Kodovi za otkrivanje i popravku grešaka

Greške u radu sa memorijom

Istraživanje iz 2010, sistem Jaguar, 360 TB 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 jednobitnih 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

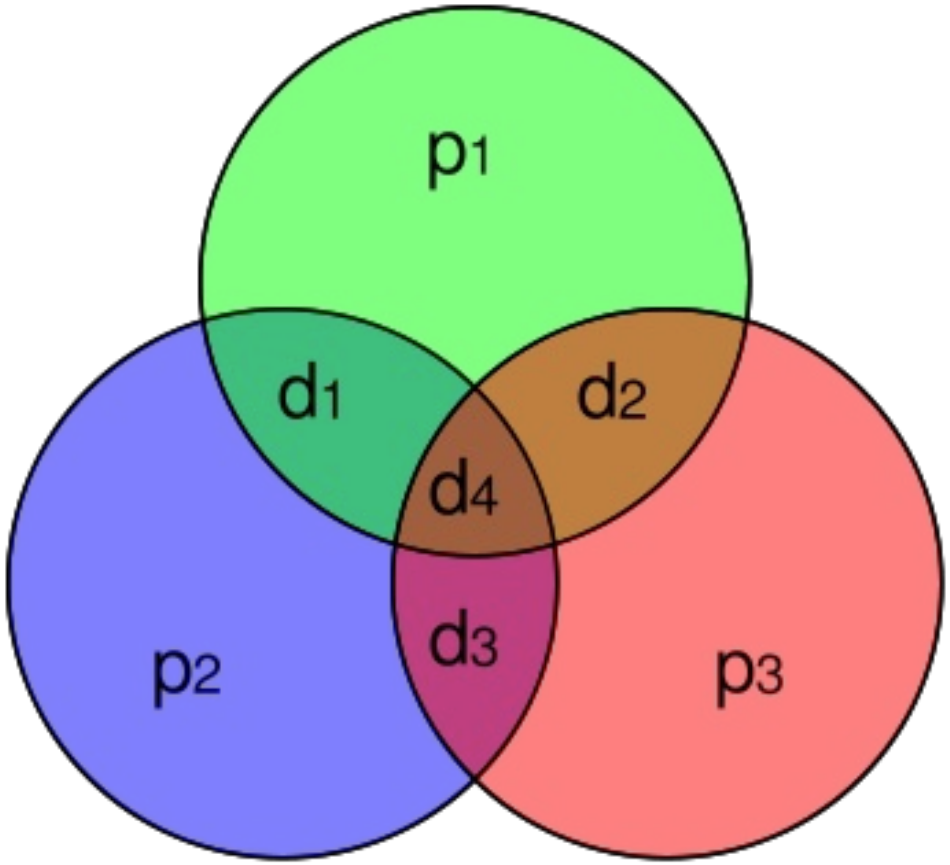
- sve kodne reči se razlikuju u dva bita =>

  može detektovati promenu jednog bita

Hamingova udaljenost – broj različitih bitova

Hamingov (7, 4) kod (1950)

Više bita parnosti može omogućiti i korekciju



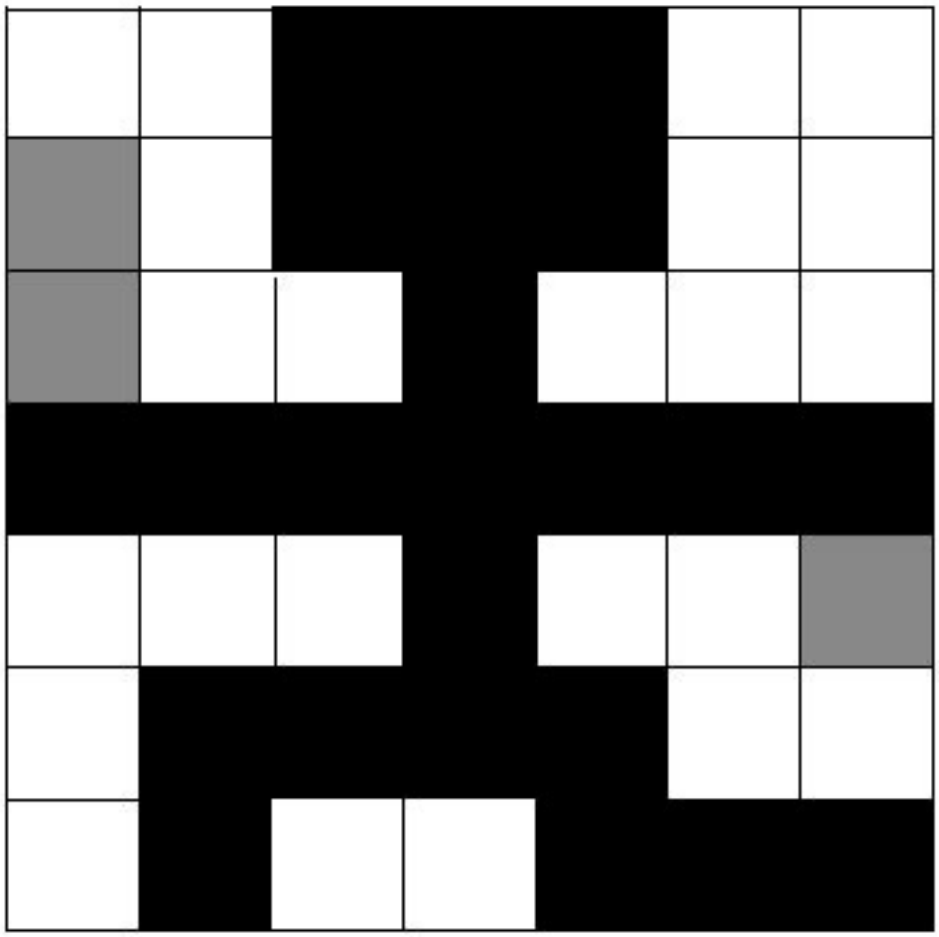
Hamingov (7, 4) kod (1950)

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

Operativni sistem

Operativni sistem

Rad sa BIOS-om i komandnim režimom oslonjenim na BIOS podrazumeva

- poznavanje šta je zauzeto, a šta slobodno u radnoj memoriji
- poznavanje šta je zauzeto, a šta slobodno u masovnoj memoriji
- poznavanje mašinskog formata naredbi

Operativni sistem

Mnogo je lakše

- čuvati podatke i programe u obliku datoteka:
  - datoteke sa podacima
  - datoteke sa programima
- pokretati programe bez ulaženja u detalje, kao što su:
  - gde će se program smestiti u memoriji
  - koje naredbe ga čine

Koncept **procesa** (engl. *process*)

- **program koji se izvršava, angažovanje procesora koje daje neki rezultat**

Koncept **datoteke** (engl. *file*)

- **razdvaja upotrebu sadržaja datoteke od načina organizacije**

Struktura operativnog sistema

Modul za rukovanje datotekama

- podržava operacije za rad sa datotekama: stvaranje, brisanje, čitanje, pisanje…
- omogućava razlikovanje datoteka putem naziva
- deskriptor datoteke: sadrži atribute datoteke
  - naziv
  - veličina
  - redni brojevi blokova
  - vreme nastanka, izmene, prava pristupa, ...
  - ...

Struktura operativnog sistema

Modul za rukovanje procesima

- podržava operacije za rad sa procesima: stvaranje, pokretanje, uništavanje
- slika procesa
  - naredbe koje čine program
  - vrednosti promenljivih
  - sadržaj steka
- deskriptor procesa: sadrži atribute procesa
  - broj lokacija za smeštanje
  - evidencija zauzetih lokacija
- inicijalna slika procesa se nalazi u izvršnoj datoteci

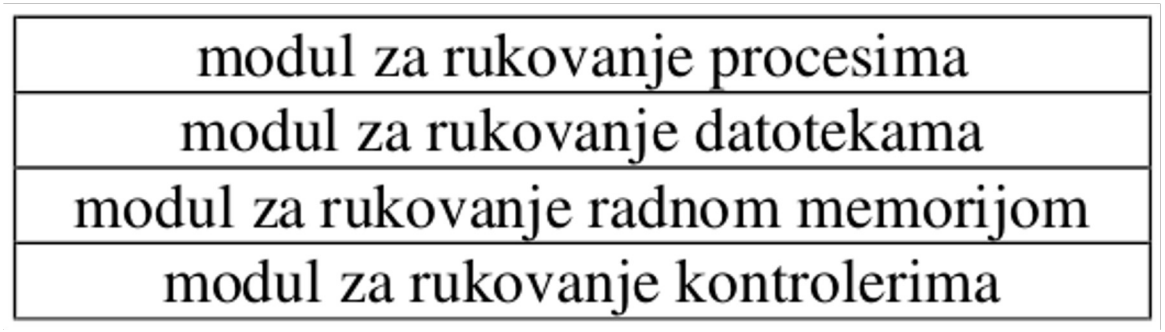
Struktura operativnog sistema

Modul za rukovanje radnom memorijom

- neophodan za rad prethodna dva modula
- rukovanje slobodnim i zauzetim lokacijama

Modul za rukovanje kontrolerima

- skup drajvera



Iznad OS-a su korisnički programi

Interpreter komandi operativnog sistema

Kada se uvedu datoteke i procesi, više nije poželjno da se

- direktno pristupa lokacijama memorije
- direkno pristupa blokovima na disku

Interpreter ostaje sa **samo jednom funkcijom**

- **pokretanje zadatog programa** (putem naziva)
  - preuzimanje imena izvršne datoteke
    - modul za rad sa kontrolerima
  - pokretanje programa
    - modul za rukovanje datotekama
    - modul za rukovanje procesima i radnom memorijom

Interpreter komandi operativnog sistema

Spada u korisničke programe

- izvršavanje se oslanja na OS
- OS se prema korisničkim programima odnosi kao prema svojim potprogramima

Dva nivoa korišćenja OS-a

- **interaktivni**
- **programski**
  - pozivanje operacija modula operativnog sistema – **sistemski pozivi**

Sistemski programi

- **editor**
- **makro pretprocesor**
- **prevodilac** (engl. *assembler/compiler*)
- **povezivač** (engl. *linker*)
- **punilac** (engl. *loader*)
- **dibager** (eng. *debugger*)
- pomoćni programi za rad sa datotekama

BIOS i OS

Računar započinje rad izvršavanjem BIOS-a

Inicijalni punilac (engl. *bootstrap loader*)

- običnо se nalazi u nultom bloku diska (engl. *boot block)*
  - MBR – *Master Boot Record*
  - GPT – *GUID Partition Table*
- BIOS (nakon početnih inicijalizacija računara) učitava nulti blok, smešta ga u memoriju i pokrene
- puni u radnu memoriju preostale delove OS-a

Više operativnih sistema – *multiboot*

Promena konteksta

Promena konteksta

Promena konteksta (engl. *context switch*) - preključivanje

Izbegavanje radnog čekanja

Višeprocesni režim rada

- više slika procesa istovremeno u memoriji
- preključivanje procesora sa jedne na drugu sliku

Stanja procesa

- aktivan
- čeka
- spreman

Sistemski proces

- aktivan kada svi ostali čekaju

Preključivanje

Do preključivanja dolazi

- kada se završi aktivnost procesa
- kada aktivnost procesa zavisi od spoljašnjeg događaja
  - komunikacija sa diskom
  - komunikacija sa terminalom
- UI vođeno preključivanje – obavljaju ga drajveri

Modul za rukovanje procesorom

- bira proces kome će se dodeliti procesor

modul za rukovanje procesima
modul za rukovanje datotekama
modul za rukovanje radnom memorijom
modul za rukovanje kontrolerima
modul za rukovanje procesorom

Preključivanje

- Ako su svi procesi nezavisni (zasebna memorija i datoteke), zajednički resurs su samo registri procesora, svakom procesu se dodeljuje registarski bafer
- %0 – bafer tekućeg aktivnog procesa, %1– bafer novog aktivnog procesa

```
                POČETAK     preključivanje
IZBACI          MAKRO       R
                PREBACI_RP  R,(%0)
                DODAJ_1     %0
                KRAJ
UBACI           MAKRO       R
                PREBACI_PR  (%1),R
                DODAJ_1     %1
                KRAJ
preključivanje: IZBACI      %2
                IZBACI      %3
                IZBACI      %4
                IZBACI      %5
                IZBACI      %6
                IZBACI      %7
                IZBACI      %8
                IZBACI      %9
                IZBACI      %10
                IZBACI      %11
                IZBACI      %12
                IZBACI      %13
                IZBACI      %14
                IZBACI      %15
                UBACI       %2
                UBACI       %3
                UBACI       %4
                UBACI       %5
                UBACI       %6
                UBACI       %7
                UBACI       %8
                UBACI       %9
                UBACI       %10
                UBACI       %11
                UBACI       %12
                UBACI       %13
                UBACI       %14
                UBACI       %15
                NATRAG
                KRAJ
```

Prekid

(engl. *interrupt*)

Prekid

Provera spoljašnjih događaja samo **prilikom preključivanja nije efikasna**

- do reakcije na spoljašnji događaj dolazi tek kada dođe trenutak preključivanja, iako se događaj mogao desiti i ranije

Dešavanje spoljašnjeg događaja treba da **odmah pokrene izvršavanje odgovarajućeg drajvera**

- prekid izvršavanja tekućeg procesa
- obrada događaja
- nastavak aktivnosti prekinutog procesa

Prekid

Prekid (engl. *interrupt*)

- **obrađivač prekida** (engl. *interrupt handler*)
  - preko vektora prekida (ulazna adresa obrađivača)
- svaka vrsta prekida (tastatura, disk, ...) ima svoj vektor i obrađivač
- prekide izazivaju kontroleri
  - javi procesoru da se desio događaj
  - dostavi vektor obrađivača prekida

Mehanizam prekida

Svi vektori prekida čine **tabelu vektora prekida** kojoj se pristupa preko broja prekida

Linija najave prekida (engl. IRQ – *interrupt request*)

– kontroler javlja da se desio događaj

Linija potvrde prekida (engl. *interrupt acknowledge*)

– procesor traži broj vektora

Čuvanje programskog brojača (%13) i status registra (%14) pre obrade prekida

Obrađivač prekida čuva preostale registre

Mehanizam prekida

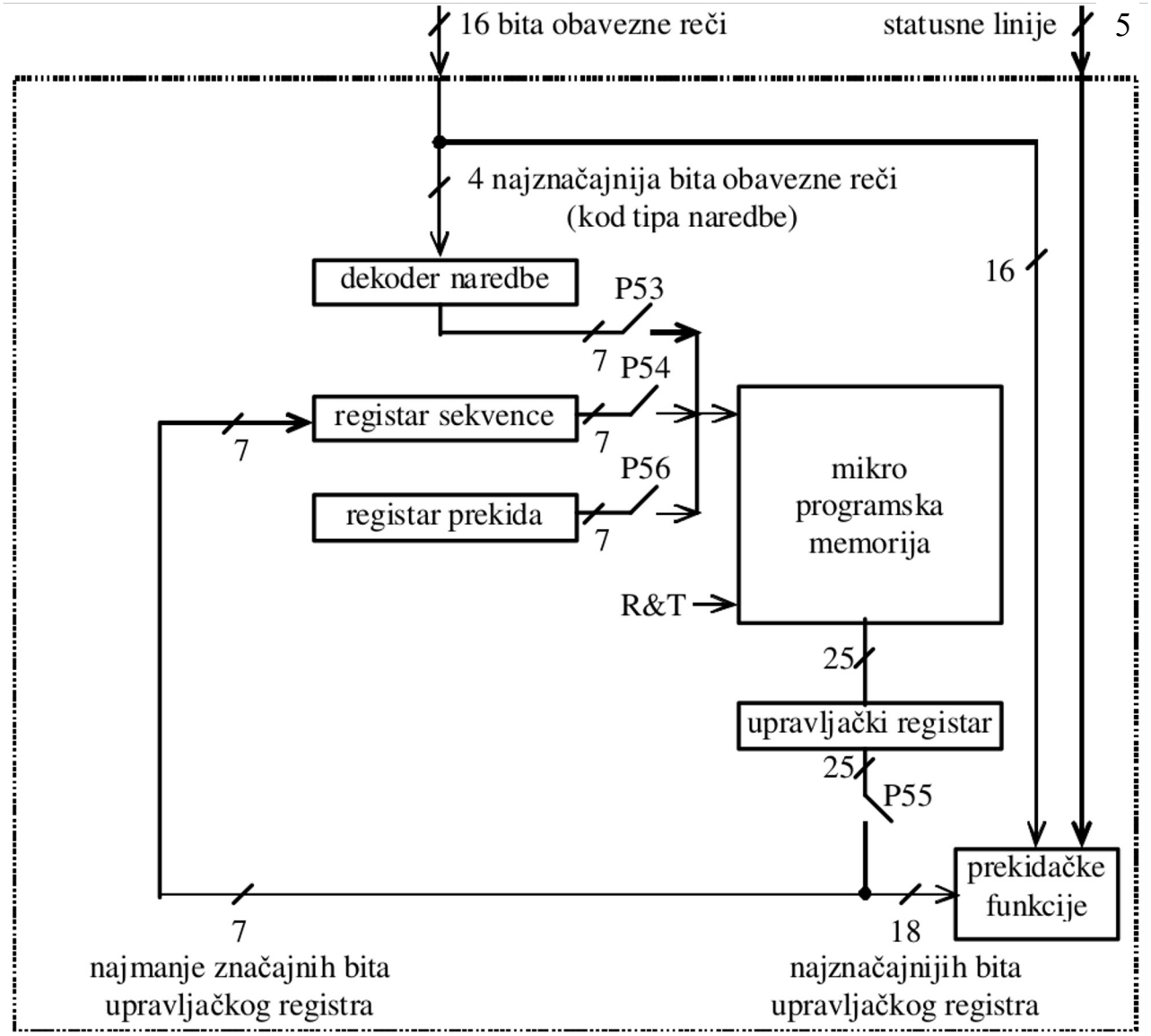
Obrađivači prekida po završetku treba da restauriraju i programski brojač i status registar

- naredba NASTAVI
- kod KONCEPT-a nema prekida unutar prekida

  SR_4 – bit prekida (1-omogućeni)
- postavlja se na 0 čim se uđe u obradu prekida

Stek omogućava prekide u više nivoa

Realizacija prekida



Upravljačka jedinica

registar prekida:

sadrži adresu mikro-programa prekida

Realizacija prekida

Logička promenljiva NAJAVA

Izvršavanje mikro programa prekida – pre faze dobavljanja sledeće naredbe

Mikro-program dobavljanja na 0000001_2

PRE_DOBAVLJANJA = $\sim RS_6 \& \sim RS_5 \& \sim RS_4 \& \sim RS_3 \& \sim RS_2 \& \sim RS_1 \& RS_0$

Mogućnost obavljanja prekida

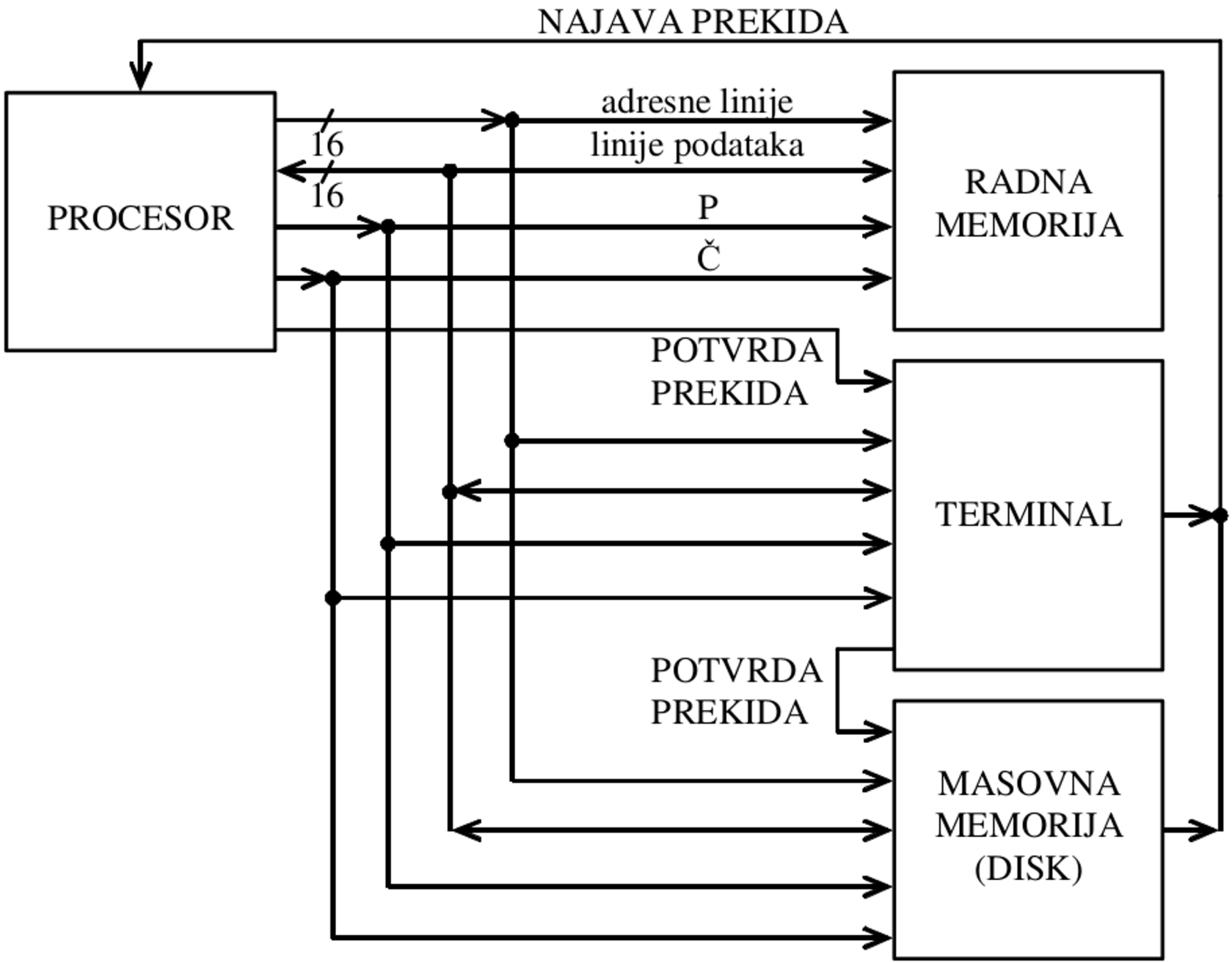
PREKID = NAJAVA&SR_4&PRE_DOBAVLJANJA

Rukovanje P56 i P54

P56 = R&T&PREKID

P54 = R&T&~IZA_DOBAVLJANJA&~PREKID

Serijsko povezivanje kontrolera na liniju potvrde


NAJAVA PREKIDA
PROCESOR
16
16
adresne linije
linije podataka
P
Č
RADNA MEMORIJA
POTVRDA PREKIDA
TERMINAL
POTVRDA PREKIDA
MASOVNA MEMORIJA (DISK)

Realizacija prekida

Svaki kontroler ima u sebi registar broja prekida

– stavlja ga na linije podataka po dobijanju potvrde

Mikro-program prekida

1. ciklus: programski brojač → %13
2. ciklus: status registar → %14
3. ciklus: 0 → SR_4
4. ciklus: 1 → POTVRDA PREKIDA
   linije podataka → pomoćni registar
5. ciklus: pomoćni registar → adresne linije
   1 → č
   linije podataka → programski brojač

Mikro-program naredbe NASTAVI

1. ciklus: %13 → programski brojač
2. ciklus: %14 → status registar

Odnos obrade prekida i preključivanja

Obrada prekida ne zahteva preključivanje, ali ga može izazvati

Prioritet procesa

Na početku rada

- inicijalizacija tabele prekida (funkcija modula za rukovanje kontrolerima)
- omogućavanje prekida

Podela drajvera

- donji deo – **obrađivač prekida**
- gornji deo – **komunikacija sa višim slojevima**

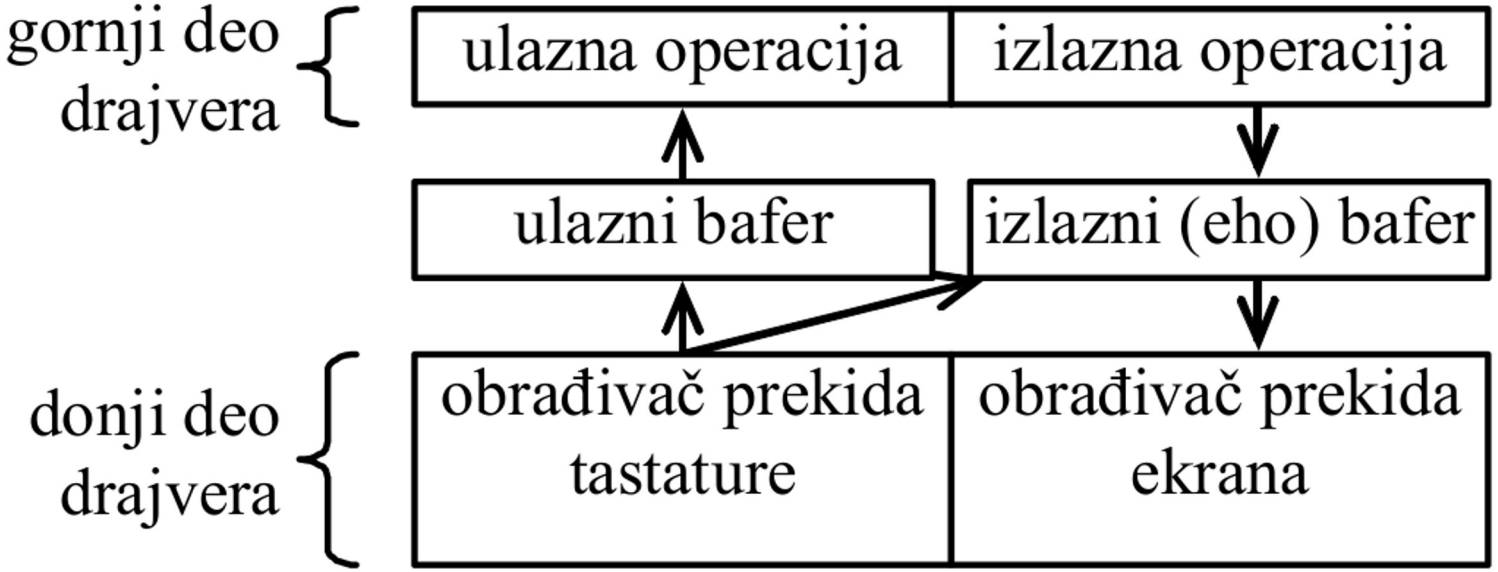
Organizacija drajvera terminala

Donji deo

- obrađivač prekida tastature
- obrađivač prekida ekrana

Gornji deo

- potprogrami terminala



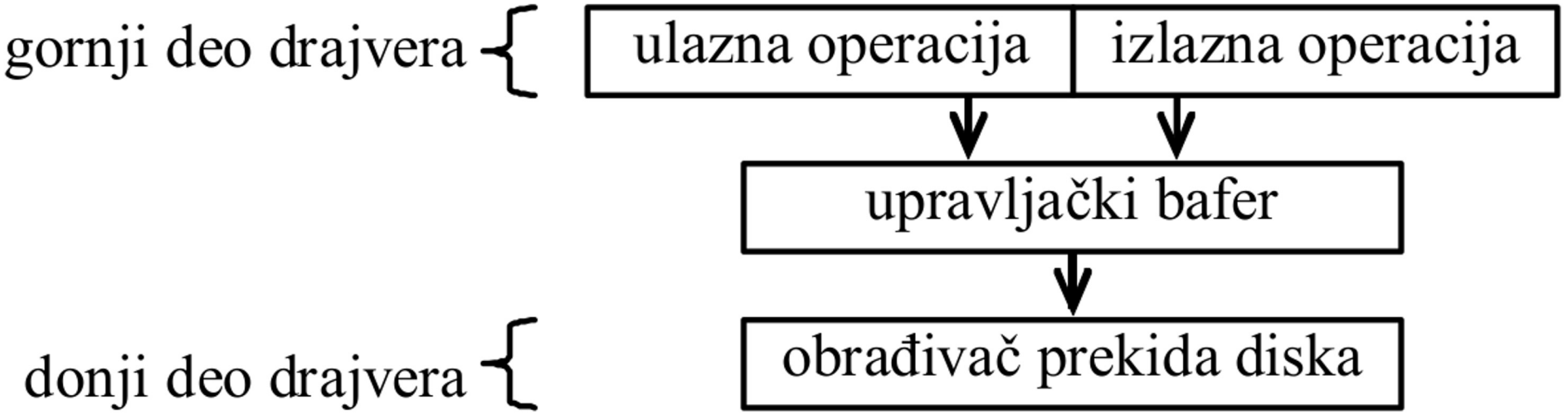
Organizacija drajvera diska

Donji deo

– obrađivač prekida diska

Gornji deo

– ulazna i izlazna operacija (prijavljuju kraj rada tek kada ceo blok bude prenet)



Usklađivanje rada kontrolera i uređaja

Asinhroni rad:

- kontroleri
- procesor

Sinhroni rad:

- kontroler
- uređaj

Rukovanje (engl. *handshaking*)

- logička promenljiva KONTROLER
- logička promenljiva UREĐAJ
- samo kada su obe na 1 moguća je komunikacija

Sabirnice (magistrale)

Sabirnica (magistrala)

Veliki broj prekida koje treba obraditi pri prenosu bloka sa diska ili na disk

Sabirnica (magistrala – engl. *bus*)

- direktna veza između svih delova računara

DMA (engl. *direct memory access*) kontroler

- rukovanje sabirnicom
  - ZAHTEV (engl. *bus request*)
  - DOZVOLA (engl. *bus grant*)

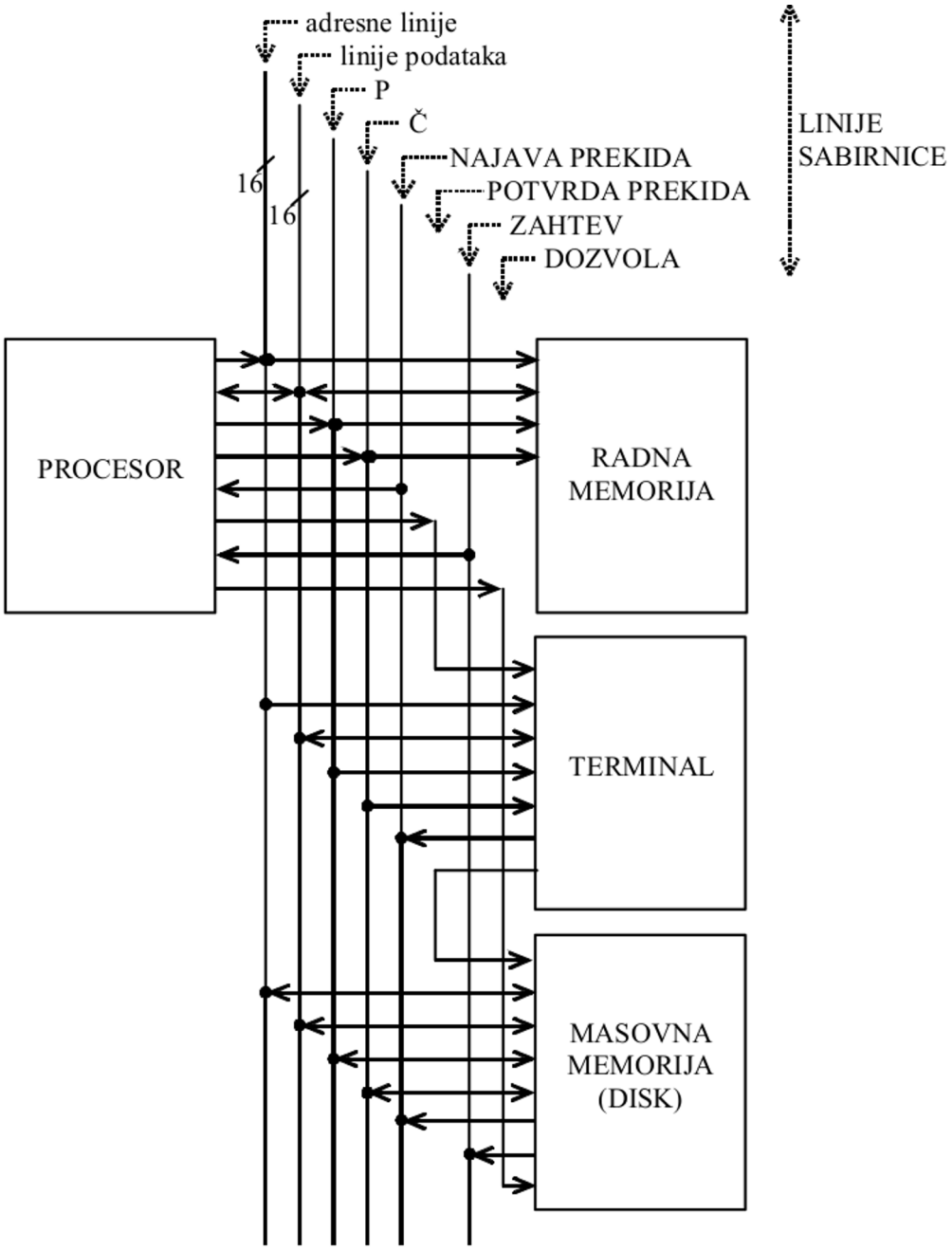
Sabirnica (magistrala)

DMA kontroler diska

- registar broja staze
- registar broja sektora
- registar broja bajtova (za prenos)
- registar adrese (prva lokacija u radnoj memoriji)
- registar stanja (smer prenosa)
- registar podataka

Manje angažovanje procesora oko prekida

- ali i usporenje procesora kada se koristi DMA



Koncept

DMA se vrši u toku dobavljanja naredbi

Početak obavljanja zavisi od P, Č i DOZVOLA

Više DMA kontrolera se serijski povezuju na signal DOZVOLA

Linije sabirnice:

- adresne linije
- linije podataka
- upravljačke linije

Višekorisnički rad

Višekorisnički rad

Periodični prekidi

- **kružno preključivanje** (engl. *round robin*)
- (sistemski) **sat**: kristalni oscilator + brojač impulsa
- **sistemsko vreme**

Višekorisnički rad

- više terminala povezanih na jedan računar
- privid da računar istovremeno opslužuje više korisnika zasnovan na velikoj brzini procesora

Logički i fizički adresni prostori

Međusobna zaštita procesa (raznih korisnika)

Logički adresni prostor

- logička adresa i fizička adresa
- logička adresa: od 0 do granične (najveća log. adresa)
- poređenje tekuće logičke adrese sa graničnom
  - *i*-ti bit viši od granične adrese $V_i = L_i \& (\sim G_i)$
  - *i*-ti bit niži od granične adrese $N_i = (\sim L_i) \& G_i$

- izlazak van logičkog adresnog prostora:

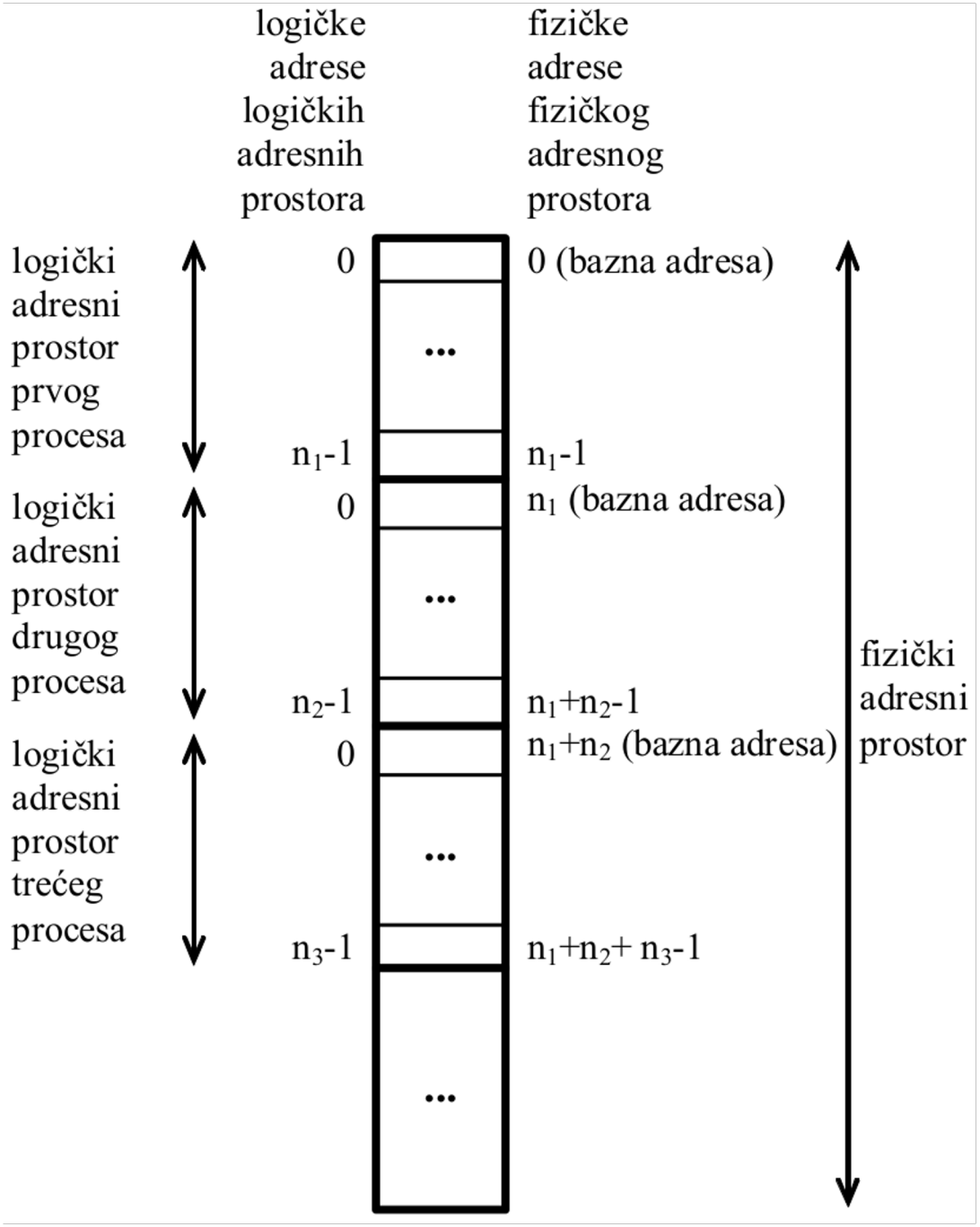
$$V = V_{15} \mid (\sim N_{15} \& (V_{14} \mid (\sim N_{14} \& (\ \ldots \ (V_1 \mid (\sim N_1 \& V_0)) \ \ldots \ ))))$$

Pretvaranje logičke adrese u fizičku

Moguće je samo ako je V netačno!

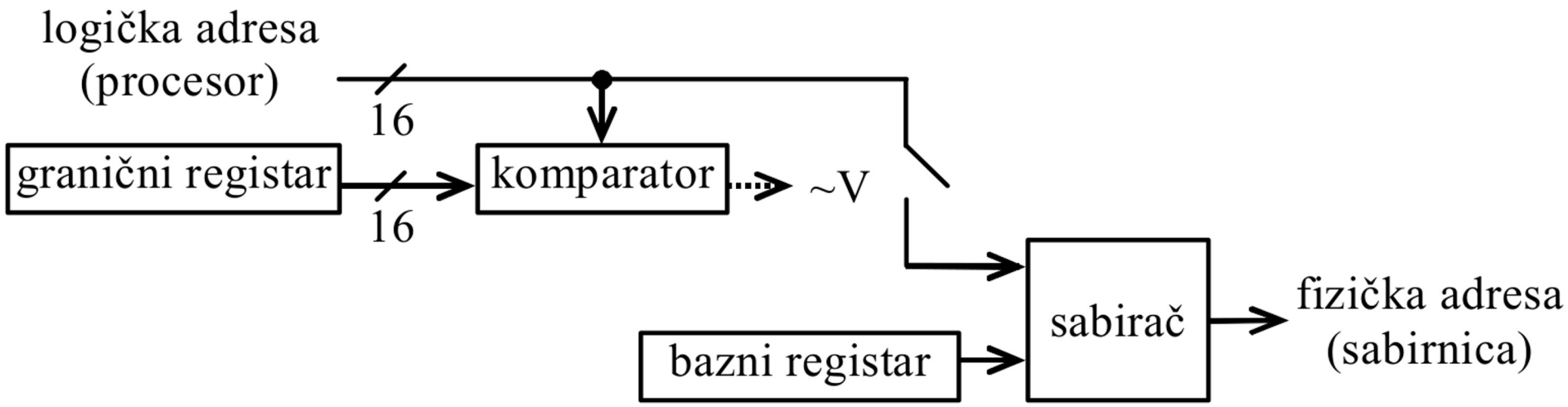
Sabiranje logičke adrese sa baznom

Granični (engl. *limit*)i **bazni** (engl. *base*) **registar** za čuvanje adresa



Pretvaranje logičke adrese u fizičku

MMU (engl. *Memory Management Unit*)



Izuzetak (engl. *exception*)

Ako je logička adresa neispravna (~V)

- MMU detektuje izlazak van opsega adresa

Procesor obrađuje **izuzetak**

- **mikro-program izuzetka**
  - sličan mikro-programu prekida
- registar broja vektora
- **obrađivač izuzetka** (engl. *exception handler*)
- **registar izuzetka** (adresa mikro-programa)
  - sličan registru prekida, dodaje se upravljačkoj jedinici

Razlika između izuzetaka i prekida

1. Pojavu izuzetka otkriva MMU, a ne kontroler
2. Broj vektora izuzetka pribavlja procesor, a ne kontroler
3. Obrada izuzetka počinje odmah po njegovom otkrivanju
4. Izuzeci ne mogu biti onemogućeni

Privilegovani i neprivilegovani režim rada

Rukovanje baznim i graničnim registrima
- privilegovane naredbe/režim rada (OS)
- neprivilegovane naredbe/režim rada (programi)

Fizička memorija
- **korisničkom prostoru** (engl. *user space*) pristupaju procesi u **neprivilegovanom režimu rada**
- **sistemskom prostoru** (engl. *kernel space*) pristupa OS u **privilegovanom režimu rada**

SR_5 – bit privilegije
- Kako se SR_5 ne bi neovlašćeno menjao, NASTAVI spada u privilegovane naredbe
  - obrađivači prekida i izuzetaka – privilegovani potprogrami

Realizacija sistemskih poziva

Kako pristupiti funkcijama operativnog sistema, kada spadaju u privilegovani kod?

Sistemski pozivi se realizuju kao **obrađivači izuzetaka**

Naredba IZAZOVI <broj_vektora>

Pored poziva operacije, prevodi i procesor u privilegovani režim rada