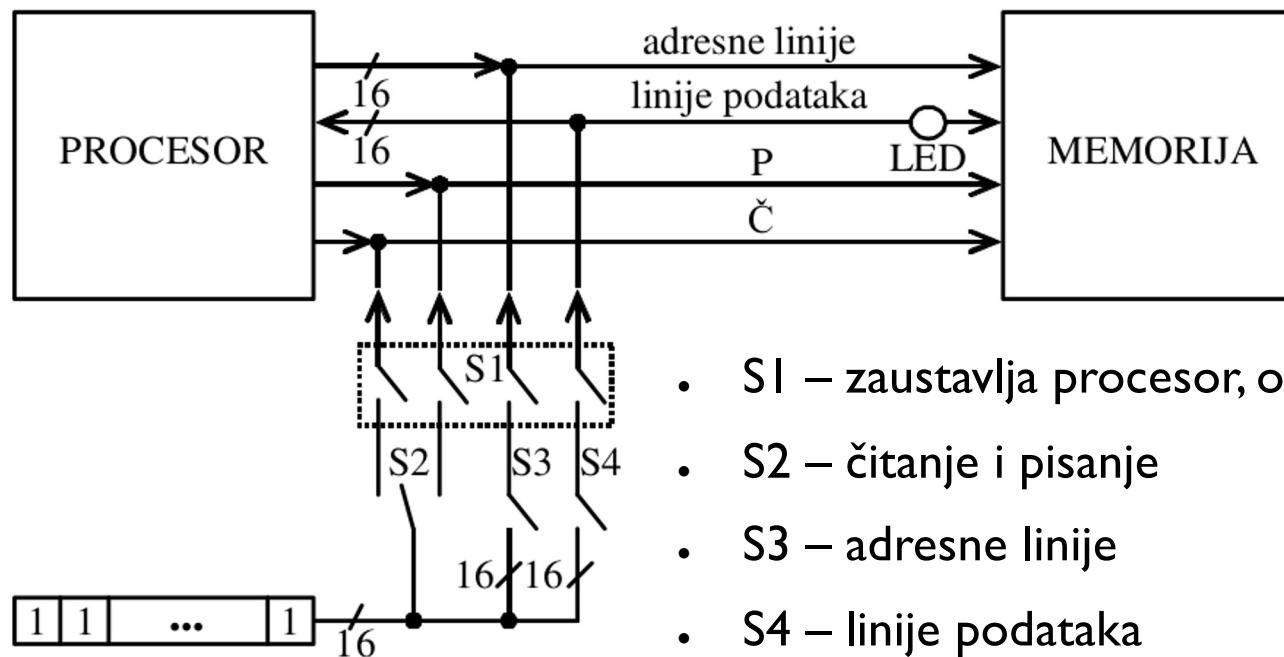
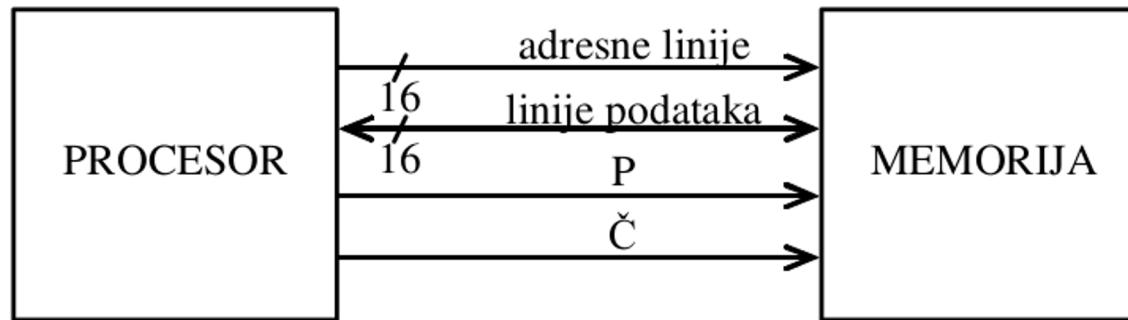


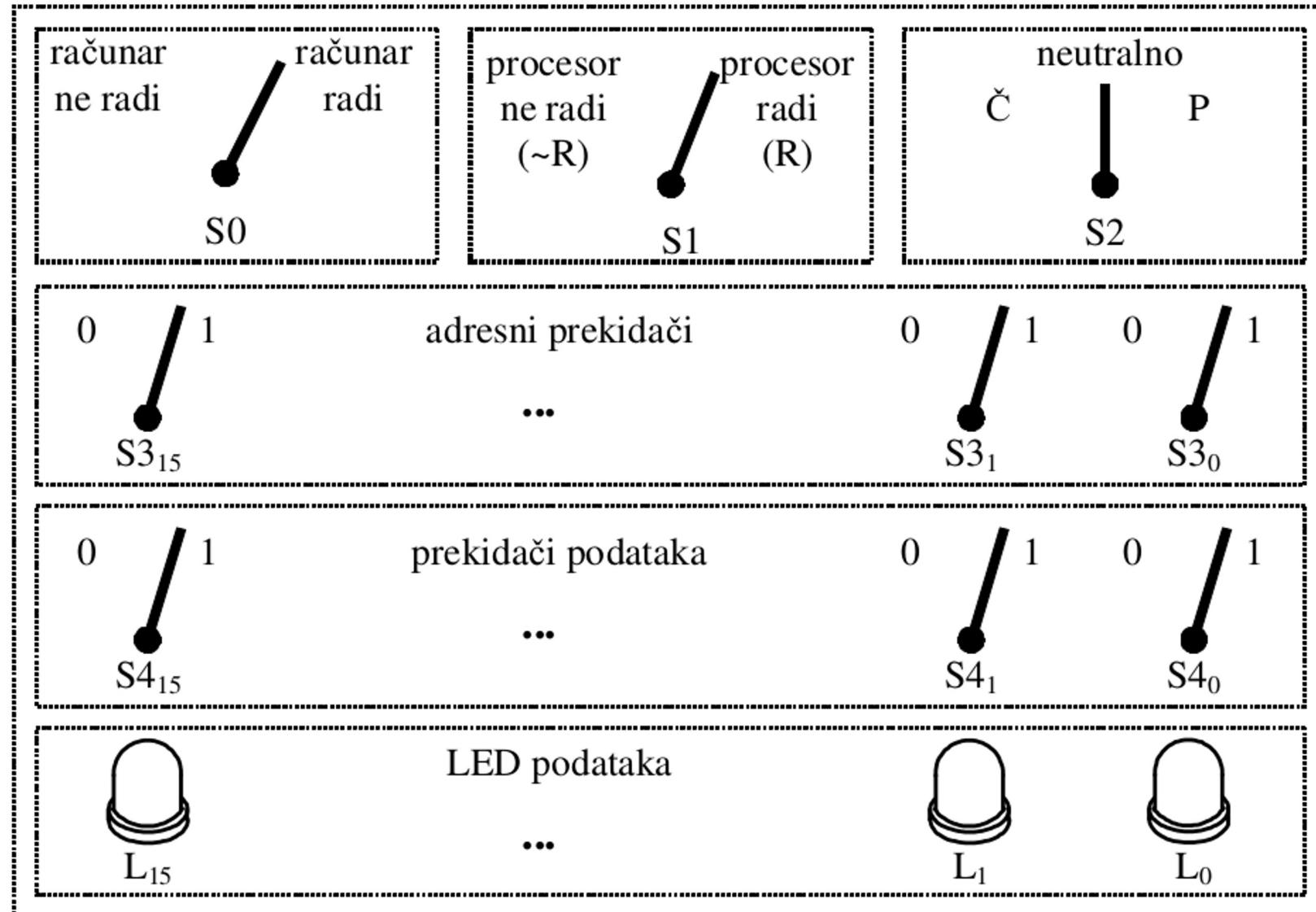
# Računar Koncept

# Računar Koncept

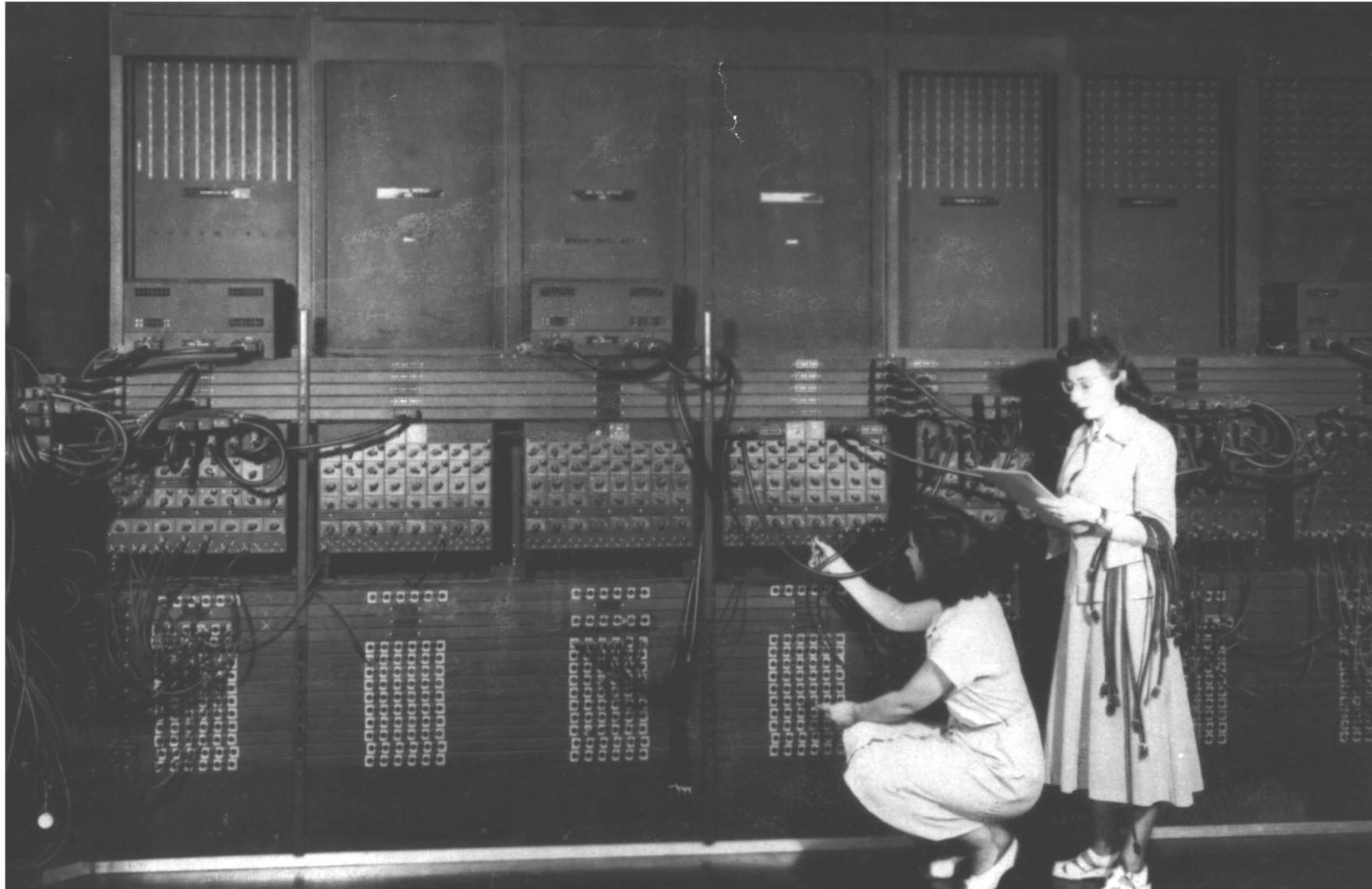


- S1 – zaustavlja procesor, omogućava pristup spolja
- S2 – čitanje i pisanje
- S3 – adresne linije
- S4 – linije podataka

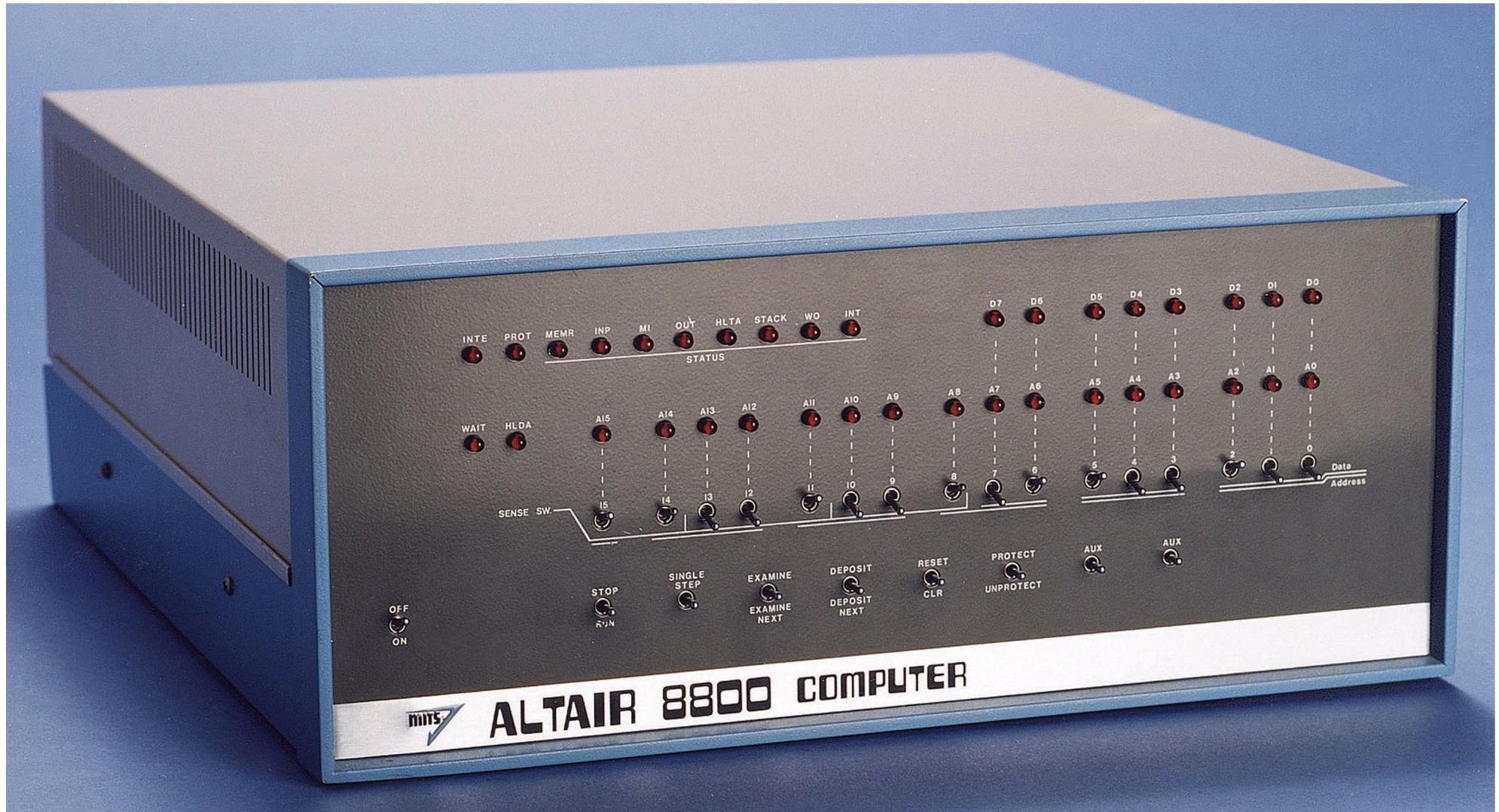
# Upravljačka tabla



# ENIAC, 1946.



# Altair 8800, 1974.



# Ulaz i izlaz – komandni jezik

P – pisanje memorijske lokacije

Č – čitanje memorijske lokacije

I – izvršavanje programa

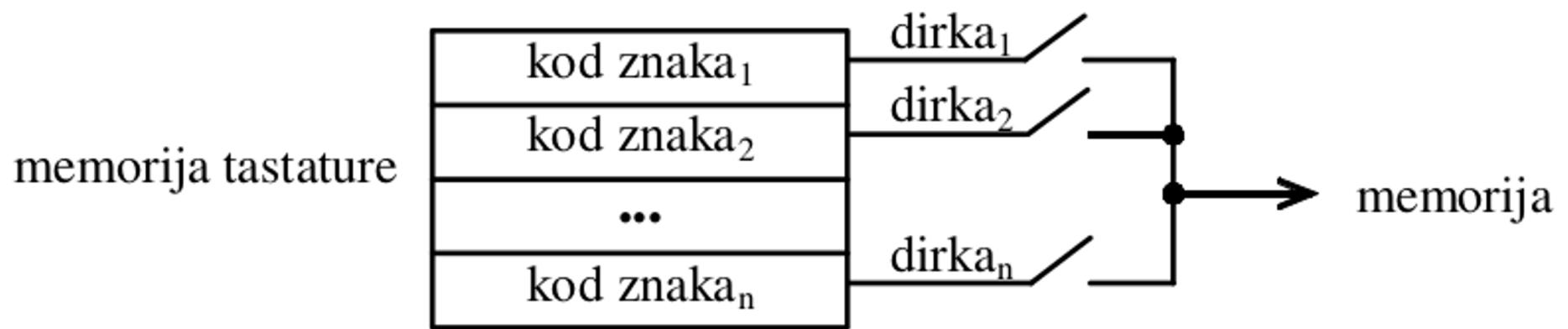
komanda -> P razmak broj razmak broj

| Č razmak broj

| I razmak broj

# Tastatura

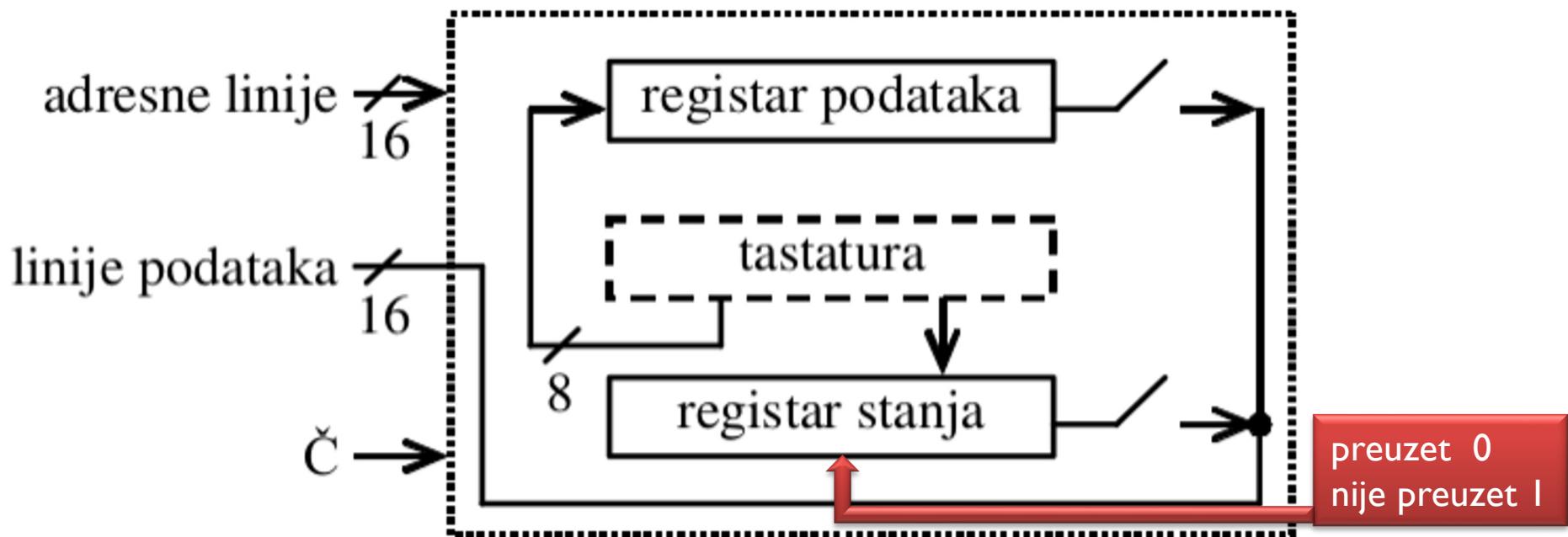
Kodovi znakova se nalaze u posebnoj memoriji tastature



# Tastatura

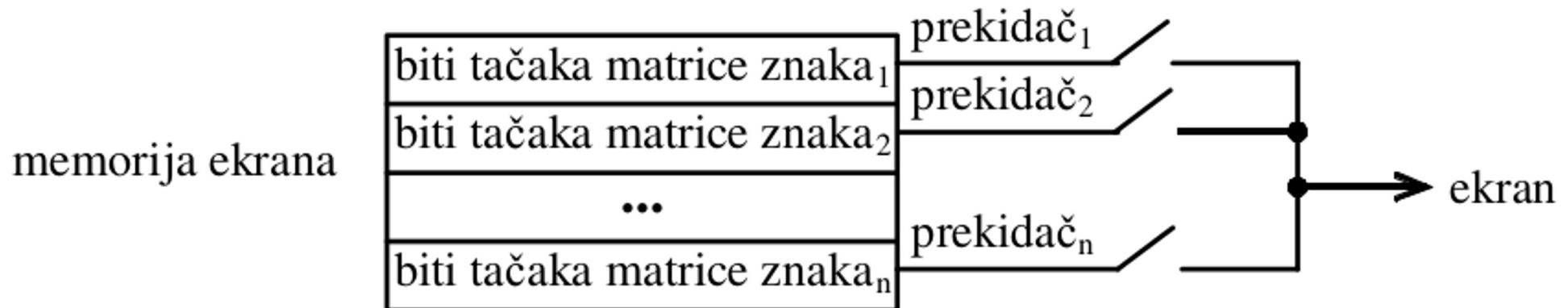
## Kontroler tastature

- registar podataka
- registar stanja



# Ekran

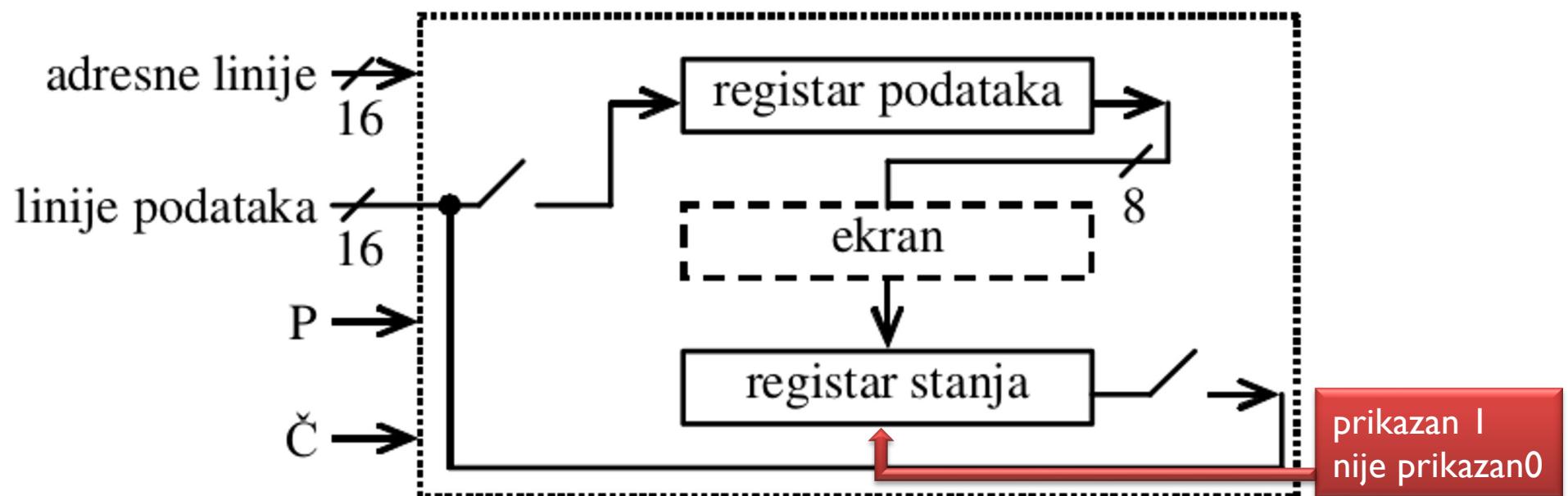
- Realizovan kao matrica, gde svaka pozicija može prikazati jedan znak
- Kod znaka određuje koje tačke u znakovnoj poziciji treba da budu osvetljene



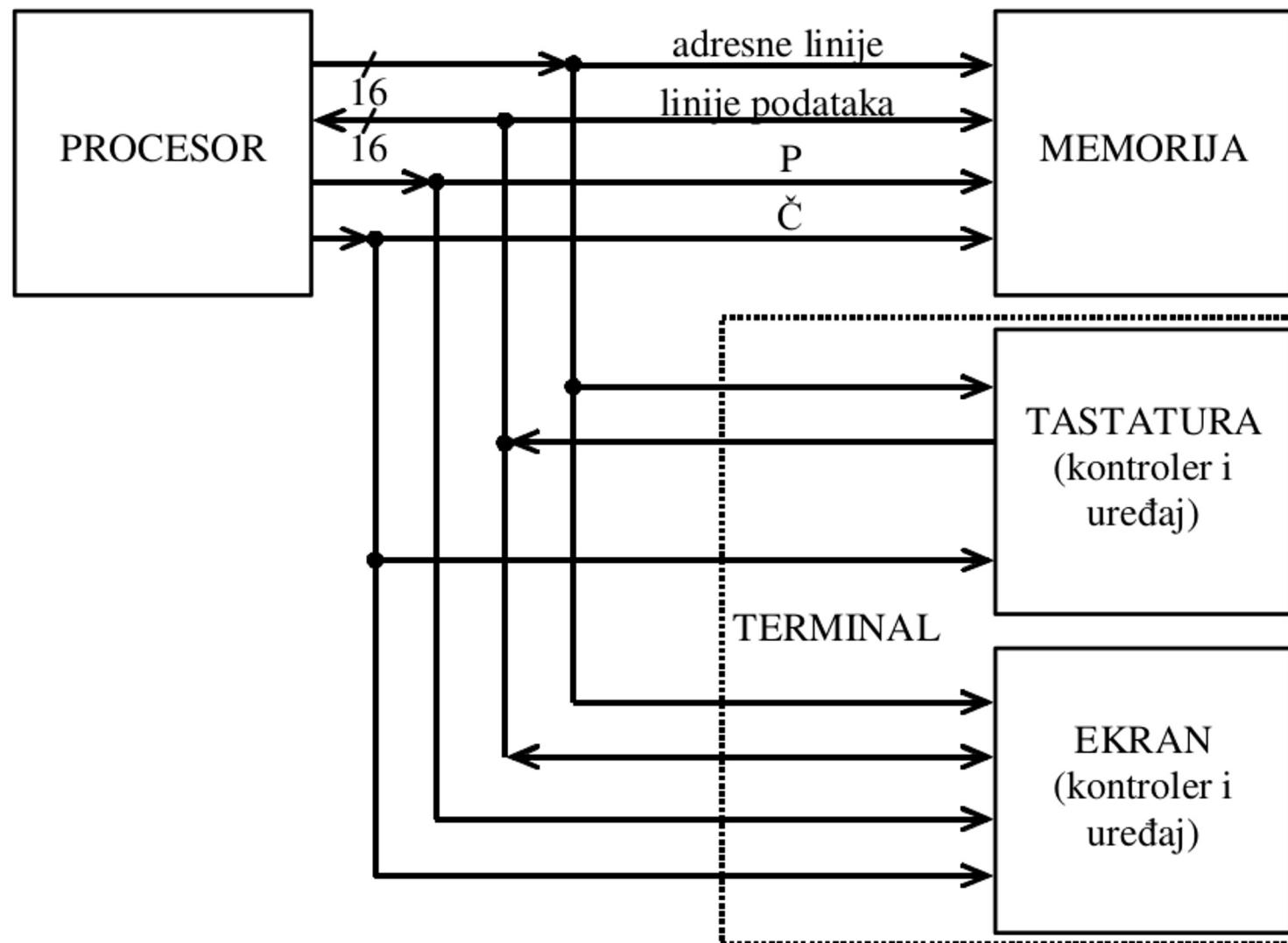
# Ekran

## Kontroler ekrana

- registar podatka
- registar stanja



# Računar Koncept sa terminalom



# DEC VT05, 1970.



# Kodiranje znakova

ASCII (engl. *American Standard Code for Information Interchange*) kod, 1963.

- 7-bitni
- 8-bitni, kodne strane

UNICODE, 1988.

- 16-bitni
- 32-bitni
- UTF-8
- UTF-16
- UTF-32



Kod	Znak	Kod	Znak	Kod	Znak	Kod	Znak
000	<i>NUL (Null character)</i>	020	<i>SP (Space)</i>	040	@	060	'
001	<i>SOH (Start of Header)</i>	021	!	041	A	061	a
002	<i>STX (Start of Text)</i>	022	"	042	B	062	b
003	<i>ETX (End of Text)</i>	023	#	043	C	063	c
004	<i>EOT (End of Transmission)</i>	024	\$	044	D	064	d
005	<i>ENQ (Enquiry)</i>	025	%	045	E	065	e
006	<i>ACK (Acknowledgment)</i>	026	&	046	F	066	f
007	<i>BEL (Bell)</i>	027	'	047	G	067	g
008	<i>BS (Backspace)</i>	028	(	048	H	068	h
009	<i>HT (Horizontal Tab)</i>	029	)	049	I	069	i
00A	<i>LF (Line Feed)</i>	02A	*	04A	J	06A	j
00B	<i>VT (Vertical Tab)</i>	02B	+	04B	K	06B	k
00C	<i>FF (Form Feed)</i>	02C	,	04C	L	06C	l
00D	<i>CR (Carriage Return)</i>	02D	-	04D	M	06D	m
00E	<i>SO (Shift Out)</i>	02E	.	04E	N	06E	n
00F	<i>SI (Shift In)</i>	02F	/	04F	O	06F	o
010	<i>DLE (Data Link Escape)</i>	030	0	050	P	070	p
011	<i>DC1 (XON/Device Control 1)</i>	031	1	051	Q	071	q
012	<i>DC2 (Device Control 2)</i>	032	2	052	R	072	r
013	<i>DC3 (XOFF/Device Control 3)</i>	033	3	053	S	073	s
014	<i>DC4 (Device Control 4)</i>	034	4	054	T	074	t
015	<i>NAK (Negative Ack)</i>	035	5	055	U	075	u
016	<i>SYN (Synchronous Idle)</i>	036	6	056	V	076	v
017	<i>ETB (End of Trans. Block)</i>	037	7	057	W	077	w
018	<i>CAN (Cancel)</i>	038	8	058	X	078	x
019	<i>EM (End of Medium)</i>	039	9	059	Y	079	y
01A	<i>SUB (Substitute)</i>	03A	:	05A	Z	07A	z
01B	<i>ESC (Escape)</i>	03B	;	05B	[	07B	{
01C	<i>FS (File Separator)</i>	03C	<	05C	\	07C	
01D	<i>GS (Group Separator)</i>	03D	=	05D	]	07D	}
01E	<i>RS (Record Separator)</i>	03E	>	05E	^	07E	~
01F	<i>US (Unit Separator)</i>	03F	?	05F	-	07F	<i>DEL (Delete)</i>

# Znakovna i interna predstava celih brojeva

$1000000_2$

- Interno:  $41_{16} = 65_{10}$
- Znakovno:  $31_{16} 30_{16} 30_{16} 30_{16} 30_{16} 30_{16} 30_{16} 31_{16}$

## Znakovna predstava – ulaz i izlaz

- konverzija iz znakovne predstave u internu
- konverzija iz interne predstave u znakovnu

# Znakovna interakcija

## Unos komandi znak po znak

- echo – prikaz znakova na ekranu
- editovanje (unos i brisanje znakova)
- interpretiranje komandi
  - prepoznavanje komande
  - izdvajanje argumenata
  - izvršavanje komande
- interpreter komandi
  - prompt – znak koji saopštava korisniku da je interpreter spremam

# Rad interpretera

*while (true)*

*prikaži prompt i preuzmi komandu*

*if (prvi znak komande je P) && (sledi heksadecimalni broj)  
    && (sledi heksadecimalni broj)*

*izmeni sadržaj navedene memorijske lokacije*

*else*

*if (prvi znak komande je Č) && (sledi heksadecimalni broj)*

*prikaži sadržaj navedene memorijske lokacije*

*else*

*if (prvi znak komande je I) && (sledi heksadecimalni broj)*

*pokreni izvršavanje korisničkog programa*

*else*

*prikaži poruku greške*

- Izvršavanje korisničkih programa – kao **potprogram**

# Drajver terminala

## Ulazna operacija

- preuzimanje znakova

čekaj\_tastaturu:

PREBACI_NR	\$0,%0
PREBACI_DR	registar_stanja_tastature,%1
UPOREDI	%1,%0
SKOČI_ZA_==	čekaj_tastaturu
PREBACI_DR	registar_podataka_tastature,%2

- echo

- editovanje

## Izlazna operacija

- prikaz znakova

čekaj\_ekran:

PREBACI_NR	\$0,%0
PREBACI_DR	registar_stanja_ekrana,%1
UPOREDI	%1,%0
SKOČI_ZA_==	čekaj_ekran
PREBACI_RD	%2,registar_podataka_ekrana

# Drajver terminala

Radno čekanje (engl. *busy waiting, polling*)

## Ulaz

- dok se ne napuni bafer, ili
  - se pritisne taster za kraj unosa
- poništavanje znaka zahteva
  - uklanjanje znaka iz bafera
  - uklanjanje znaka sa ekrana

## Izlaz

- izlazni bafer

# BIOS

*Basic Input Output System (CP/M, 1974.)*

- UEFI (engl. *Unified Extensible Firmware Interface*), 2005.
- Coreboot 2000/2008.
  - Chromebook

Kod KONCEPT-a, BIOS čine

- interpreter komandi
- drajver terminala
- ulazna lokacija ( $0000_{16}$ ) sadrži adresu početka BIOS-a, tačnije interpretera komandi

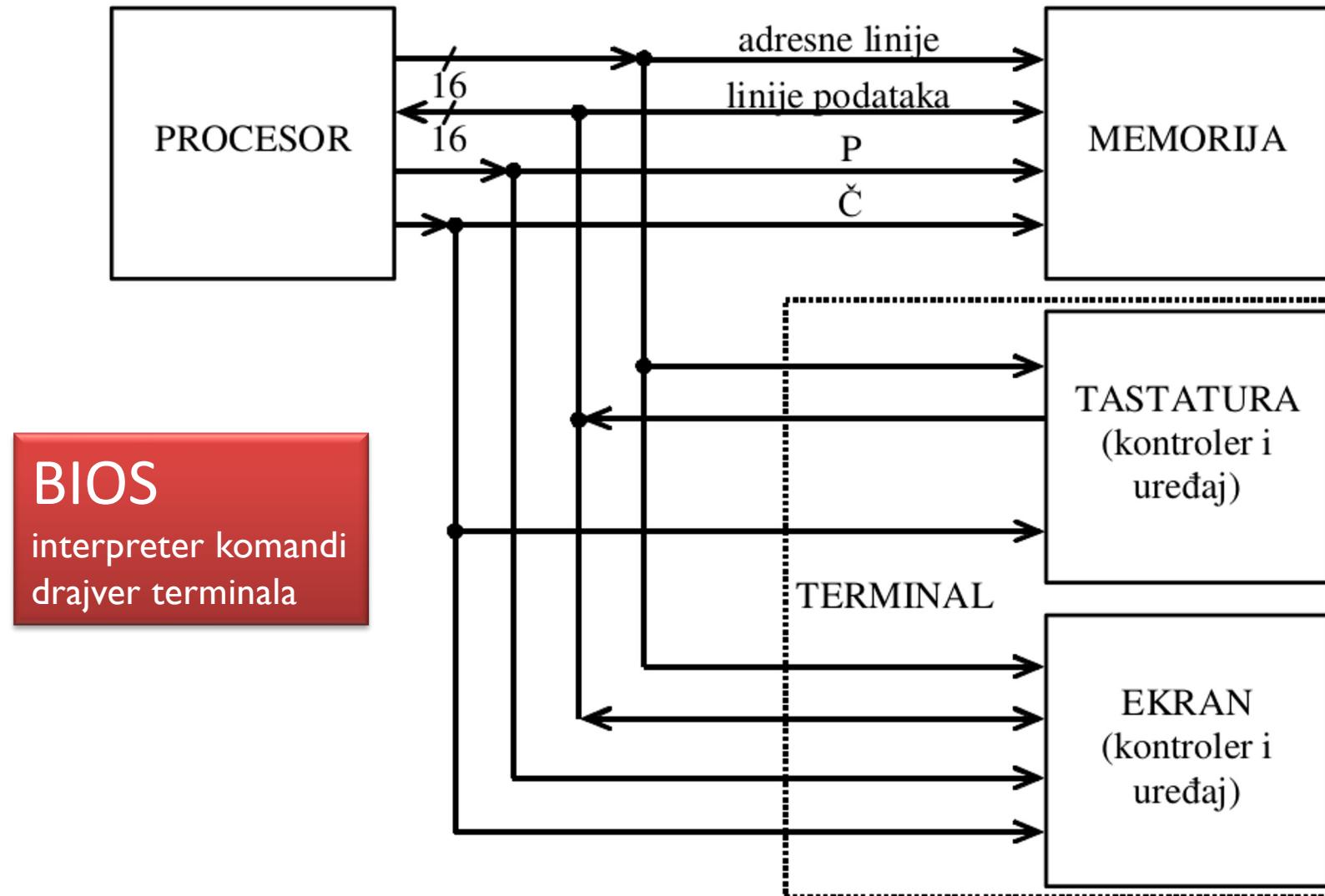
Interaktivni i programski nivo korišćenja BIOS-a

Računar Koncept sa terminalom i BIOS-om ima samo 2 prekidača (napajanje, reset)



# Vrste memorije

# Računar Koncept sa terminalom



# Vrste memorije

## Poluprovodničke

- Narušive (engl. *volatile*) – RAM (Random Access Memory)
  - SRAM (tipično za keš), DRAM, DDR, GDDR...
- Nenarušive (engl. *non-volatile*) – ROM (Read-Only Memory)
  - PROM, EPROM, EEPROM, FLASH

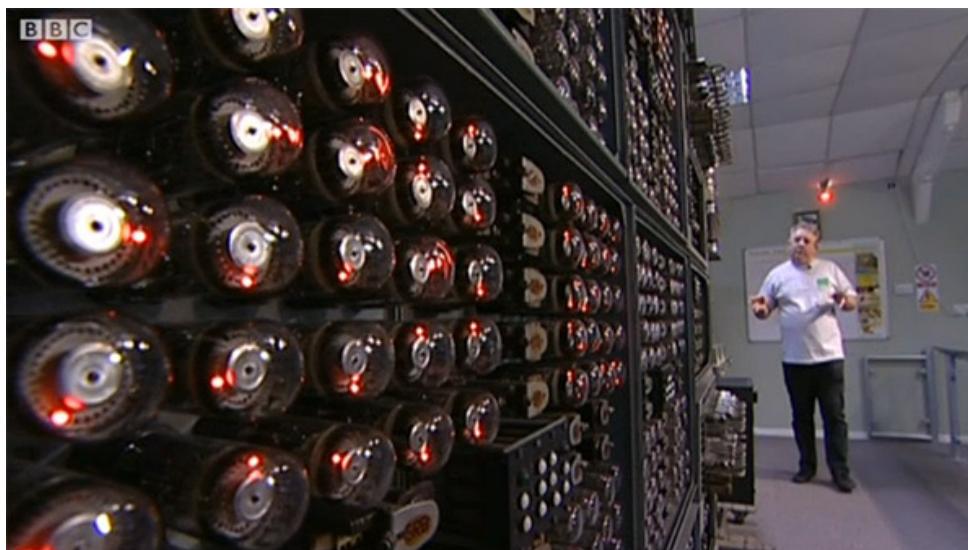
## Magnetne

- Hard disk
- Floppy disk

# Williams-ova cev, 1946

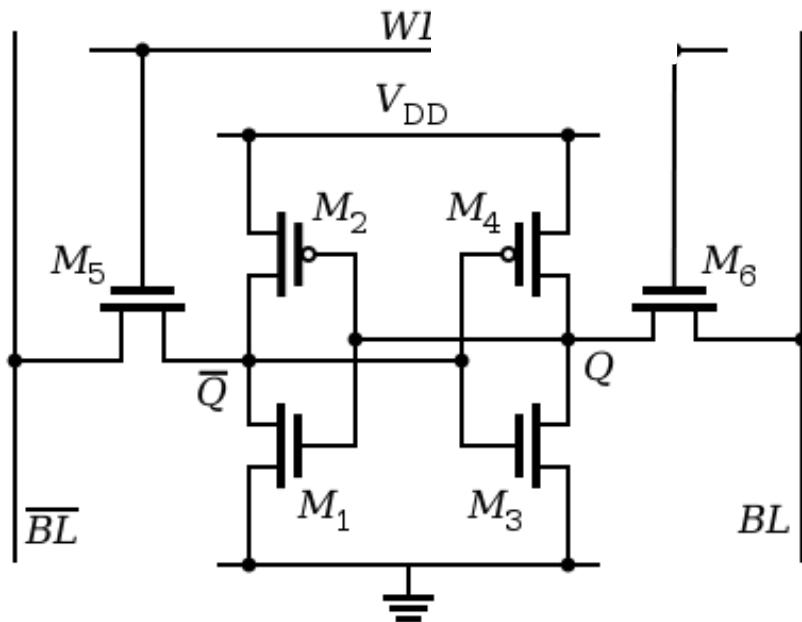


# Dekatron cev, WITCH, 1949

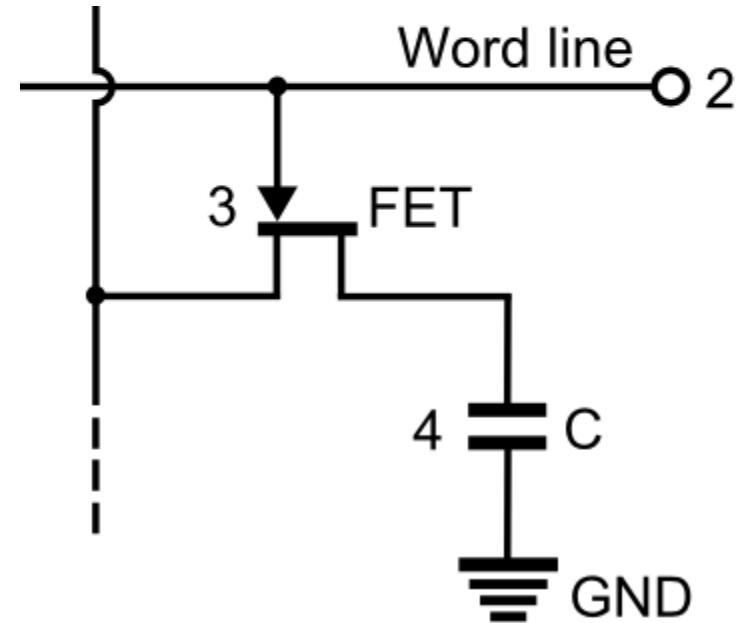


# SRAM i DRAM

SRAM ćelija (6 tranzistora)



DRAM ćelija  
(1 tranzistor i 1 kondenzator)



# Hard disk

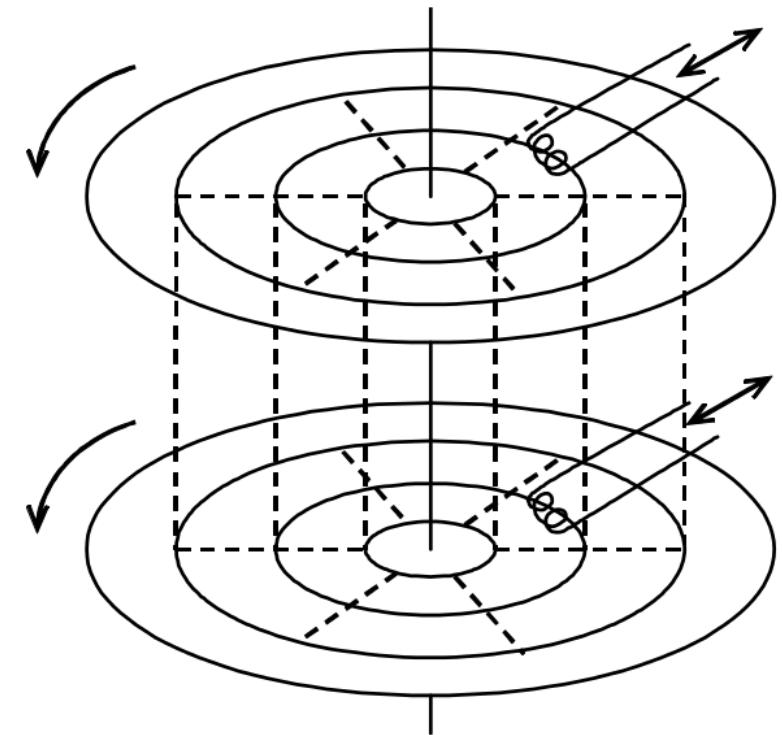
## Jedna ili više ploča

- više koncentričnih **staza**
- staze se dele na **sektore**

## Glava za čitanje i pisanje

## Rad sa diskom

- jedinica čitanja i pisanja je **blok** sa jednim ili više sektora
- bafer za bar 1 blok



# Poređenje memorija

## Hard disk

- veliki kapacitet
- mala brzina
- trajno pamćenje

## RAM

- manji kapacitet
- velika brzina
- pamćenje dok ima napona

## Radna (RAM) i masovna (HD) memorija

# Hijerarhija memorije



Izvor: <http://computerscience.chemeketa.edu/cs160Reader/ComputerArchitecture/MemoryHeirarchy.html>

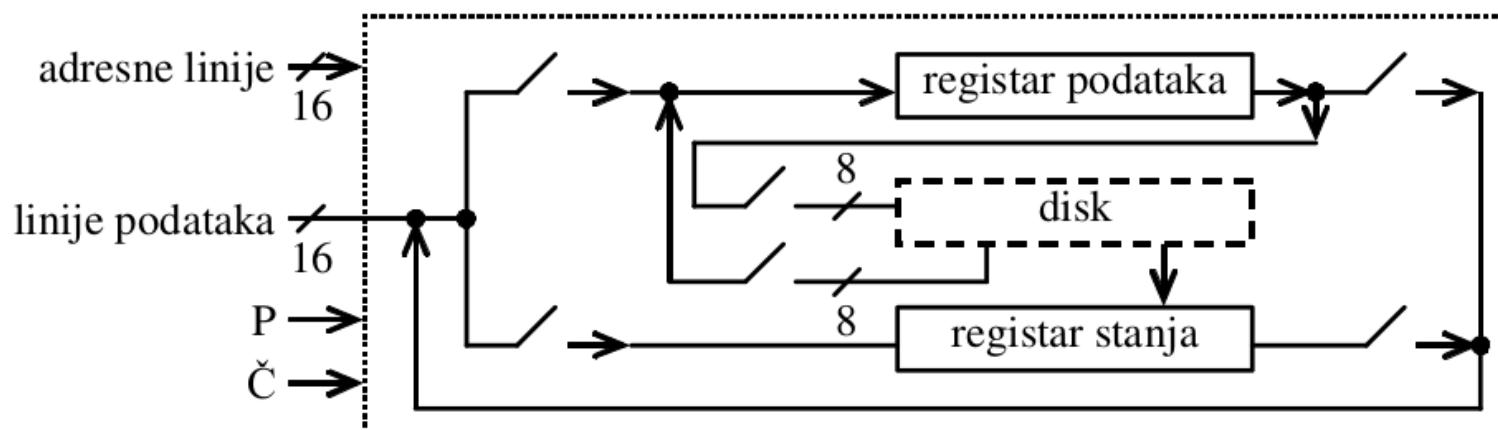
# Kontroler diska

Prenos bajta ka disku i od diska

Prenos broja staze i broja sektora

Registar stanja

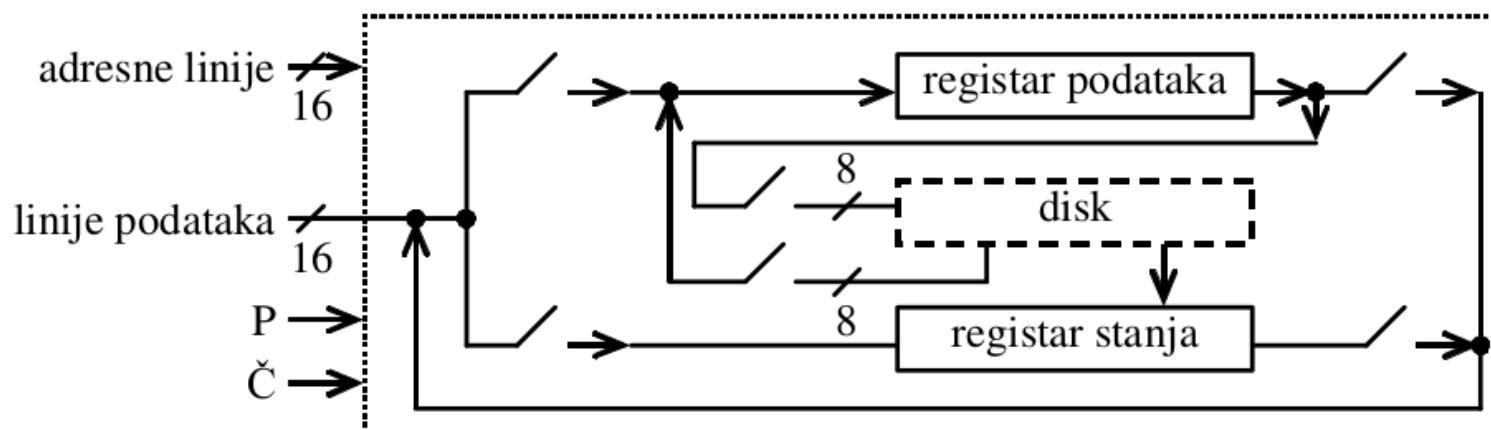
- bit 0 – komunikacija je moguća
- bit 1 i 2 – smer slanja podatka (na disk/sa diska)



# Kontroler diska

Postupak:

1. Procesor postavlja smer prenosa
2. Prenos broja staze
3. Prenos broja sektora
4. Prenos bloka bajt po bajt



# Drajver diska

**Drajver diska** brine da pristup registrima kontrolera diska bude u skladu sa pravilima upotrebe

Pretvara redni broj bloka u broj staze i sektora

Ima 2 potprograma:

- **ulazna operacija** (čitanje sadžaja bloka – adresa ulaznog bafera programa i broj bloka)
- **izlazna operacija** (pisanje sadžaja bloka – adresa izlaznog bafera programa i broj bloka)

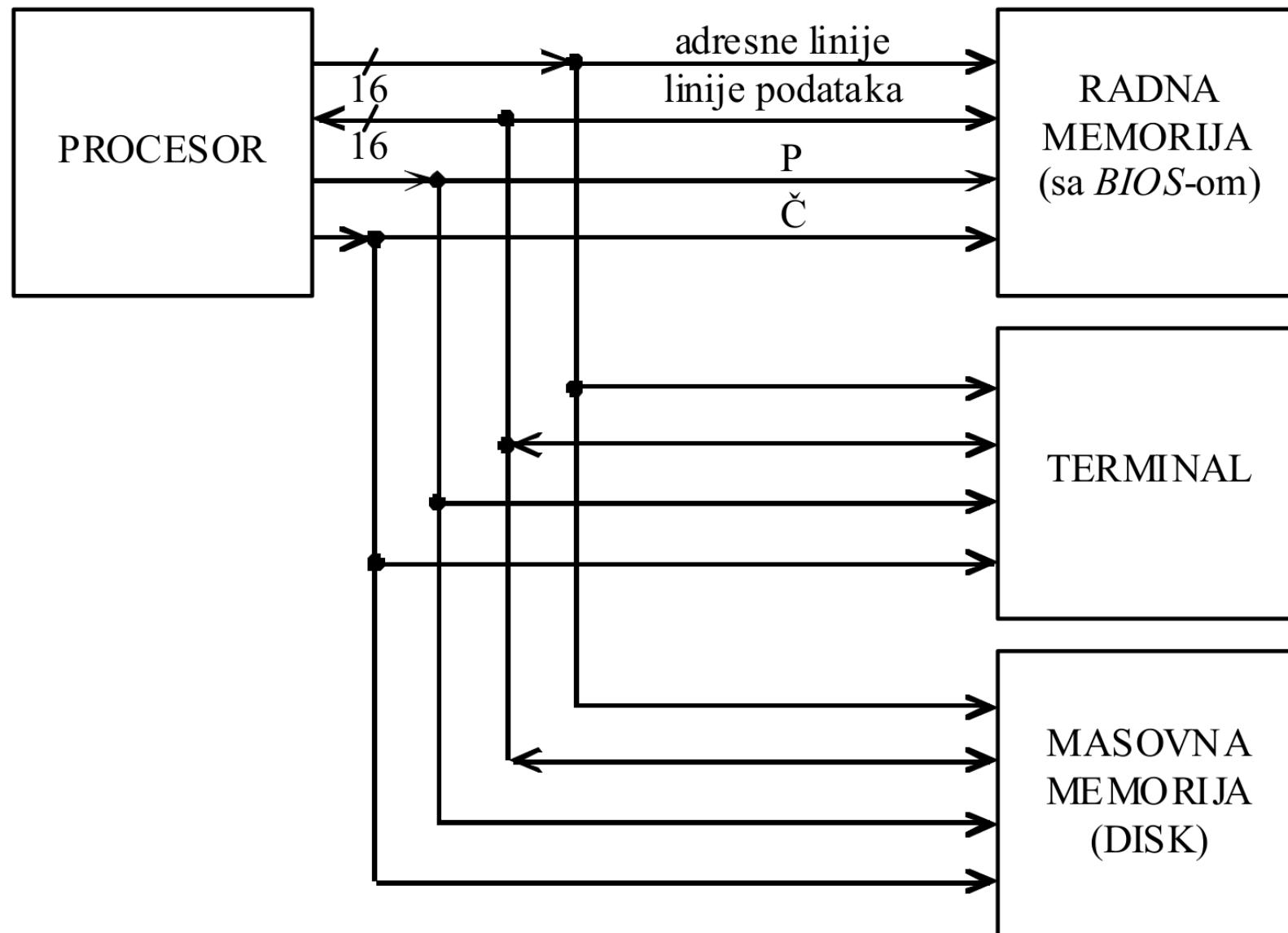
# Drajver diska

Prošireni interpreter i BIOS

**komanda -> P razmak broj razmak broj**  
|Č razmak broj  
|I razmak broj  
|N razmak broj razmak broj  
|S razmak broj razmak broj

prošireni interpreter komandi	
drayver terminala	drayver diska

# Računar Koncept sa masovnom memorijom



# (Ne tako) nove tehnologije

## **MRAM** (engl. *Magnetoresistive Random-Access Memory*)

- 1 bit – dve feromagnetne ploče razdvojene izolatorom
- još uvek nedovoljno razvijena za široku upotrebu
- današnji vodeći operativni sistemi nemaju podršku

## **SSD** (engl. *Solid State Disk*)

- Flash memorija sa hard disk interfejsom
- mnogo brži od magnetnog diska
- još uvek postoje problemi sa ograničenim brojem upisa (nije preporučljivo da se koristiti za swap/pagefile!)
- (Mini) PCIe SSD

# **Kodovi za otkrivanje i popravku grešaka**

# Greške u radu sa memorijom

Istraživanje iz 2010, sistem Jaguar, 360TB ECC RAM (engl. Error-Correcting Code)

- greške u radu sa memorijom: 350 u minuti
- skalirano na računar sa 8 GB memorije:  
otprilike 1 greška na svaka 2 sata

Google istraživanje iz 2009: oko 5 single-bit grešaka na 8 GB memorije na sat

Manji tranzistori – manje energije za pobudu

Pozadinsko zračenje (mahom od kosmičkih zraka)

Izvori: How To Kill A Supercomputer: Dirty Power, Cosmic Rays, and Bad Solder, *IEEE Spectrum*, 2016; Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors, Proc. ISCA 2014.

# Kodovi za otkrivanje i popravku grešaka

Zasnivaju se na **dodavanju (redundantnih) informacija** u podatak

Dodavanjem **bita provere** (engl. *check bit*) podatku se dobija **kodna reč** (engl. *codeword*)

Svi bitovi kodne reči moraju zadovoljiti neko **pravilo**

Skup kodnih reči je podskup svih kombinacija bitova

# Kodovi za otkrivanje i popravku grešaka

Jednostavan način – **bit parnosti** (engl. *parity bit*)

- parna parnost (ukupan broj jedinica je paran)

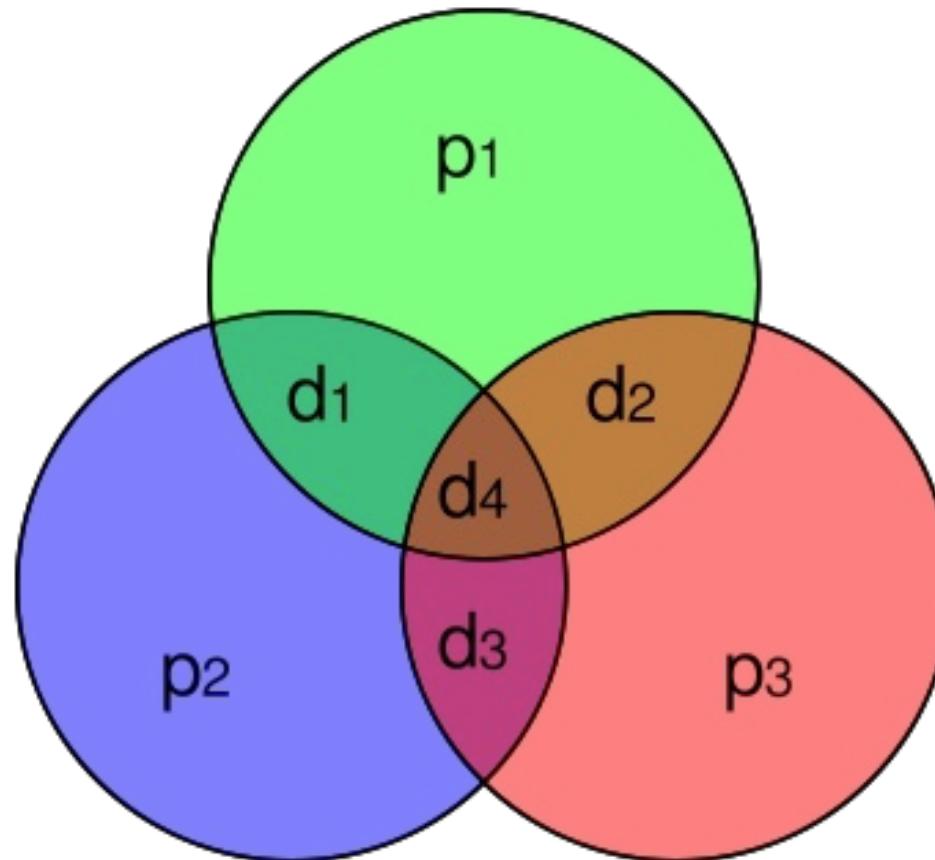
kodne reči	
biti podataka	bit parnosti
00	0
01	1
10	1
11	0

- sve kodne reči se razlikuju u dva bita => može detektovati promenu jednog bita

## Hamingova udaljenost – broj različitih bitova

# Hamingov (7, 4) kod (1950)

Više bita parnosti može omogućiti i korekciju



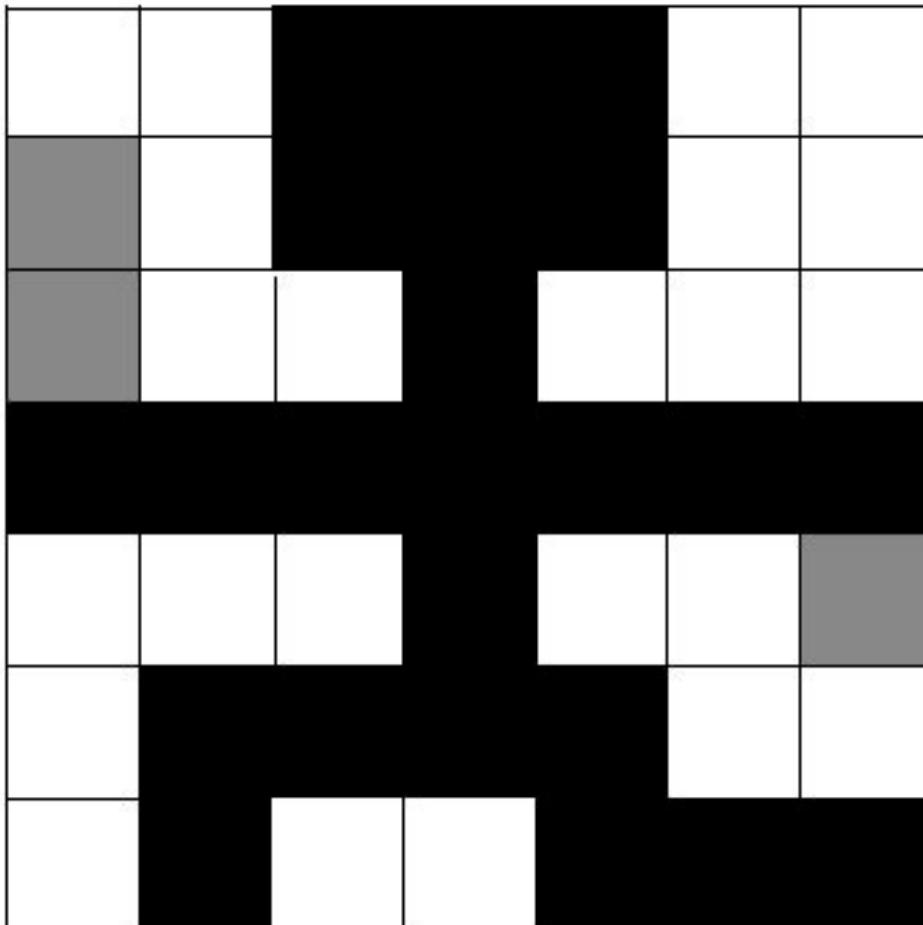
# Hamingov (7, 4) kod (1950)

Više bita parnosti može omogućiti i korekciju

	redni brojevi bita kodne reči						
biti parnosti	1	2		4			
biti podataka			3		5	6	7
podskup 1. bita parnosti	→		+		+		+
podskup 2. bita parnosti		→	+			+	+
podskup 3. bita parnosti				→	+	+	+
primer kodne reči	0	1	1	0	0	1	1

1. ako je netačan samo jedan od bita parnosti, tada je on pogrešan
2. ako su netačni biti parnosti 1 i 2, tada je pogrešan bit podataka 3
3. ako su netačni biti parnosti 1 i 4, tada je pogrešan bit podataka 5
4. ako su netačni biti parnosti 2 i 4, tada je pogrešan bit podataka 6, a
5. ako su netačni biti parnosti 1, 2 i 4, tada je pogrešan bit podataka 7

# Reed-Muller-ov kod i Mariner 9



000000 – bela

000111 – siva

111111 – crna

Kodne reči:

- dužine 32 bita
- ukupno 64 različitih
- Hamingova distanca 16  
(ispravlja do 7 grešaka)

Slika sa 64 nivoa sive – 6 bitova + 26 bitova za ECC