Ispravljanje gresaka

- Racunarske memorije su podlozne greskama, usled raznih smetnji, strujnih udara I slicno.
- Za zastitu od takvih gresaka, neke memorije koriste kodove za otkrivanje I ispravljanje gresaka.
- U tom slucaju svaka memorijska rec dobija dodatne bitove cija se vrednost formira na tacno definisan nacin.
- Kada se rec cita iz memorije, proveravaju se dodatni bitovi na osnovu cega se utvrdjuje da li je sadrzaj memorije ispravan.

Greške pri prenosu podataka

- Pri prenosu podataka često dolazi do promene pojedinih bitova podataka zbog:
- smetnji na prenosnom putu.
- različitih tipova šumova na lokacijama salnja i prijema podataka.
- smetnje se dešavaju bez obzira na udaljenost uređaja, kako pri prenosu podataka između dva računara tako i između komponenti istog računara.

Greške pri prenosu podataka

Postoje dva osnovna pristupa rešavanju ovog problema:

- kontrola sa povratnom spregom
- kontrola grešaka unapred

Kontrola sa povratnom spregom

- Uz podatke se šalju dodatne informacije koje služe da se ustanovi da postoje greške, ali ne i da se one otklone.
- Neispravno preneseni podaci se ponovo šalju.

Kontrola grešaka unapred

- Uz podatke se šalju dodatne informacije koje služe kako da se ustanovi da greške postoje, tako i da se odredi njihova lokacija.
- Neispravno preneseni podaci se automatski koriguju.

Pouzdanost komunikacije

- Pouzdanost komunikacije se predstavlja *brojem bitova sa greškom*(engl. *bit error rate*–BER)
- BER je verovatnoća pojavljivanja neispravnog bita u definisanom vremenskom intervalu
- računarske mreže imaju BER oko 10^{-12} (u proseku 1 od 10^{12} bitova ima grešku u definisanom vremenskom intervalu
- unutar računarskog sistema BER je oko 10^{-18} ili manje

Najcesce metode za otkrivanje grešaka

- kontrola parnosti
- provera zbira bloka
- ciklična provera redundanci

Kontrola parnosti

- Kontrola parnosti je jedan od najjednostavnijih metoda za otkrivanje grešaka
- Koristi se za otkrivanje pogrešnih vrednosti bitova

Kontrola parnosti

- Uz svaku n-bitnu reč se dodaje po jedan bit tako da ukupan broj binarnih jedinica u tako proširenoj reči bude paran (neparan)
- Verovatnoća da se greška ne primeti je reda $n^2/4*BER^2$
- Greška se ne primećuje ako je izmenjen paran broj bitova

Kontrola parnosti u 2D

Pri prenosu bloka podataka:

- i dalje se svaka osnovna reč proširuje radi tzv. “horizontalne” provere parnosti
- čitavom bloku dodaje se još jedna (proširena) reč tako da za svaku vrednost bita postoji dodatna “vertikalna” provera parnosti

Primer

	Bitovi parnosti	ASCII kod	Karakter
	0	1000010	B
	1	1000101	E
	1	1001111	O
	0	1000111	G
	1	1010010	R
	0	1000001	A
	0	1000100	D
reč parnosti	1	1011000	

Kontrola zbira

Ako je blok duži od jedne reči, često se primenjuje metod kontrole zbira.

- Formira se zbir svih reči u bloku i prenese zajedno sa porukom.
- Obično se zbir skraćuje, recimo na 32 bita.
- Primalac ponovo izračunava zbir i poredi sa primljenim podatkom

Kontrola zbira

- Ne može da prepozna greške neispravnog redosleda reči, kao ni dodavanje ili gubljenje jedinica bloka čiji su svi bitovi binarne nule.

Ciklična provera redundanci

Ciklička provera redundanci (engl. *Cyclic Redundancy Checking*–CRC) otkriva:

- sve greške na neparnom broju bitova
- sve 2-bitne greške
- proširene greške čija je dužina manja od broja redundantnih bitova

Ciklična provera redundanci

Metod CRC je nešto složeniji, ali se često implementira hardverski I bazira se na:

- aritmetici po modulu 2 (tj. bez prenosa i pozajmica)
-

$$0 + 0 = 1 + 1 = 0,\ 0 + 1 = 1 + 0 = 1$$
$$1 - 1 = 0 - 0 = 0,\ 0 - 1 = 1 - 0 = 1$$

Ciklična provera redundanci

- deljenju polinoma (u aritmetici po modulu 2)
- niz bitova bloka koji se prenosi se posmatra kao niz koeficijenata polinoma, npr.

$a_n a_{n-1} \ldots a_1 a_0$ odgovara polinomu $M(x)=a_n x_n+a_{n-1} x_{n-1}+\ldots+a_1 x+a_0$

Ciklična provera redundanci

Postupak kodiranja:

- odabere se “polinom generator” *G(x)*stepena *k*
- izračunava se $x^k M(x)/G(x)$; dobijeni ostatak se označava sa *R(x)*
- koeficijenti ostatka (njih *k*) se dodaju na kraj poruke

Ciklična provera redundanci

Postupak dekodiranja:

- primljena polinomijalna kodna reč se deli sa *G(x)*
- ako je ostatak deljenja 0, nema grešaka pri prenosu
- ako ostatak nije nula, postoje greške pri prenosu

Ciklična provera redundanci

Postoje greške koje se ovako ne mogu otkriti, ali se dobrim izborom polinom generatora njhov broj smanjuje:

•

Otkrivanje I ispravljanje gresaka u radu sa memorijom

- Osim pri prenosu, postoji i mogućnost nastajanja greške pri zapisivanju i čitanju podataka, kao i tokom njihovog čuvanja
- Neki od bitova podataka može da bude promenjen posle zapisivanja podataka.
- Međutim, prenos podataka od prvobitnog izvora ne može da se ponovi, jer u najvećem broju slučajeva izvor više i ne postoji.
- Stoga se podaci tako zapisuju da je pri njihovom čitanju omogućeno ne samo otkrivanje, već i korekcija pronađene greške.

Otkrivanje I ispravljanje gresaka u radu sa memorijom

Stalno prisutni defekti

- Usled neispravnosti memorijska ćelija nije u stanju da pouzdano čuva podatke i ona ih bez spoljašnjeg uzroka menja sa 0 na 1 ili obratno

Otkrivanje I ispravljanje gresaka u radu sa memorijom

Prolazni defekti

- predstavljaju slučajne izmene sadržaja jedne ili više memorijskih ćelija

•obično su posledica smetnji u napajanju ili elektromagnetnog ili radioaktivnog zračenja

Vrste kodova

- SED (single error detection) –kod koji omogućava detekciju grešaka na jednom bitu
- SEC (single error correction) –kod koji omogućava korekciju grešaka na jednom bitu

Hamingovi kodovi

Najprostiji kod za otkrivanje i korekciju grešaka je *Hamingov kod*

Ispravljanje gresaka

- Ako imamo dve kodne reci, 10001001 i 10110001 mozemo da utvrdimo koliko se odgovarajucih bitova u njima medjusobno razlikuje.
- 10001001
- 10110001
- Najaksi nacin da se utvrdi koliko bitova se razlikuje, primeni se logicko XOR I prebroje se jedinice u rezultatu.

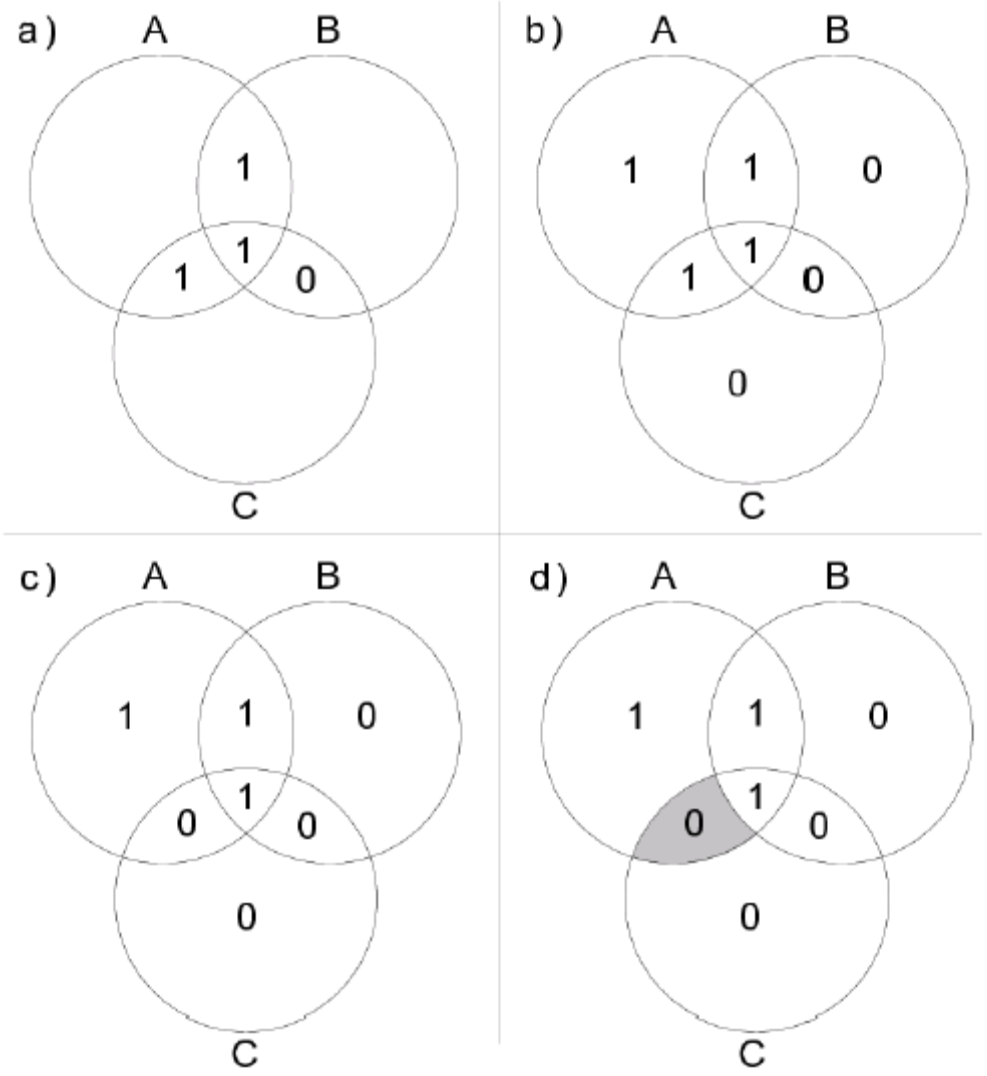
Hammingovo rastojanje

- Broj uparenih bitova po kojima se razlikuju dve kodne reci zove se Hammingova rastojanje.
- Na osnovu Hammingovog rastojanja, da se dve kodne reci cije je Hammingovo rastojanje d mogu izjednaciti nakon d jednobitnih gresaka.
- Za prethodni primer, Hammingovo rastojanje je 3.

Primer Hamingovog SEC koda

- primer za reči dužine 4 (3 dodatna bita)
- u polje koje predstavlja presek bar dva kruga upiše se po jedna vrednost bita iz reči
- u preostala polja se upisuje 0 ili 1 tako da se u svakom krugu očuva parnost

Primer Hamingovog SEC koda



a) bitovi podatka

b) sa bitovima parnosti

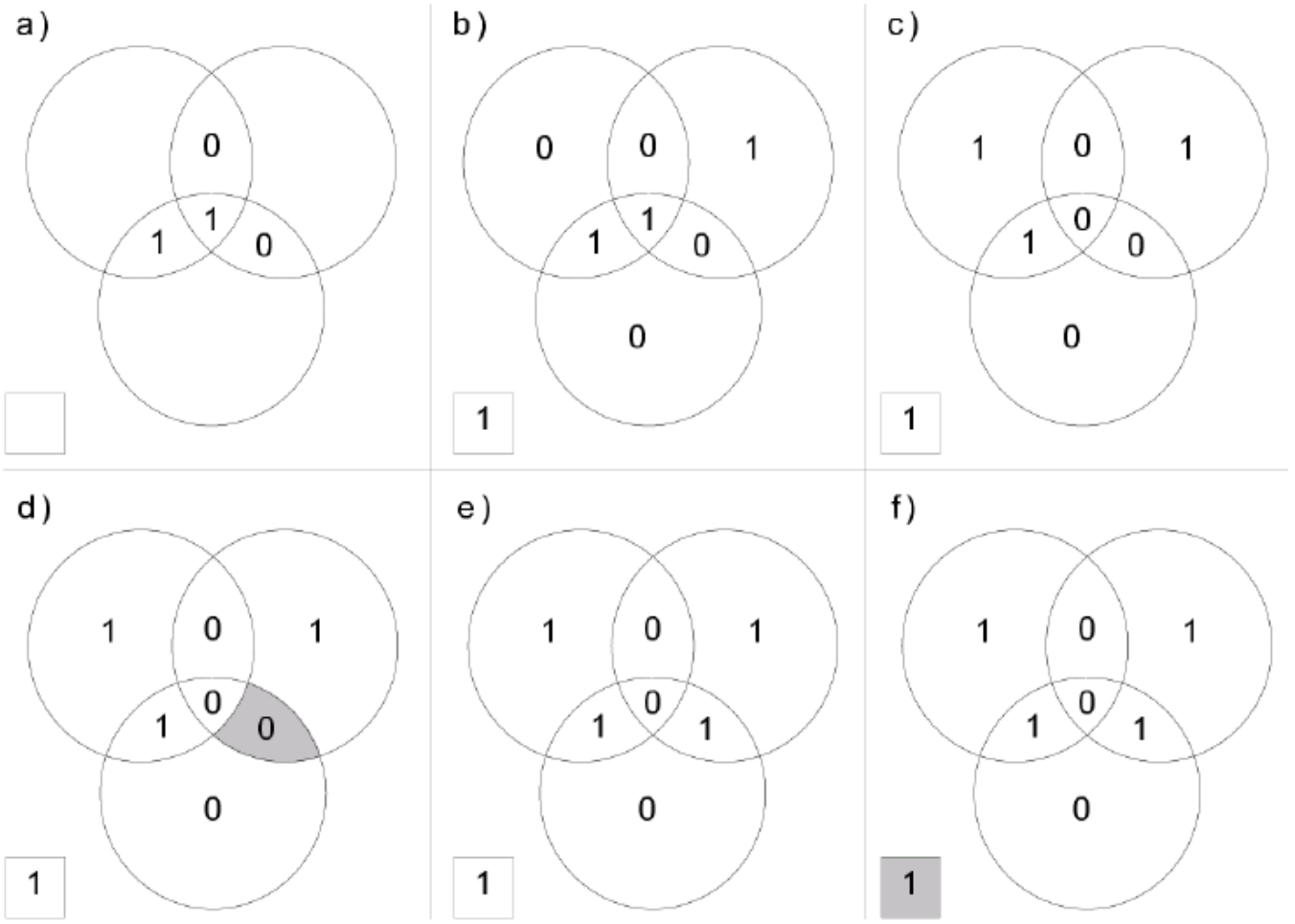
c) promena bita usled greške

d) lokalizovana greška

Hamingovo SEC DED(double error detection) kodiranje

- Obično je za to dovoljno dodavanje svega jednog bita
- Memorija najčešće poseduje ugrađeni SEC-DED kod.

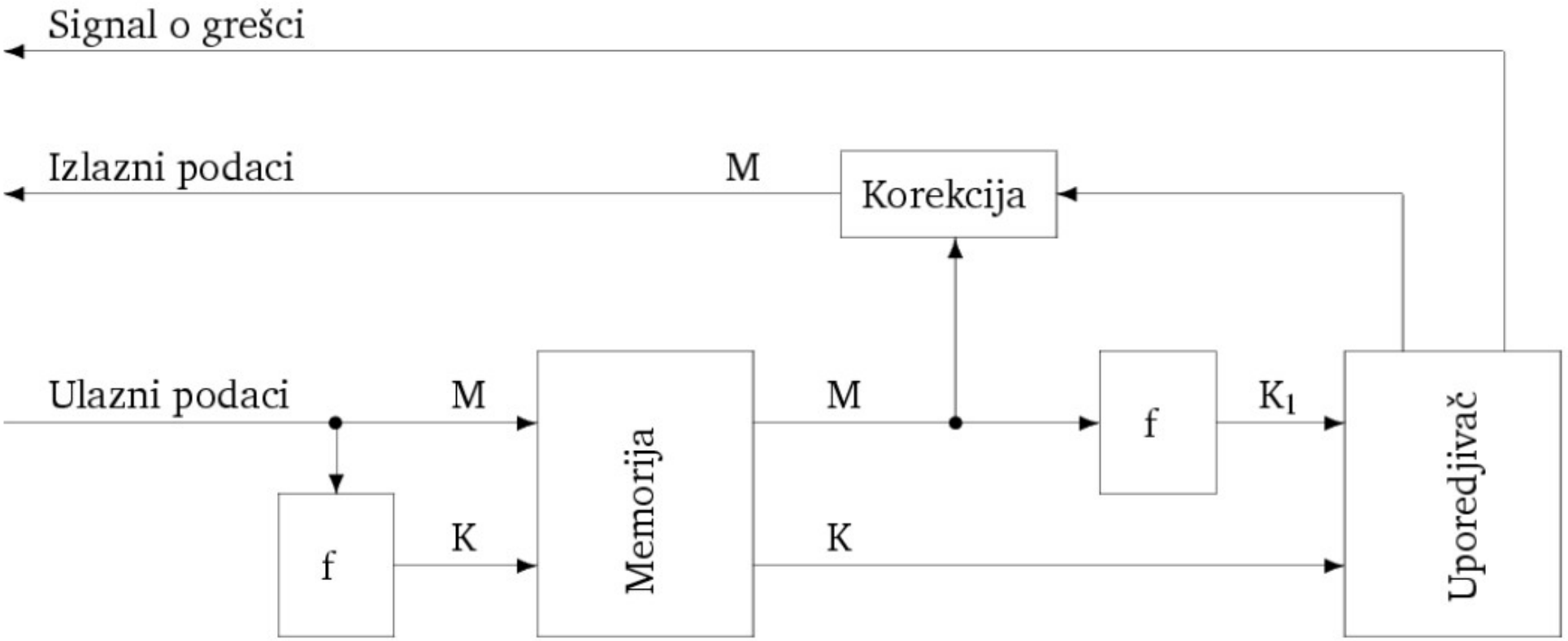
Hamingovo SEC DED(double error detection) kodiranje



Dodaje se još jedan bit tako da je ukupan zbir bitova paran

a) bitovi podatka
b) bitovi parnosti
c) promenjena dva bita
d) lokalizovana greška
e) ispravljena greška
g) prepoznato da postoji greška

Čuvanje i provera korektnosti zapisa



Koncept provere

- Za reč dužine M bitova generiše se kod dužine K
- Zapisuje se M+K bitova
- Nakon čitanja se ponovo generiše ključ K_1 i poredi sa K ekskluzivnom disjunkcijom (reč koja se dobija naziva se *sindrom*):
- ako je razlika 0, smatra se da nema greške