nedeljainformatike v6.0

# Da li je poruja ispravna?

Katarina Krivokuća

Matematička gimnazija

27. 04. 2021.

# “Poruja”=Poruka

- Kako vidimo da postoji greška?
- Kako znamo da je ispravimo?

**Azbuka(V)-** skup mogućih slova
**Skup reči nad azbukom(V*)-** skup svih mogućih reči
**Jezik(L)-** skup svih validnih reči - *CODEWORDS*

# Prenos jednog bita

- Od sada je azbuka uvek V= {0,1}
- Šaljemo poruke preko kanala sa **verovatnoćom greške jednog bita p**

Pravila za greške:

1. Slučajne
2. Nezavisne
3. Simetrične

**Podrazumevamo da je p dovoljno malo da je verovatnoća da se dve greške dese skoro nepostojeća!**

# Prenos jednog bita

Šaljemo poruku 0 ili 1.

- Verovatnoća pogrešnog tumačenja je p
- Ne znamo da kažemo da li je nastala greška pri slanju

**Pametnije: Trostruki bit**

$$V^*=\{000, 001, 010, 011, 101, 110, 111\}$$
$$L=\{000, 111\}$$

# Prenos jednog bita

**Pametnije: Trostruki bit**

Kodiranje:

| | |
|---|---|
| 0 | 000 |
| 1 | 111 |

Dekodiranje:

| | |
|---|---|
| 000 | 0 |
| 001 | 0 |
| 010 | 0 |
| 011 | 1 |
| 100 | 0 |
| 101 | 1 |
| 110 | 1 |
| 111 | 1 |

Verovatnoća greške: $p^2(3-2p)$

Šaljemo 0:

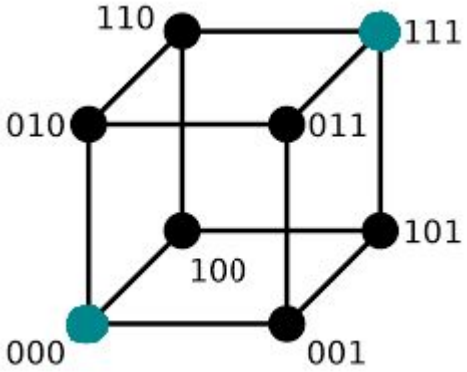
| stiglo | dekodirano | verovatnoca |
|---|---|---|
| 000 | 0 | $(1-p)^3$ |
| 001 | 0 | $(1-p)^2p$ |
| 010 | 0 | $(1-p)p(1-p)$ |
| 011 | 1 | $(1-p)p^2$ |
| 100 | 0 | $p(1-p)^2$ |
| 101 | 1 | $p(1-p)p$ |
| 110 | 1 | $p^2(1-p)$ |
| 111 | 1 | $p^3$ |

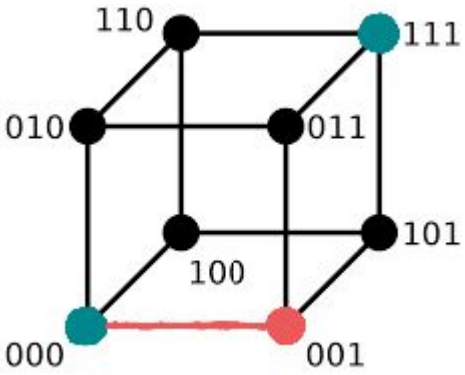
# Prenos jednog bita


Šta predstavlja ova strelica?
**Linearni operator!**

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} [x] = \begin{bmatrix} x \\ x \\ x \end{bmatrix}$$

# Prenos jednog bita

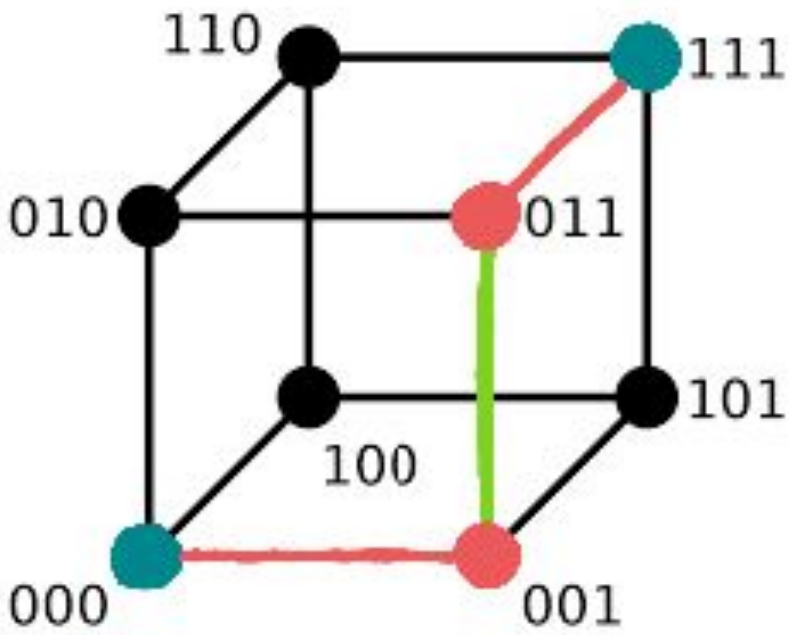

Jedna greška odgovara prolasku kroz jednu stranicu kocke.

Dužina najkraćeg puta između dve reči na kocki je **Hamingova distanca** tih reči.

**Hamingova distanca jezika** je najmanja Hamingova distanca njegovih reči- ovde je 3.

# Prenos jednog bita

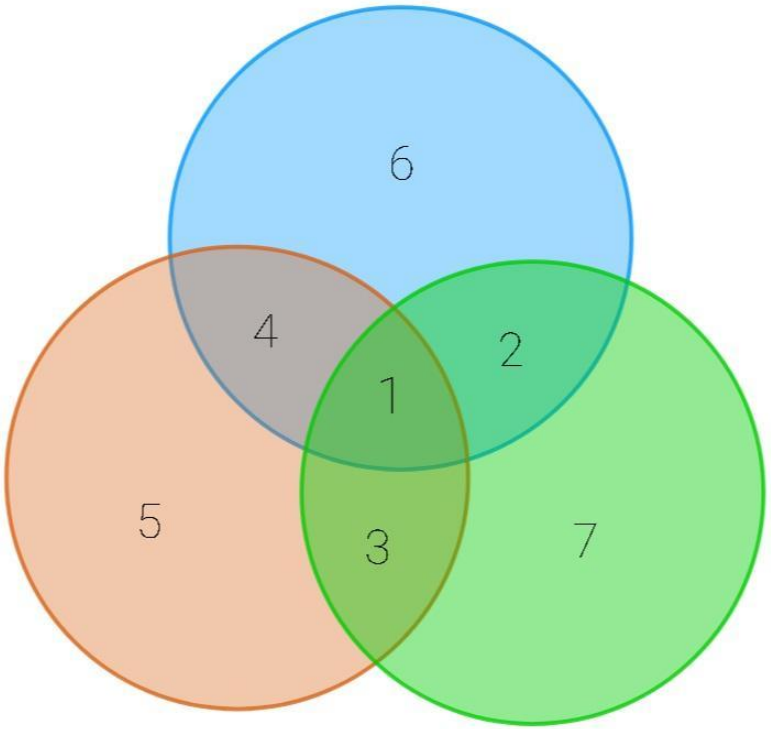
**/Teorema/:** Kod moze ispraviti n grešaka ako je Hamingova distanca njegovog jezika barem 2n+1.



# Hamingov (7,4) kod

Želimo da smanjimo broj bitova koje dodajemo, ali da zadržimo sposobnost ispravljanja jedne greške.

Šaljemo 4 bita kao jednu poruku!




**Ideja kodiranja:**

Za poruku dužine 4 šaljemo 7 bita:

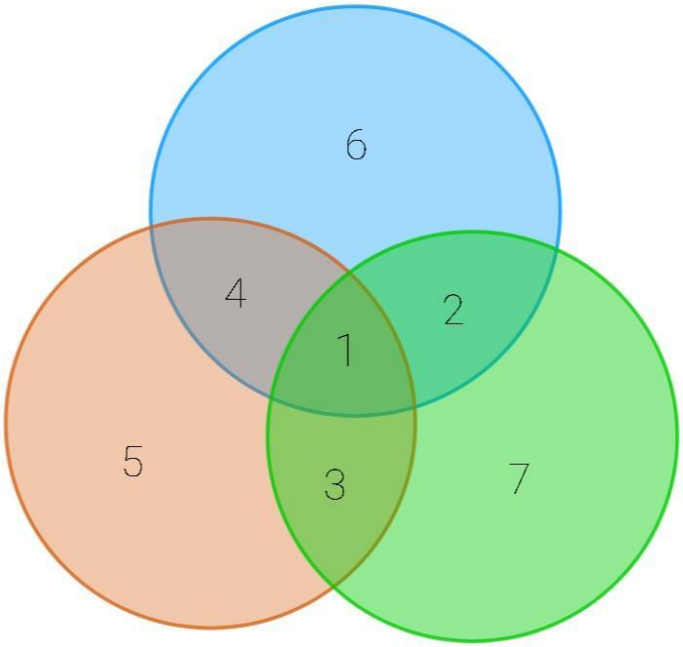
- na mestima 1-4 su bitovi poruke
- na mestima 5-7 su pomoćni bitovi

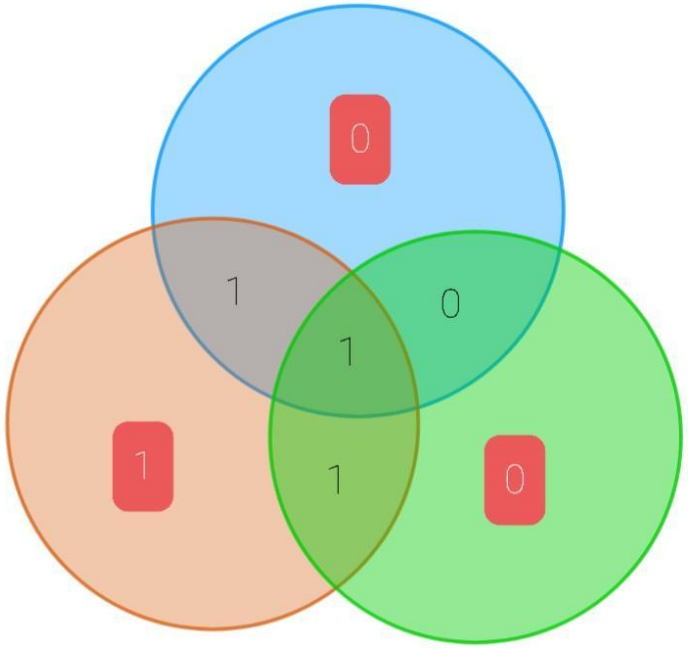
Pomoćne bitove biramo tako da **u svakoj od tri kružnice broj jedinica bude paran.**

Prevod- **xor brojeva u svakoj kružnici je 0**

# Hamingov (7,4) kod

Šaljemo poruku **1011**:



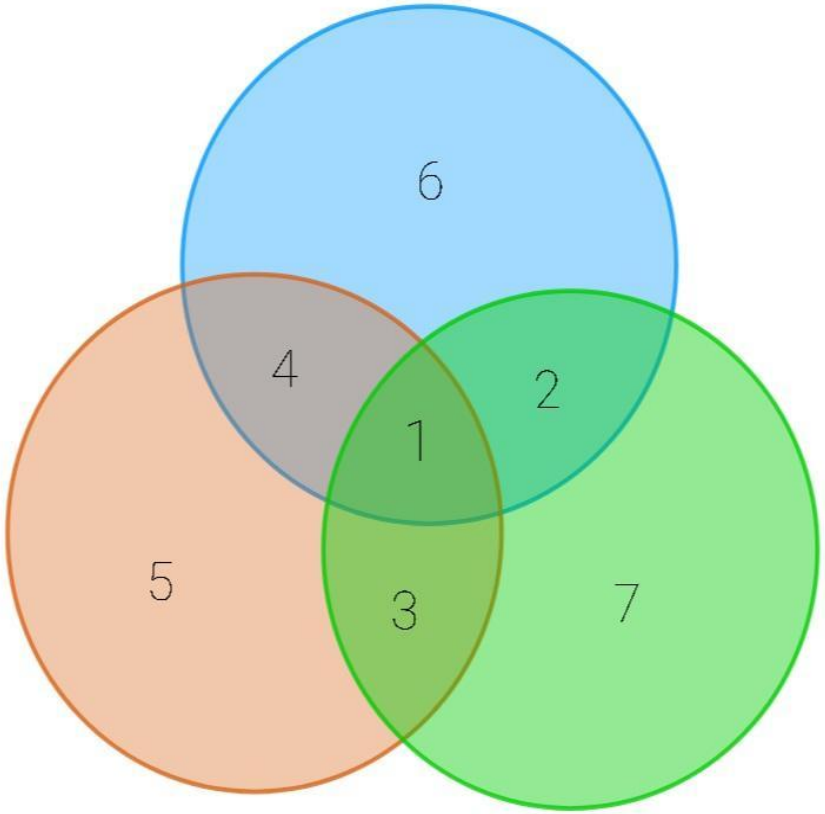




Poslato je: **1011100**

# Hamingov (7,4) kod

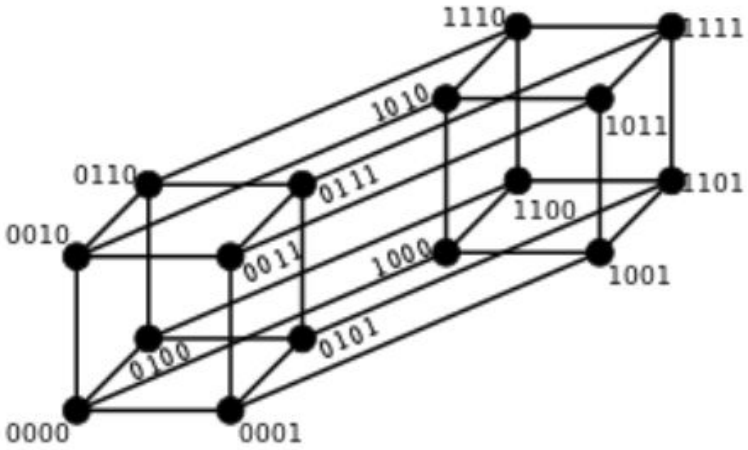
**Ideja dekodiranja:**

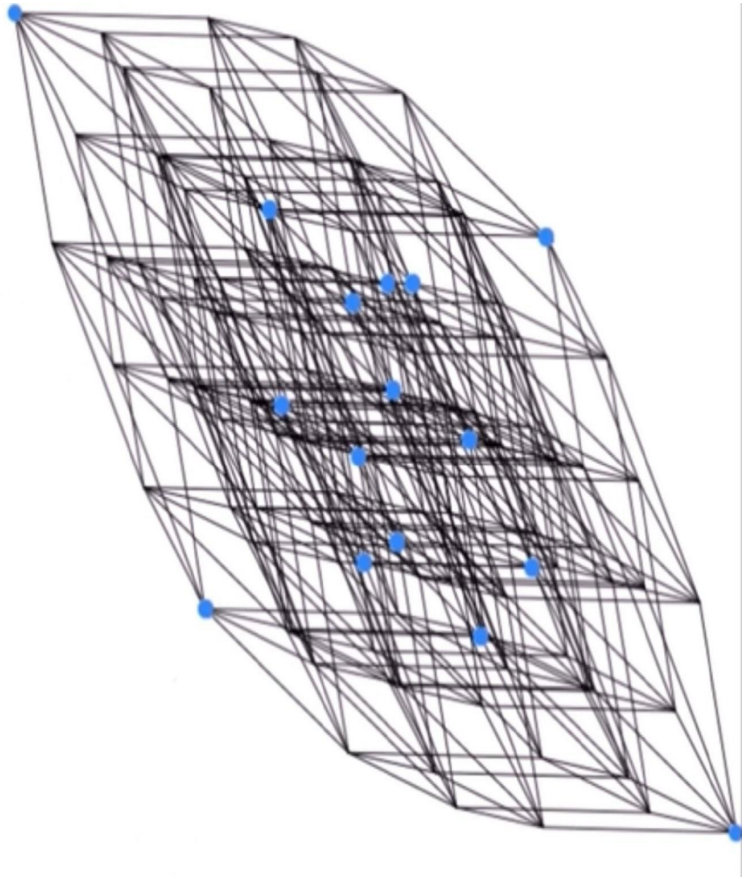
1. Postavimo bitove poruke u Venov dijagram
2. Proveravamo xor u skupovima:
   - Ako su sva tri xora 0, poruka je tačna
   - Ako je samo crveni 1, bit 5 je pogrešan
   - Ako su crveni i plavi 1, bit 4 je pogrešan
   - Ako su sva tri 1, bit 1 je pogrešan




**Mesto greške zavisi samo od toga u kojim krugovima je xor 1.**

# Hamingov (7,4) kod



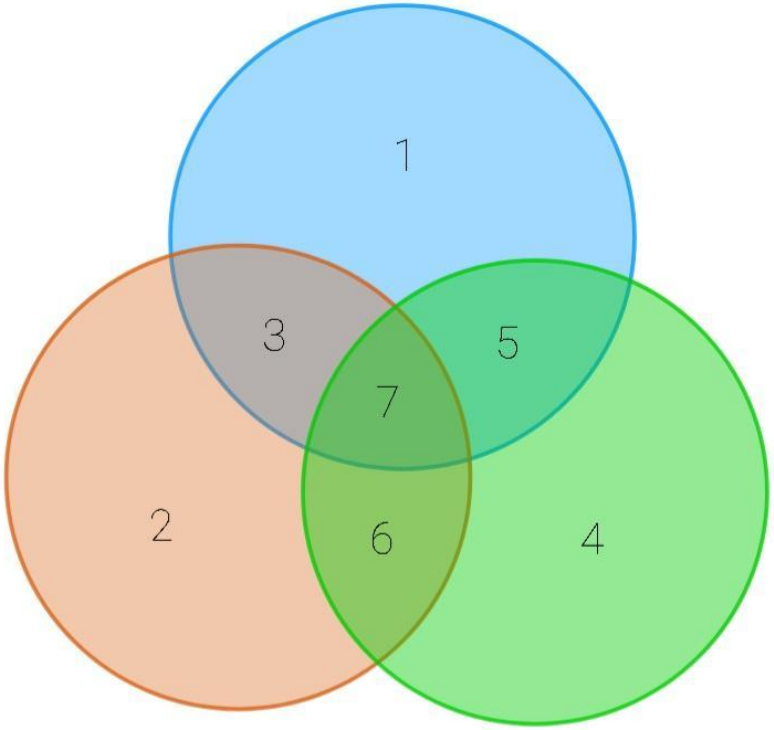



**Hamingova distanca ovog koda je takođe 3**
(verujte mi na reč)

# Hamingov (7,4) kod

„A code is, in practice, only as good as its decoding algorithm”
-neki programer, nekada

Želimo ovu ideju da prilagodimo za predstavljanje matricom.
Renumerišemo svoj Venov dijagram:




1. Pomoćni bitovi su na stepenima dvojke
2. Presek predstavlja sabiranje binarnih brojeva

# Hamingov (7,4) kod

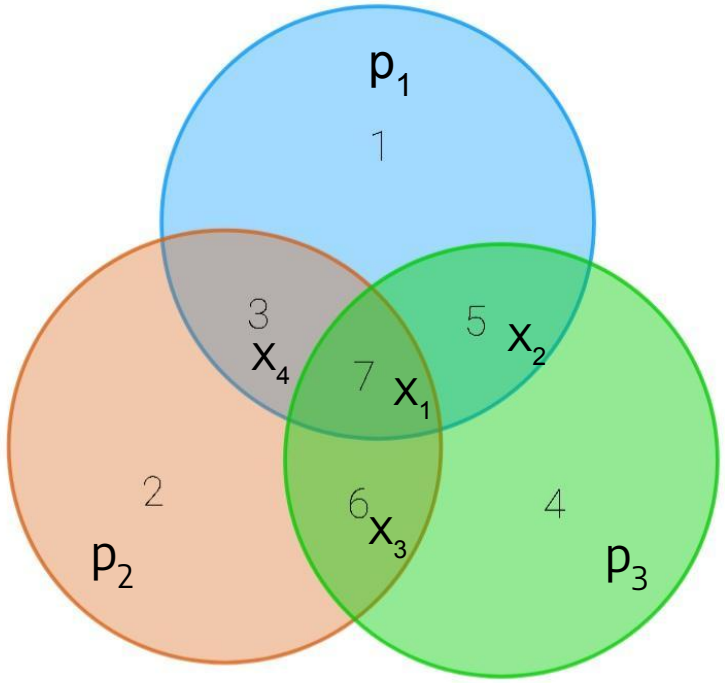
**Kodiranje:** Zelimo da prosledimo poruku $x_1x_2x_3x_4$ sa pomoćnim bitovima $p_1$, $p_2$ i $p_3$.

Šaljemo poruku:

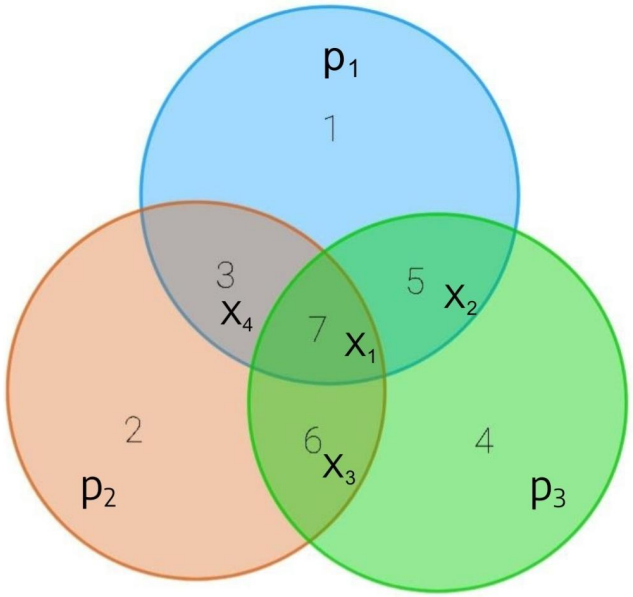
$$\mathbf{p_1p_2x_4p_3x_2x_3x_1}$$

koju dobijamo kao

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} p_1 \\ p_2 \\ x_4 \\ p_3 \\ x_2 \\ x_3 \\ x_1 \end{bmatrix}$$



# Hamingov (7,4) kod




$$\begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ x_4 \\ p_3 \\ x_2 \\ x_3 \\ x_1 \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

**Dekodiranje:**

j-to mesto i-tog reda kaže da li u broju j postoji jedinica na binarnom mestu i

Množenjem dobijamo binaran broj abc koji može biti bilo šta od 0 do 7.

Dekodiramo tako što proveravamo:

- abc=0, nema greške
- abc≠0, greška na mestu abc

# Hamingov (7,4) kod

**Zaključak:**

- I ovaj kod je linearan
- Može da ispravi jednu grešku
- Za poruku od 4 bita je dodato 3 pomoćna bita- znatno manje nego u trostrukom bitu
- Verovatnoća pogrešnog tumačenja je mala ($p^2$ * nešto), jer bi morale barem dve greške da se dese pri prenosu

**Jos par lepih svojstava:**

- Kod linearnih kodova, xorovanjem dve reči iz jezika ostajemo u jeziku
- Princip sličan Hamingovom (7,4) kodu se može primeniti i na duže reči
- Dodavanjem jednog kontrolnog bita parnosti čitave reci, iz Hamingovog (7,4) single error correcting koda dobijamo kod koji je single error correcting i istovremeno double error detecting!

# Hamingov (7,4) kod

**Zašto je ovo zanimljivo?**

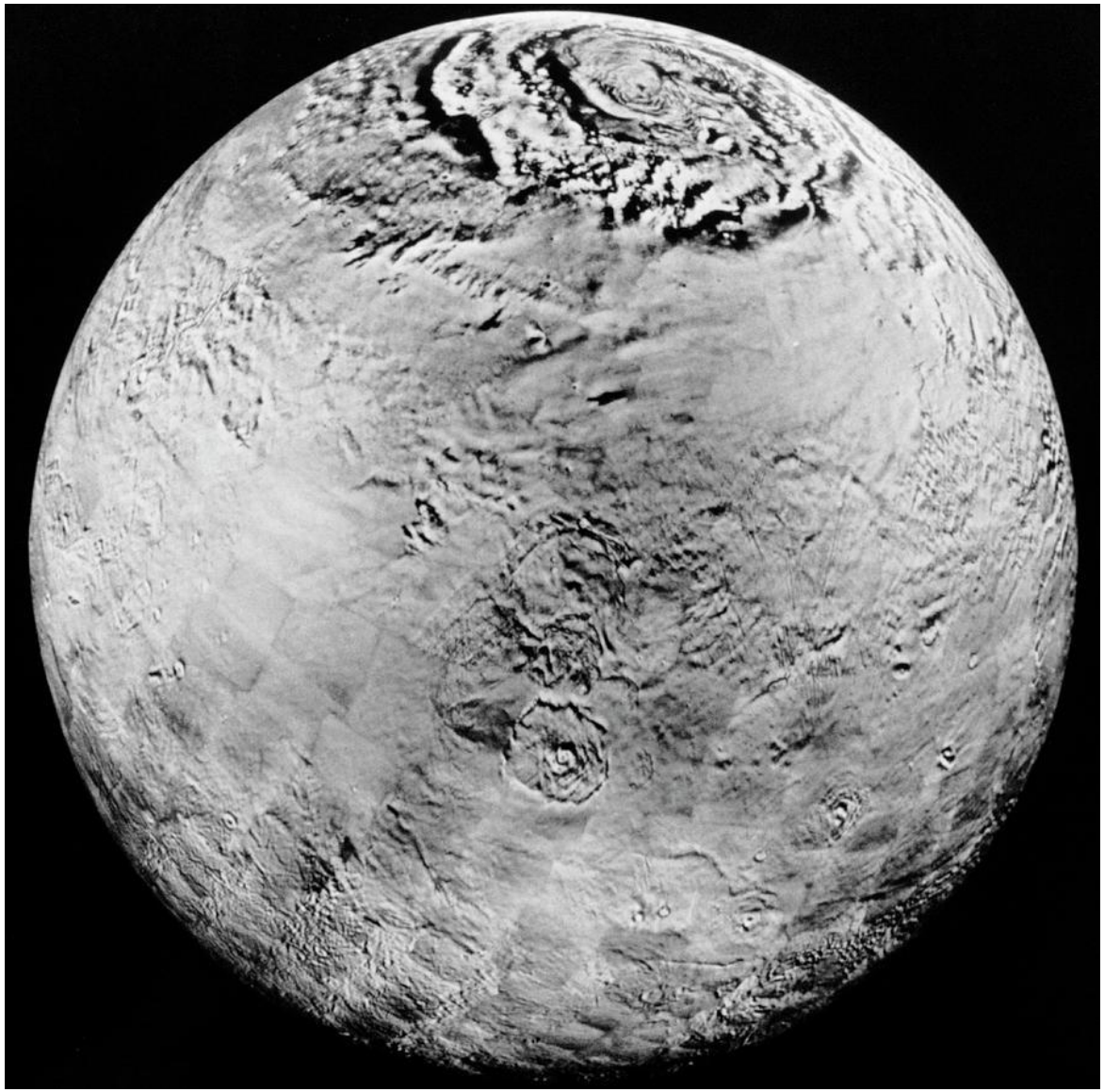
(Teorijski zanimljivo):

Postoje samo dve vrste **savršenih SEC** kodova- cela familija Hamingovih kodova i Golajevi binarni i ternarni kodovi (ostavljamo za čitanje kod kuće)

(Praktično zanimljivo):

Hamingov (64, 57) kod se koristi u **kompjuterskoj memoriji** na niskom nivou- pri čitanju sa silicijumskih čipova

# Mariner 9



# Adamarov kod

Želimo kod sa svojstvima:

- Može ispraviti mnooogo grešaka
- Nije nam toliko bitna dužina koda
- Bilo bi lepo da je linearan
- Brzo se dekodira

Rešenje- Adamarovi kodovi!

1. Dužina poruke: **k**
2. Dužina reči koju šaljemo: $\mathbf{2^k}$
3. Hamingova distanca: $\mathbf{2^{k-1}}$
4. Brzi algoritmi za kodiranje i dekodiranje: **postoje**
5. (Moze biti) **linearan**

# Adamarov kod (kodiranje)

**Primer:** Malo matematičke magije

Ako su k-dimenzioni vektori

$$e_1,\ e_2, \ldots,\ e_{2^k}$$

**Sortirani u leksikografskom poretku**, tada je linearni kod dat sa

$$\begin{bmatrix} \leftarrow & e_1 & \rightarrow \\ \leftarrow & e_2 & \rightarrow \\ \vdots & \vdots & \vdots \\ & & \\ \leftarrow & e_{2^k} & \vdots \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ \vdots \\ x_k \end{bmatrix} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_{2^k-1} \\ m_{2^k} \end{bmatrix}$$

Jedan Adamarov kod.

Matrica za k=3:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

# Adamarov kod(kodiranje)

**U praksi, ovaj kod se ne mora dobijati množenjem poruke matricom!**

**“Adamarove binarne matrice”:**

- Koristimo samo brojeve 0 i 1
- Matrica je kvadratna dimenzije n
- Operacija je xor
- Svake dve kolone se razlikuju na tačno n/2 mesta

Iz poslednje stavke o ovim matricama, njene kolone su zapravo reči udaljene za n/2

**=> Ako znamo da nađemo neku Adamarovu binarnu matricu reda $2^k$, znamo jezik nekog Adamarovog binarnog koda!**

# Adamarov kod(kodiranje)

**Silvesterova konstrukcija Adamarove $2^k$ matrice:**

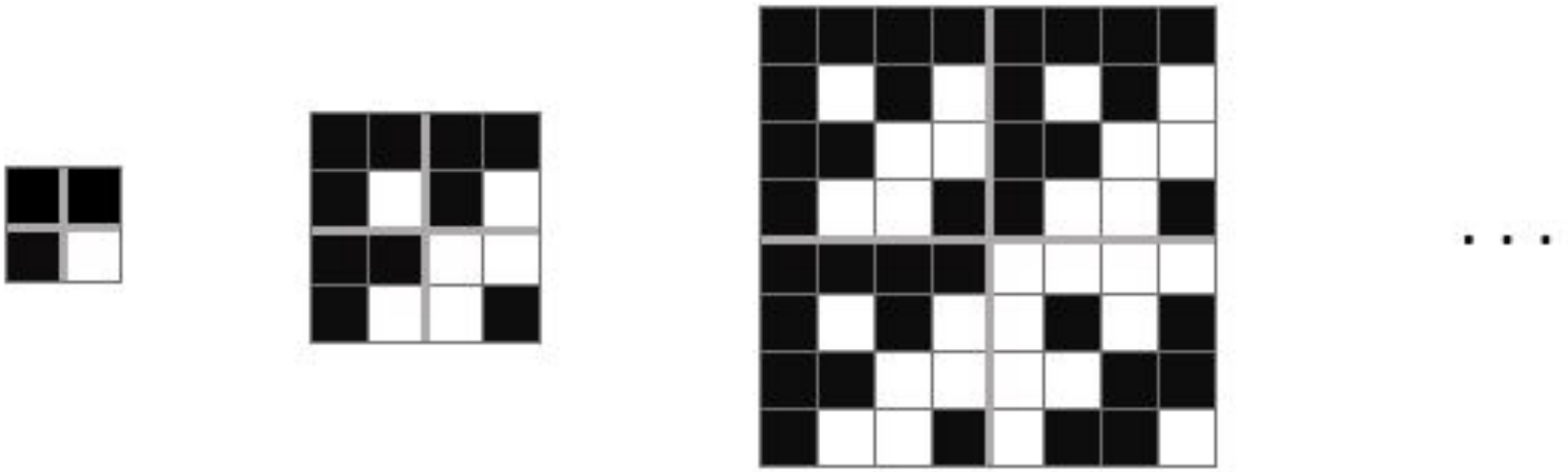
$$H_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

$$H_4 = \begin{bmatrix} 0 \oplus \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} & 0 \oplus \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \\ 0 \oplus \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} & 1 \oplus \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$$H_{2^k} = \begin{bmatrix} 0 \oplus H_{2^{k-1}} & 0 \oplus H_{2^{k-1}} \\ 0 \oplus H_{2^{k-1}} & 1 \oplus H_{2^{k-1}} \end{bmatrix}$$

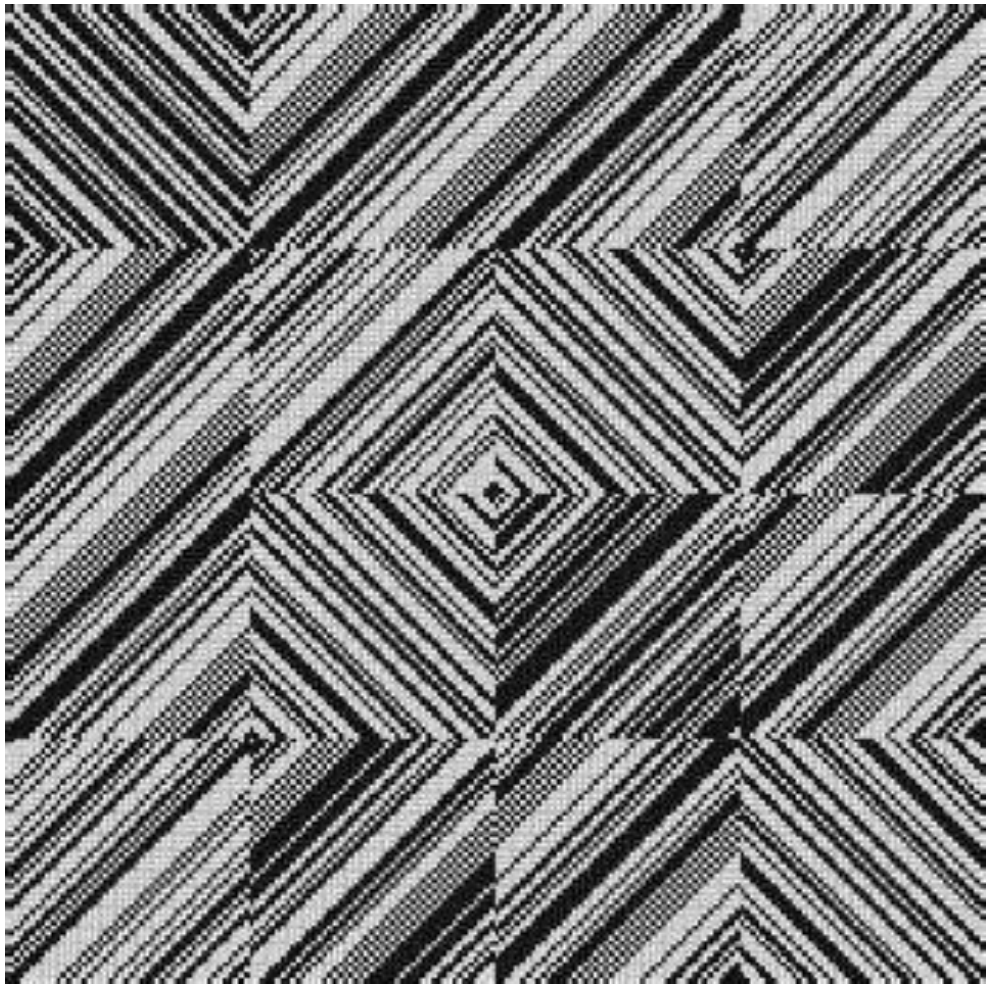
# Adamarov kod(kodiranje)

**Silvesterova konstrukcija Adamarove $2^k$ matrice:**



# Adamarov kod(kodiranje)

**Silvesterova konstrukcija Adamarove $2^k$ matrice:**



# Adamarov kod(dekodiranje)

Dobili smo poruku $x_1 x_2 ... x_{2^k}$, dekodiramo je kao:

**Za svako i od 1 do k:**

1. Izaberimo slučajno $j \in \{1, 2, ..., 2^k\}$
2. Izaberimo $l \in \{1, 2, ..., 2^k\}$ takvo da $j \oplus l = e_i$
3. Postavimo $y_i := x_j \oplus x_l$

**Poruka koju smo pročitali je:** $y_1 y_2 ... y_k$

# Adamarov kod

**Zašto je ovo zanimljivo?**

- Koristan kada šaljemo podatke vrlo nepouzdanim kanalima
- Koristili su ga da šalju slike sa Marsa na Zemlju!
- Može se lokalno dekodirati, tj. jedno slovo nezavisno od ostalih
- Adamarove matrice imaju razne primene i u računarskim naukama i u matematici

**Napomena:** Ovaj kod toliko ljudi koristi za toliko različitih stvari da ima mnogo imena, npr. Walsh codes, Hadamard-Walsh codes, first order Reed-Muller codes. Takođe postoje augmented Hadamard codes koje neki ljudi zovu samo Hadamard codes.

# Dalje čitanje

D J. Baylis- *Error Correcting Codes: A Mathematical Introduction*

# Hvala na pažnji!

Pitanja?