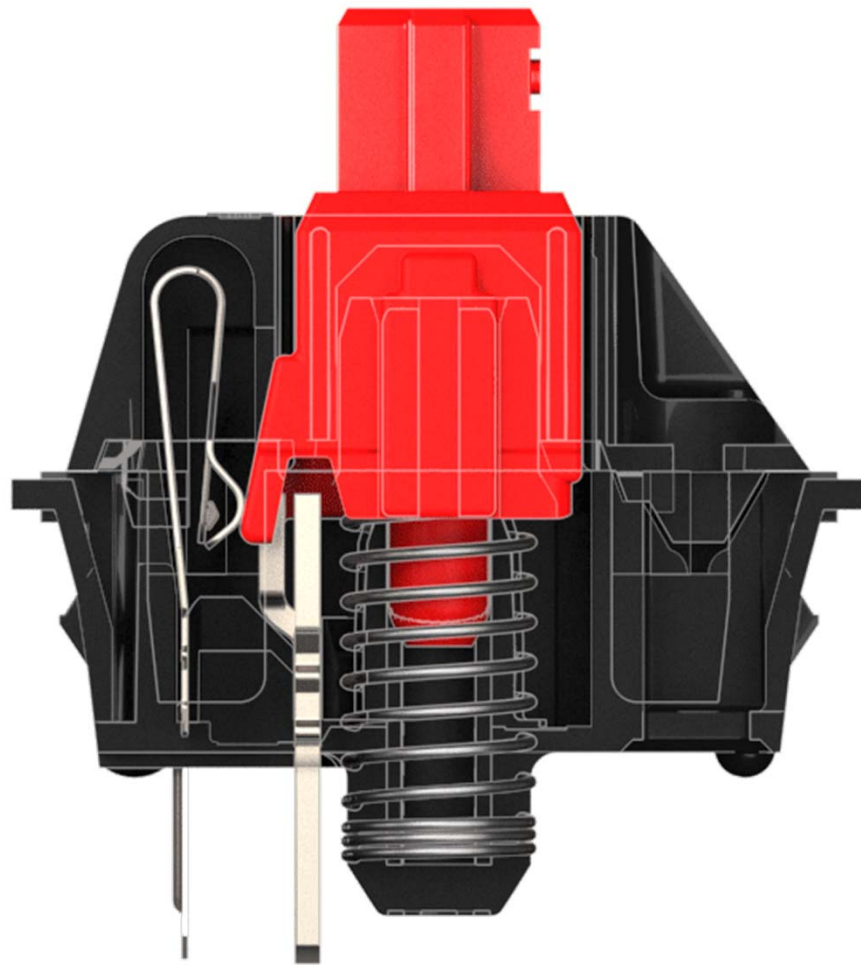
# Cherry MX tasteri

Cherry MX	<b>brown</b>	<b>blue</b>	<b>green</b>	<b>red</b>	<b>black</b>
Osećaj	pregib	klik	klik	Glatko linearan	Glatko linearan
Otpor na pritisak	srednji-nizak	nizak	visok	vrlo nizak	visok

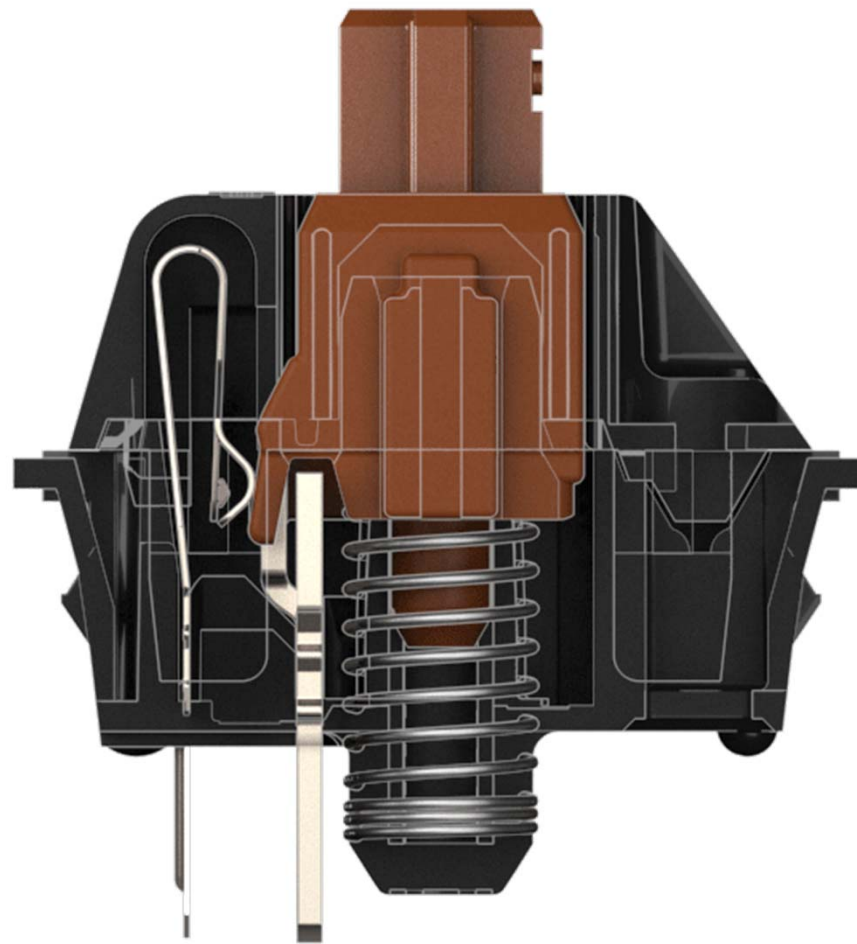


Otpor tastera  
prilikom kretanja

# Cherry MX Red tasteri

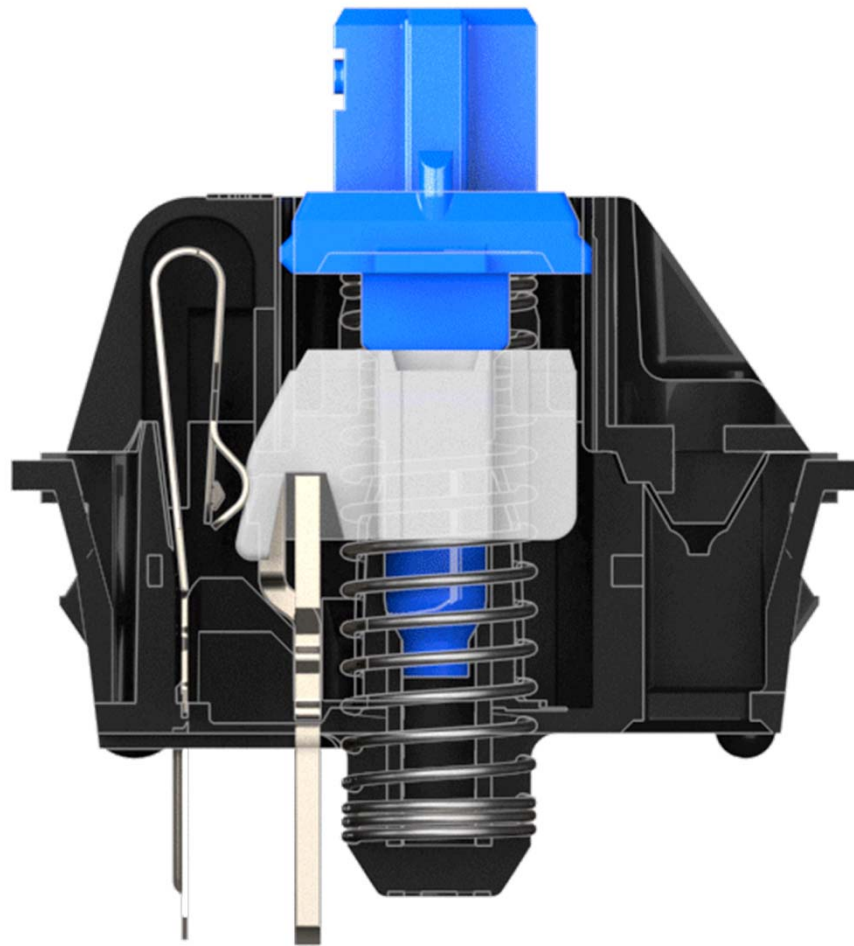


# Cherry MX Brown tasteri





# Cherry MX Blue tasteri

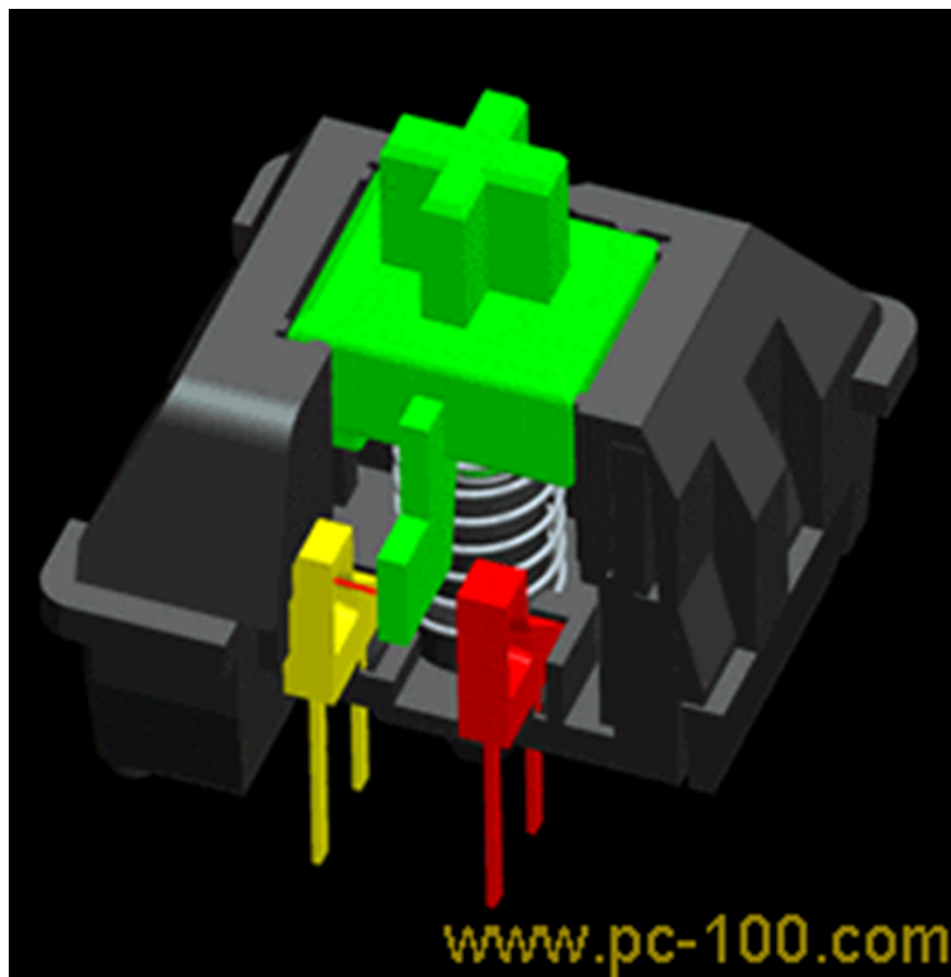




# Električni kontakt Cherry MX tastera



# Infra-red optički tasteri

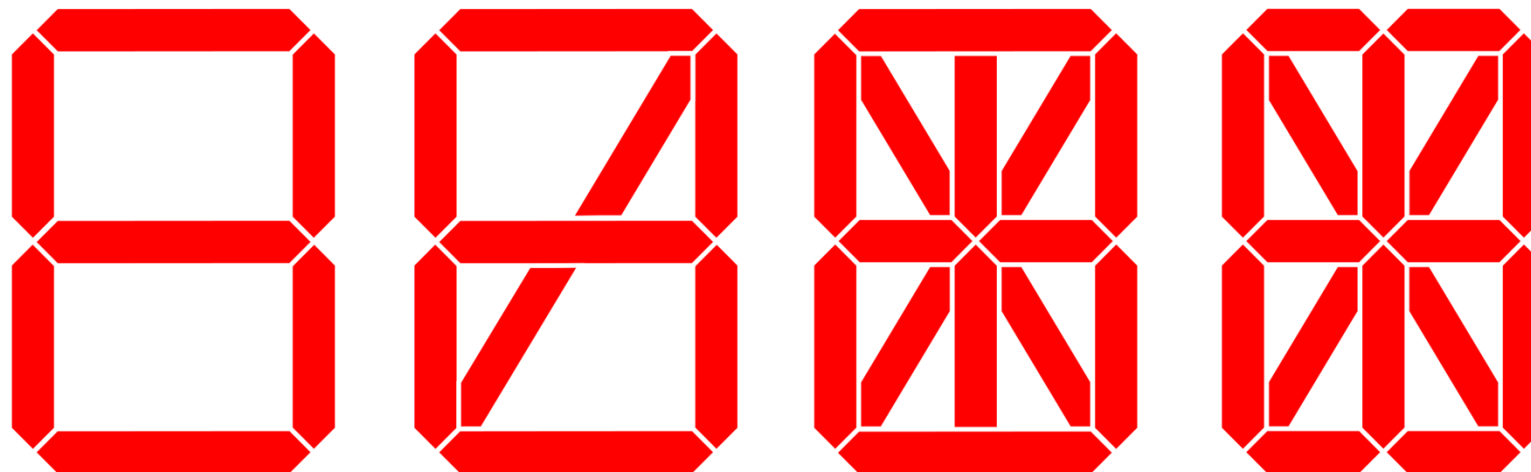




# **LED alfanumerički displej**

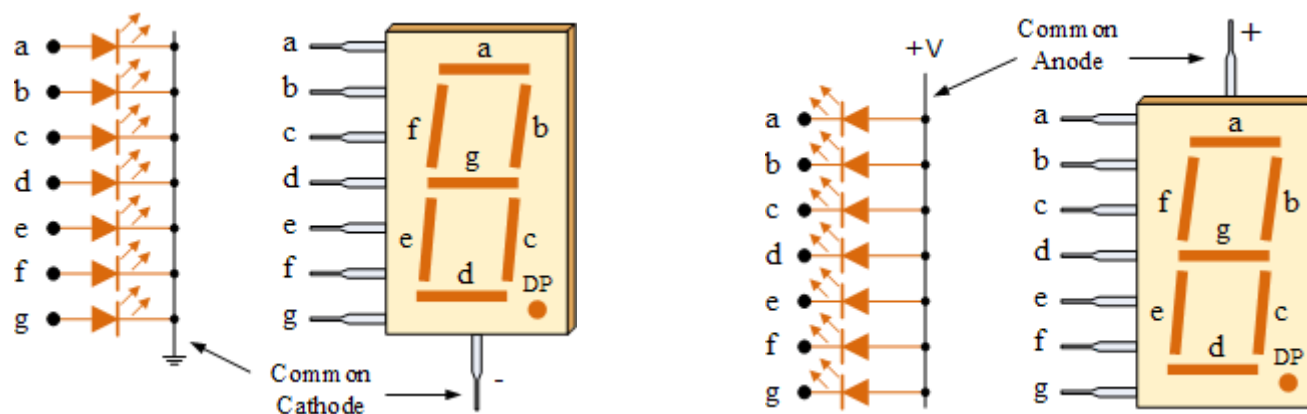
# Tipovi segmentnih displeja

- 7-segmentni
- 9-segmentni
- 14-segmentni
- 16-segmentni



# Tipovi 7-segmentnog displeja

- Sa zajedničkom katodom (Common Cathode Display – CCD) – sve katode povezane su na uzemljenje, segment svetli kada se dovede logička „1“ (visoki napon) na odgovarajuću anodu (pin) LED diode
- Sa zajedničkom anodom (Common Anode Display – CAD) – sve anode povezane su na logičku „1“, a svaki pojedinačni segment se osvetljava kada se na odgovarajuću katodu (pin) dovede „0“ (niski napon).



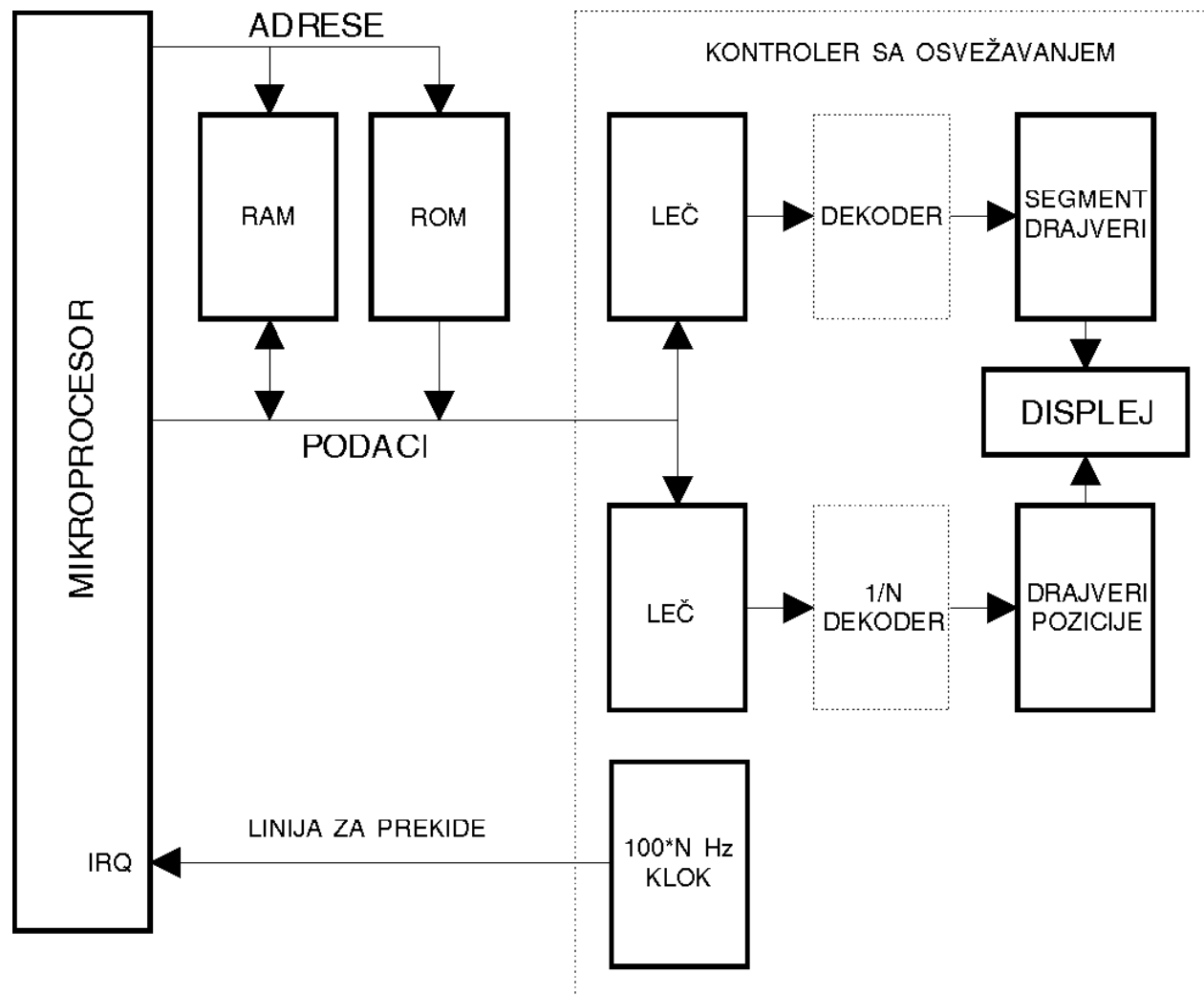


# Tehnike povezivanja LED alfanumeričkog displeja

- Kontroler sa **osvežavanjem** (Refresh controller)
- Kontroler sa **dekodiranim podacima** (Decoded data controller)
- Kontroler sa **kodiranim podacima** (Coded data controller)
- Kontroler sa **pomoćnim procesorom** (Display processor controller)



# Kontroler sa osvežavanjem (Refresh controller)

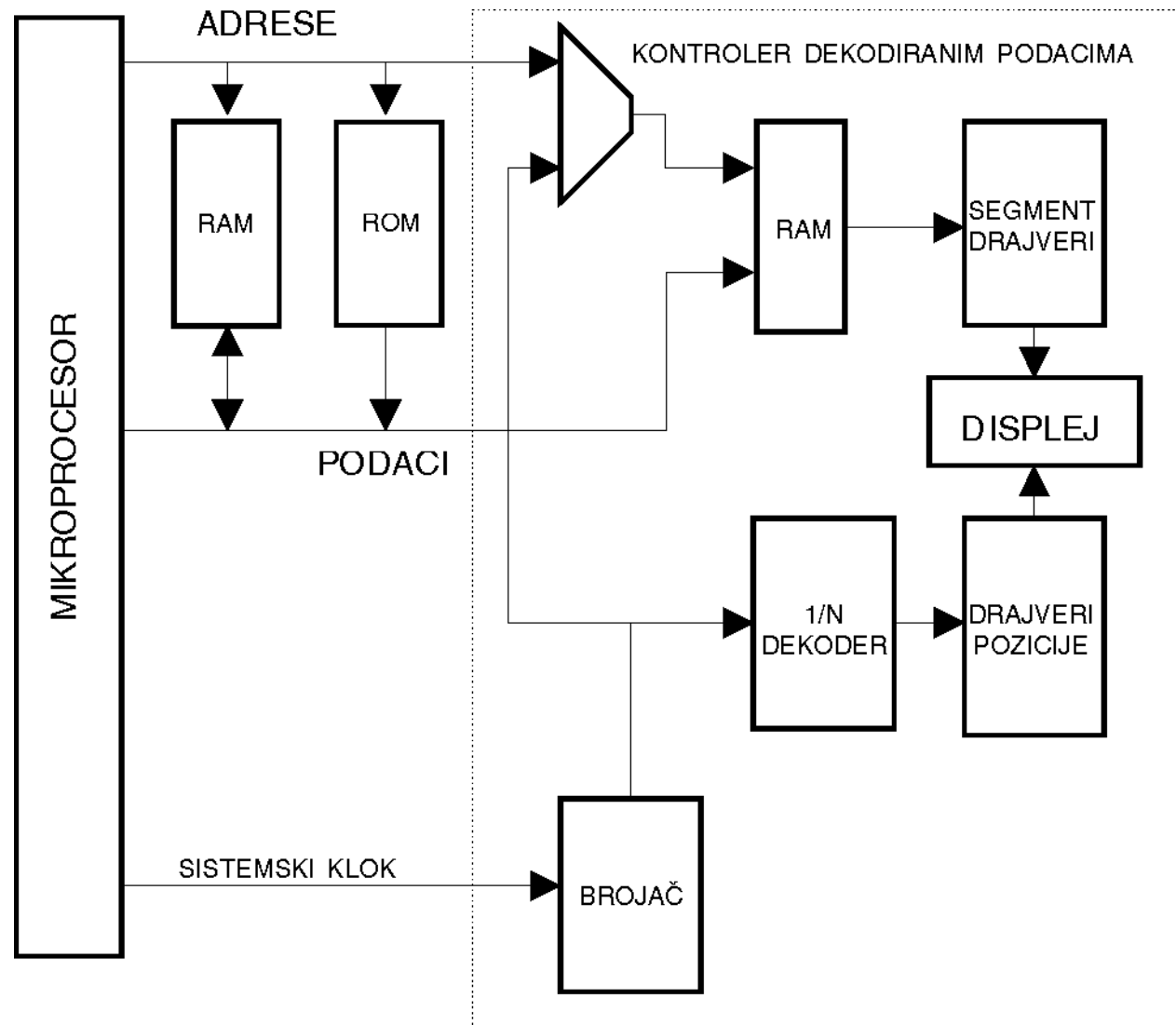




# Kontroler sa osvežavanjem (Refresh controller)

- Kontroler periodično proizvodi prekide i nakon svakog prekida mikroprocesor preduzima aktivnosti za prikazivanje sledećeg znaka na displeju.
- Dekodiranje se može vršiti hardveski ili softverski (zato je na slici dekodirer nacrtan isprekidanom linijom).
- Obezbeđuje najmanju cenu hardvera za bilo koju veličinu (dužinu) displeja.
- ASCII dekodirer može biti u kolima za spregu ili u memoriji.
- Trenutak osvežavanja određen je pojavom prekida, čija je učestanost proporcionalna dužini displeja  $N$  (uobičajeno je  $100 \cdot N$  Hz).
- Zauzetost procesora displejom zavisi od dužine displeja.

# Kontroler sa dekodiranim podacima (Decoded data controller)

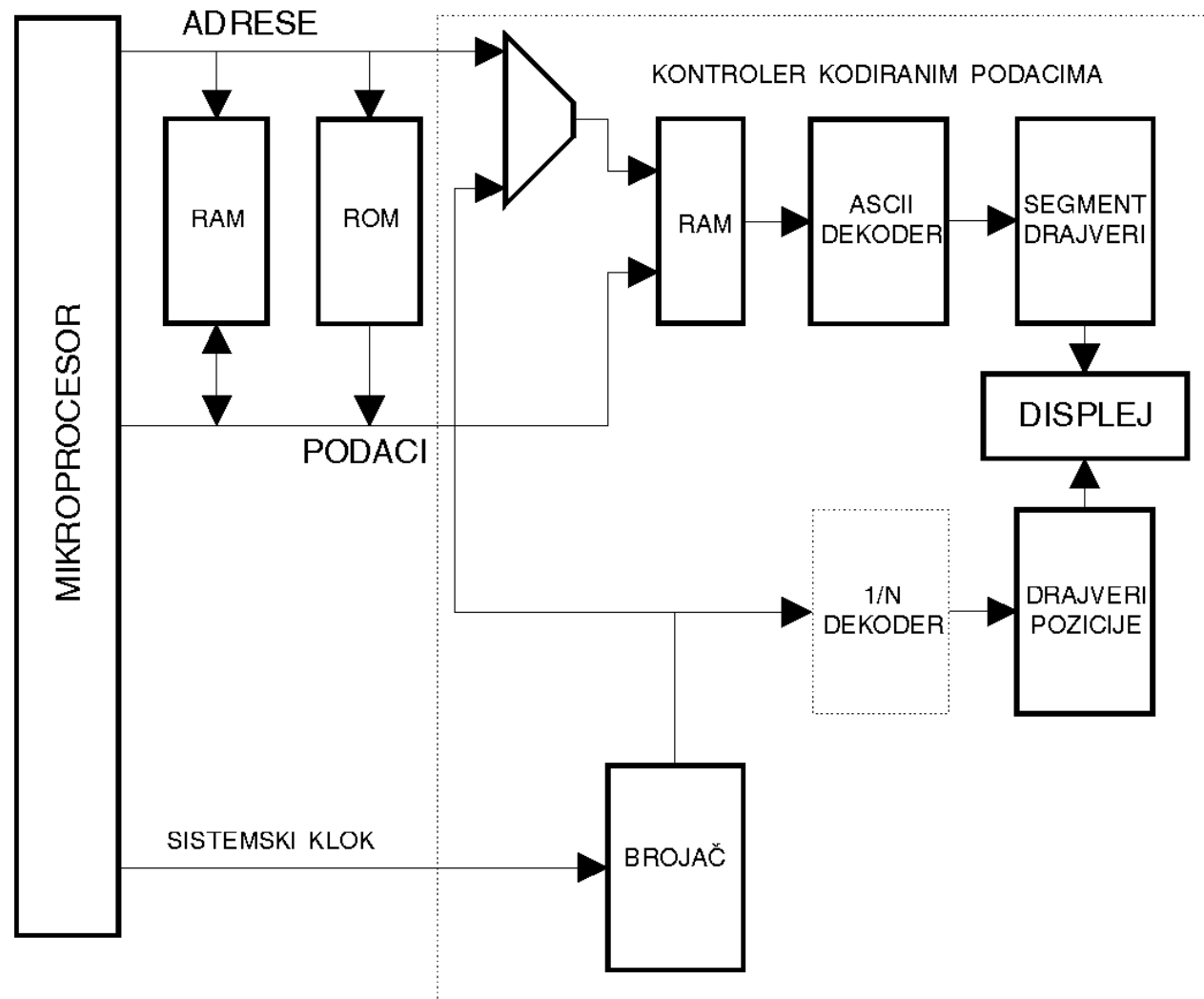




# Kontroler sa dekodiranim podacima (Decoded data controller)

- Osvežavanje se vrši nezavisno od mikroprocesora.
- Lokalni RAM pamti dekodirane podatke
- Kontroler ih kontinuirano ciklično čita i njima osvežav displej.
- Mikroprocesor upisuje dekodirane podatke u RAM samo kada je potrebno izmeniti sadržaj koji se prikazuje na displeju.
- Mikroprocesor može da menja pojedinačne podatke na displeju upisom odgovarajućeg podatka u lokalni RAM.

# Kontroler sa kodiranim podacima (Coded data controller)



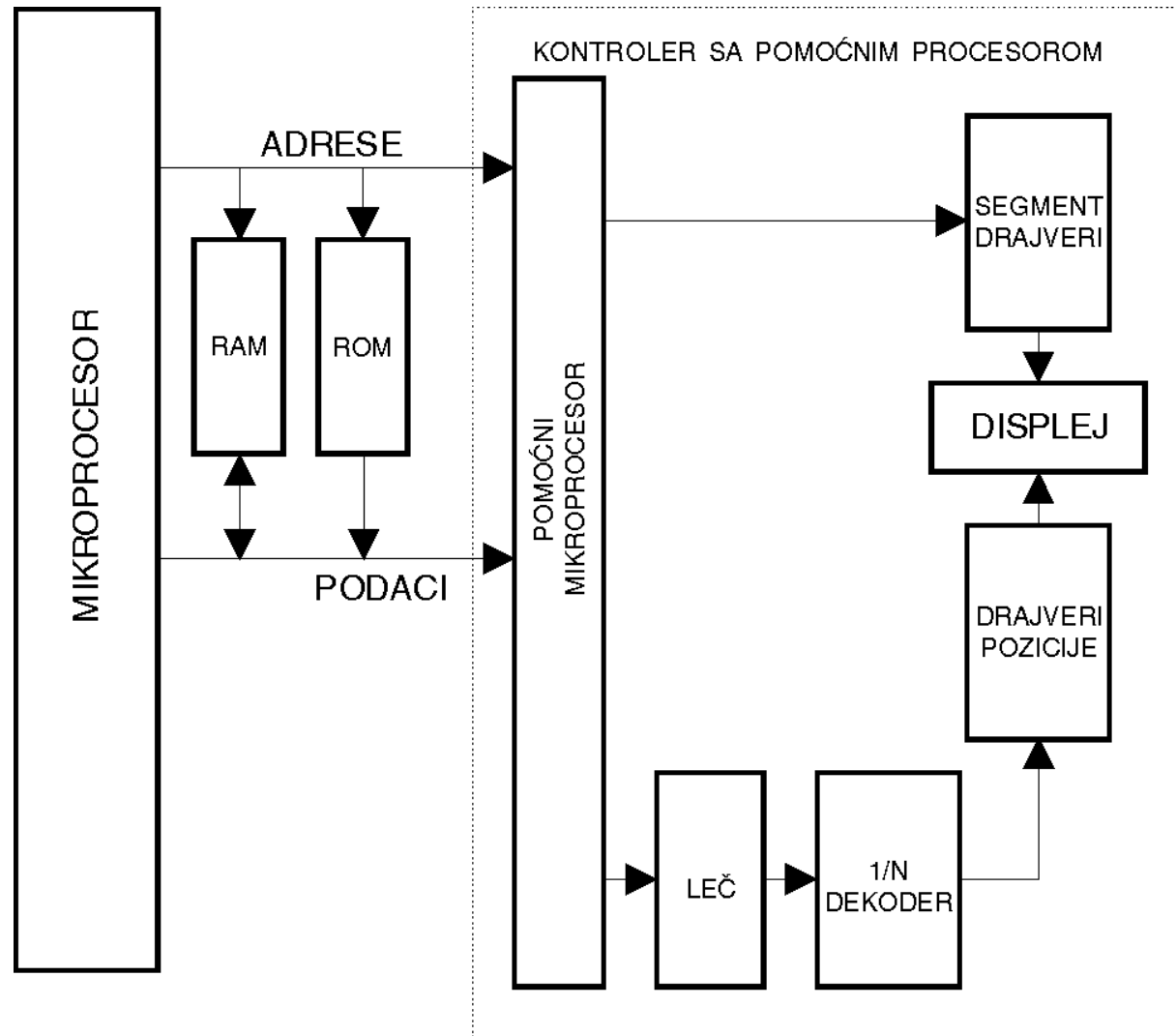


# Kontroler sa kodiranim podacima (Coded data controller)

- I ovaj kontroler osvežava displej nezavisno od mikroprocesora.
- Lokalni RAM sadrži ASCII podatke koje je potrebno prikazati na displeju.
- Znake kontroler čita, dekodira i prikazuje na displeju.
- Mikroprocesor upisuje ASCII kod znaka koji je potrebno prikazati samo kada je potrebno izmeniti informacije na displeju.
- Obzirom da je ASCII kod mnogo kompaktniji od podatka koji se prikazuje, kontroler zahteva manje lokalnog RAM-a.
- Mikroprocesor može da menja pojedinačne podatke na displeju upisom odgovarajućeg podatka u lokalni RAM.



# Kontroler sa pomoćnim procesorom (Display processor controller)





# Kontroler sa pomoćnim procesorom (Display processor controller)

- Koristi se poseban mikroprocesor za upravljanje radom displeja.
- Zadatak glavnog mikroprocesora je da u zajednički RAM upisuje ASCII kodove znakova koje je potrebno prikazati na displeju.
- Zahteva interakciju samo kada se menja podatak koji koji se prikazuje na displeju.
- Posebni procesor obezbeđuje dodatne mogućnosti kao što su: različiti načini ulaza podataka (levi i desni), tepćući kursor, neke editorske komande, kao i izlaz podataka.



# Dekodiranje ASCII znakova

- Hardversko
  - brže
  - manje zauzeće mikropcesora
  - fiksni set karaktera
- Softversko
  - sporije
  - veće zauzeće mikropcesora
  - vrlo fleksibilno (različita pisma, grafički simboli) i potpuno pod kontrolom programa