KATEDRA INFORMATIKY
A VÝPOČETNÍ TECHNIKY

FAKULTA APLIKOVANÝCH VĚD
ZÁPADOČESKÉ UNIVERZITY
V PLZNI

Předmět KIV/TI - přednáška 8

Lineární kódy - opravování chyb

Ing. Václav Vais, Ph.D.

vais@kiv.zcu.cz

Opravování chyb podle syndromu

- Operace, kterou provádí dekodér s přijatou *n*-ticí: $\boldsymbol{s} = \boldsymbol{H} \cdot \boldsymbol{w}$
  - je-li $\boldsymbol{s} = \boldsymbol{0}$, je $\boldsymbol{w}$ kódovou značkou
  - v opačném případě je nekódovou kombinací; je detekována chyba
- Nebylo by možné využít hodnotu $\boldsymbol{s}$ k lokalizaci a následné opravě chyby?
- Důsledek linearity kódu:

$$\boldsymbol{s} = \boldsymbol{H} \cdot \boldsymbol{w} = \boldsymbol{H} \cdot (\boldsymbol{v} + \boldsymbol{e}) = \boldsymbol{H} \cdot \boldsymbol{v} + \boldsymbol{H} \cdot \boldsymbol{e} = \boldsymbol{0} + \boldsymbol{H} \cdot \boldsymbol{e} = \boldsymbol{H} \cdot \boldsymbol{e}$$

- **Důsledek 1: U lineárních kódů syndrom závisí pouze na chybovém vektoru a nezávisí na přenášené kódové značce.**

Opravování chyb podle syndromu

- **Důsledek 2: Lineární kód není schopen odhalit chyby s chybovým vektorem ve tvaru kódové značky.**
- Myšlenka: podle syndromu určit chybový vektor.
- Přiřazení syndromů chybovým vektorům: zobrazení $\omega : \mathbf{T}^n \rightarrow \mathbf{T}^{n-k}$, $\omega(\boldsymbol{e}) = \boldsymbol{H} \cdot \boldsymbol{e}$
- Definiční obor tohoto zobrazení má více prvků než obor hodnot, zobrazení není injektivní, inverzní zobrazení ω^{-1} neexistuje $\Longrightarrow$ ze syndromu nelze určit chybový vektor jednoznačně
- Vzhledem k binomickému rozložení p-stí *t*-násobných chyb v *n*-prvkové značce má smysl hledat **nejpravděpodobnější chybový vektor**

Opravování chyb podle syndromu

- Zobrazení ω jednoznačně definuje rozklad $\mathbf{T}^n$ (množiny všech chybových vektorů) na $n - k$ tříd.
- Každému syndromu odpovídá jedna třída rozkladu; jsou v ní všechny chybové vektory, které tento syndrom generují.
- Z každé třídy vybereme chybový vektor $\widehat{\boldsymbol{e}_J}$ s nejmenší Hammingovou vahou (*reprezentanta chybové třídy*).
- Zobrazení ω^{-1} pak lze jednoznačně definovat takto:

Syndrom:	$\boldsymbol{s_1}$	$\boldsymbol{s_2}$	$\boldsymbol{s_3}$	… … … … … … … …	$\boldsymbol{s_{n-k}}$
Reprezentant:	$\widehat{\boldsymbol{e_1}}$	$\widehat{\boldsymbol{e_2}}$	$\widehat{\boldsymbol{e_3}}$	… … … … … … … …	$\widehat{\boldsymbol{e_{n-k}}}$

Opravování chyb podle syndromu

- Opravu chyby pak provedeme takto:

$$\boldsymbol{s} = \boldsymbol{H} \cdot \boldsymbol{w} = \boldsymbol{s_j} \quad , \quad \widehat{\boldsymbol{v}} = \boldsymbol{w} - \omega^{-1}(\boldsymbol{s_j}) = \boldsymbol{w} - \widehat{\boldsymbol{e_j}}$$

$\widehat{\boldsymbol{v}}$ je opravená kódová značka (z praktického pohledu nejpravděpodobněji odesílaná kódová značka)

- Třídou odpovídající nulovému syndromu je množina kódových značek $\mathbf{K}$

  a jejím reprezentantem je nulová značka.

Ilustrační příklad

Binární lineární kód je zadán kontrolní maticí $\boldsymbol{H} = \begin{bmatrix} 0001111 \\ 0110011 \\ 1010101 \end{bmatrix}$

Spočítáme syndromy pro různé chybové vektory – nejprve pro všechny jednoduché chyby:

$\boldsymbol{e} = [1000000]^T$ $\boldsymbol{s} = [001]^T$

$\boldsymbol{e} = [0100000]^T$ $\boldsymbol{s} = [010]^T$

$\boldsymbol{e} = [0010000]^T$ $\boldsymbol{s} = [011]^T$

$\boldsymbol{e} = [0001000]^T$ $\boldsymbol{s} = [100]^T$

$\boldsymbol{e} = [0000100]^T$ $\boldsymbol{s} = [101]^T$

$\boldsymbol{e} = [0000010]^T$ $\boldsymbol{s} = [110]^T$

$\boldsymbol{e} = [0000001]^T$ $\boldsymbol{s} = [111]^T$

Budeme pokračovat syndromy dvojitých chyb:

$\boldsymbol{e} = [1100000]^T$ $\boldsymbol{s} = [011]^T$

$\boldsymbol{e} = [1010000]^T$ $\boldsymbol{s} = [010]^T$

.....

Syndromy generované jednoduchými chybami jsou jedinečné a tvoří množinu všech nenulových syndromů.

!POZOR! NENÍ TO OBECNOU VLASTNOSTÍ LINEÁRNÍCH KÓDŮ (JEN TÉTO MATICE H)

Ilustrační příklad

- Reprezentanty chybových tříd jsou nulový vektor (pro třídu odpovídající nulovému syndromu) a všechny chybové vektory s Hammingovou vahou 1 (pro třídy odpovídající jednotlivých nenulovým syndromům).

Konec příkladu, zobecnění:

- Uvažujeme-li binární lineární *(n,k)* kód, existuje 2^{n-k} různých syndromů.
- Počet syndromů musí postačit k rozlišení všech přípustných chybových vektorů.
- Pro opravu *t*-násobných chyb existuje $1 + n + \binom{n}{2} + \cdots + \binom{n}{t}$ přípustných chybových vektorů.

Počet zabezpečovacích prvků *versus* násobnost opravovaných chyb

- Konkretizace pro binární lineární *(n,k)* kód:
  - k rozlišení (a opravě) **jednoduchých** chyb potřebujeme
    - jeden nulový syndrom (pro nulový chybový vektor)
    - *n* dalších různých syndromů pro rozlišení *n* chybových vektorů s vahou 1
    - musí tedy platit $2^{\mathrm{n-k}} \geq 1 + n$
  - k rozlišení (a opravě) **dvojitých** chyb potřebujeme
    - jeden nulový syndrom (pro nulový chybový vektor)
    - *n* dalších různých syndromů pro rozlišení *n* chybových vektorů s vahou 1
    - $\binom{n}{2}$ dalších různých syndromů pro rozlišení chybových vektorů s vahou 2
    - musí tedy platit $2^{\mathrm{n-k}} \geq 1 + n + \binom{n}{2}$

Počet zabezpečovacích prvků *versus* násobnost opravovaných chyb

- Zobecnění pro binární lineární *(n,k)* kód:
  - k lokalizaci (a opravě) ***t*-násobných** chyb musí platit

$$2^{n-k} \geq 1 + n + \binom{n}{2} + \cdots + \binom{n}{t}$$

- **Počet kontrolních prvků je určen počtem informačních prvků a násobností opravovaných chyb.**

Hammingovy kódy

- Hammingovy kódy jsou kódy, které opravují jednoduché chyby a přitom mají minimální redundanci ze všech kódů pro opravu jednoduchých chyb.
- Předpokládejme, že při přenosu došlo k jednoduché chybě reprezentované chybovým vektorem $\boldsymbol{e_i} = [0 \ldots 010 \ldots 0]^{\boldsymbol{T}}$ s jedničkou v *i*-tém znaku.

  Pak platí $\boldsymbol{s} = \boldsymbol{H} \cdot \boldsymbol{w} = \boldsymbol{H} \cdot (\boldsymbol{v} + \boldsymbol{e_i}) = \boldsymbol{H} \cdot \boldsymbol{v} + \boldsymbol{H} \cdot \boldsymbol{e_i} = \boldsymbol{0} + \boldsymbol{H} \cdot \boldsymbol{e_i} = \boldsymbol{H}_{*,\boldsymbol{i}}$

  kde $\boldsymbol{H}_{*,\boldsymbol{i}}$ představuje *i*-tý sloupec kontrolní matice $\boldsymbol{H}$.
- V případě jednoduché chyby v *i*-tém znaku tedy syndrom odpovídá *i*-tému sloupci kontrolní matice.

Hammingovy kódy

- Důsledek - vlastnosti kontrolní matice kódu pro opravu jednoduchých chyb:

nesmí obsahovat nulový sloupec $\boldsymbol{H}_{*,i} \neq \boldsymbol{0} \quad \forall\, i = 1,2,\dots,n$

nesmí obsahovat stejné sloupce $\boldsymbol{H}_{*,i} \neq \boldsymbol{H}_{*,j} \quad \forall\, i,j = 1,2,\dots,n,\ \ i \neq j$

- Kontrolní matice Hammingova kódu obsahuje ve sloupcích **všechna nenulová** slova dané délky a žádné z nich se neopakuje.

- Poznámka: pokud by matice neobsahovala všechna nenulová slova, ale splňovala by dvě výše uvedené vlastnosti, kód by opravoval jednoduché chyby, ale neměl by minimální redundanci.

- Redundance binárního lineárního (*n, k*) kódu: $\rho = 1 - \frac{log_2 2^k}{log_2 2^n} = 1 - \frac{k}{n}$

Hammingovy kódy

- Při daném počtu kontrolních znaků $r = n - k$ pro binární Hammingův kód platí:
  - počet různých syndromů: 2^r
  - počet sloupců matice $\boldsymbol{H}$: $2^r - 1$ (musíme „odpočítat“ nulové slovo)
  - počet znaků v kódové značce: $n = 2^r - 1$ (jako počet sloupců kontrolní matice)
  - počet informačních znaků ve značce: $k = n - r$
  - musí tedy platit rovnost $2^r = k + r + 1$
- Parametry binárních (n, k) Hammingových kódů (včetně *informačního poměru k/n*):

r	2^r	n	k	(n,k)	k/n
3	8	7	4	(7,4)	0,57143
4	16	15	11	(15,11)	0,73333
5	32	31	26	(31.26)	0,83871
6	64	63	57	(63,57)	0,90476
.	.	.	.	.	.
.	.	.	.	.	atd.

S rostoucím r roste efektivita kódu.

Hammingovy kódy - příklady

- Příklady kontrolních matic Hammingových kódů (7,4)

$$\boldsymbol{H} = \begin{bmatrix} 0001111 \\ 0110011 \\ 1010101 \end{bmatrix}$$

nesystematický kód (se zajímavou vlastností):
v případě jednoduché chyby je syndrom binárním vyjádřením pozice chyby

$$\boldsymbol{H} = \begin{bmatrix} 0111\ 100 \\ 1011\ 010 \\ 1101\ 001 \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} 1000\ 011 \\ 0100\ 101 \\ 0010\ 110 \\ 0001\ 111 \end{bmatrix}$$

systematický kód

$$\boldsymbol{H} = \begin{bmatrix} 1110\ 100 \\ 0111\ 010 \\ 1101\ 001 \end{bmatrix} \qquad \boldsymbol{G} = \begin{bmatrix} 1000\ 101 \\ 0100\ 111 \\ 0010\ 110 \\ 0001\ 011 \end{bmatrix}$$

systematický kód
(navíc ještě cyklický)

na pořadí sloupců v matici $\boldsymbol{H}$ nezáleží

Hammingovy kódy - konstrukce

- Pro zadaný počet kontrolních prvků r z rovnosti $2^r = k + r + 1$ spočítáme k
- Systematická generující matice $\boldsymbol{G}$ má k řádků, $k + r$ sloupců a tvar $\boldsymbol{G} = [\boldsymbol{I_k} \mid \boldsymbol{B}]$
- Levá submatice $\boldsymbol{I_k}$ je jednotková matice řádu k
- Do řádků pravé submatice $\boldsymbol{B}$ umístíme všechny r-bitové řetězce obsahující **alespoň dvě jedničky** (tedy binární rozvoje čísel, které nejsou mocninami dvou; žádný n-k znakový řetězec se nemůže opakovat, žádný nemůže být použit vícekrát).
- Na pořadí v řádků v matici $\boldsymbol{B}$ nezáleží. Dobrý zvyk – psát je ve stoupajícím pořadí.
- Systematická kontrolní matice $\boldsymbol{H}$ má r řádků, $k + r$ sloupců
- Kontrolní matici $\boldsymbol{H}$ pak vytvoříme jako $\boldsymbol{H} = [-\boldsymbol{B}^T \mid \boldsymbol{I}_{n-k}]$.

Hammingovy kódy – jaká je d_0 ?

- Minimální Hammingova vzdálenost v lineárním kódu je rovna minimální Hammingově váze nenulové značky.
- Kód má 2^k kódových značek, k z nich tvoří řádky generující matice $\boldsymbol{G}$.
- Každý řádek $\boldsymbol{G}$ obsahuje alespoň tři jedničky (jednu v části $\boldsymbol{I_k}$, přinejmenším dvě v části $\boldsymbol{B}$).
- Součet libovolných dvou řádků $\boldsymbol{G}$ obsahuje dvě jedničky v části $\boldsymbol{I_k}$ a přinejmenším jednu jedničku v části $\boldsymbol{B}$ (protože v $\boldsymbol{B}$ nemohou být dva stejné řádky).
- Součet libovolných tří řádků $\boldsymbol{G}$ obsahuje tři jedničky v části $\boldsymbol{I_k}$.
- V Hammingově kódu tedy neexistuje nenulová značka s menším počtem jedniček než tři $\Rightarrow$ $d_0 = 3$.

Zkracování Hammingových kódů

- Hammingovy kódy umožňují kódovat jen informační části délky k , která pro nějaké r vyhovuje rovnici $2^r = k + r + 1$ (k = 4, 11, 26, 57, ... viz tabulka na jednom z předchozích slajdů) .
- Pokud máme zkonstruovat kód pro opravu jednoduchých chyb pro jinou hodnotu k , provedeme *zkrácení* (Hammingova) *kódu*:
  - najdeme počet zabezpečovacích znaků r vyhovující nerovnici $2^r \geq k + r + 1$
  - zkonstruujeme matice $\boldsymbol{G}$ a $\boldsymbol{H}$ Hammingova kódu pro r zabezpečovacích znaků
  - z matic $\boldsymbol{G}$ a $\boldsymbol{H}$ vypustíme řádky a sloupce odpovídající nepotřebným informačním prvkům (z matice $\boldsymbol{G}$ řádky a sloupce, z matice $\boldsymbol{H}$ sloupce)

Zkracování Hammingových kódů

- **Příklad:** Najděte systematický kód pro kódování šestiznakových informačních částí, který bude opravovat jednoduché chyby.
- **Řešení:**

$$2^r \geq k + r + 1 \quad \Rightarrow \quad 2^r \geq 6 + r + 1 \Rightarrow \quad 2^r \geq r + 7 \quad \Rightarrow r = 4 \quad \text{(nejmenší vyhovující } r\text{)}$$

$$\boldsymbol{G} = \begin{bmatrix} 100000\,00000\ 0011 \\ 010000\,00000\ 0101 \\ 001000\,00000\ 0110 \\ 000100\,00000\ 0111 \\ 000010\,00000\ 1001 \\ 000001\,00000\ 1010 \\ 00000010000\ 1011 \\ 00000001000\ 1100 \\ 00000000100\ 1101 \\ 00000000010\ 1110 \\ 00000000001\ 1111 \end{bmatrix} \qquad \boldsymbol{H} = \begin{bmatrix} 000011\,11111\ 1000 \\ 011100\,01111\ 0100 \\ 101101\,10011\ 0010 \\ 110110\,10101\ 0001 \end{bmatrix}$$

$$\boldsymbol{G} = \begin{bmatrix} 100000\ 0011 \\ 010000\ 0101 \\ 001000\ 0110 \\ 000100\ 0111 \\ 000010\ 1001 \\ 000001\ 1010 \end{bmatrix} \qquad \boldsymbol{H} = \begin{bmatrix} 000011\ 1000 \\ 011100\ 0100 \\ 101101\ 0010 \\ 110110\ 0001 \end{bmatrix}$$

modře podbarvené řady vypustíme

Rozšiřování kódů

- Lineární kód **s lichou minimální Hammingovou vzdáleností** d_0 lze *rozšířit* přidáním dalšího zabezpečovacího prvku – celkové kontroly parity (doplnění 0 nebo 1 tak, aby byl počet jedniček ve značce sudý).
- Minimální Hammigova vzdálenost kódu po rozšíření pak bude $d_0 + 1$.
- **POZOR ! Rozšíření kódu se sudou minimální Hammingovou vzdáleností tuto vzdálenost nezvětší.**
- Zabezpečovací vlastnosti kódu se tím zvětší o možnost detekce další násobnosti chyb.

Rozšiřování kódů

- **Příklad:** Rozšíření Hammingova kódu (*7,4*).
- **Řešení:**

$$\boldsymbol{G} = \begin{bmatrix} 1000\ 011\ 1 \\ 0100\ 101\ 1 \\ 0010\ 110\ 1 \\ 0001\ 111\ 0 \end{bmatrix} \qquad \boldsymbol{H_1} = \begin{bmatrix} 0111\ 1000 \\ 1011\ 0100 \\ 1101\ 0010 \\ 1110\ 0001 \end{bmatrix} \qquad \boldsymbol{H_2} = \begin{bmatrix} 0111\ 100\ 0 \\ 1011\ 010\ 0 \\ 1101\ 001\ 0 \\ 1111\ 111\ 1 \end{bmatrix}$$

žlutě podbarvené – standardní zabezpečovací prvky Hammingova kódu (*7,4*)

modře podbarvené – zabezpečovací prvky parity

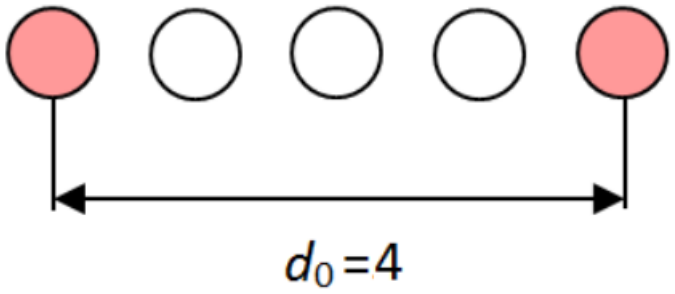
matice $\boldsymbol{H_1}$ - kontrolní matice vytvořená na základě vztahu $\boldsymbol{H} = [-\boldsymbol{B}^T \mid \boldsymbol{I}_{n-k}]$

matice $\boldsymbol{H_2}$ - kontrolní matice vytvořená na základě tvorby kontrolních znaků

Rozšiřování kódů

- **Příklad:** Rozšíření Hammingova kódu (*7,4*).

  $d_0 = 3 + 1 = 4$



- Rozšířený Hammingův kód je tedy schopen
  - jednoduché chyby opravovat **a současně** dvojité chyby detekovat

    nebo

  - trojité chyby detekovat

Perfektní kódy

- Lineární kód je *perfektní pro opravu t-násobných chyb* jestliže všechna chybová slova váhy $\leq$ t jsou reprezentanty chybových tříd a neexistují žádné jiné chybové třídy.
- Jinak řečeno:
  - každému syndromu přísluší jedinečný chybový vektor váhy $\leq$ t
  - každému chybovému vektoru váhy $\leq$ t přísluší jedinečný syndrom
- Perfektní kódy mají ze všech kódů pro opravu t-násobných chyb minimální redundanci.
- Existuje jen málo perfektních kódů:
  - Hammingovy kódy pro opravu jednoduchých chyb
  - Golayovy kódy pro opravu trojnásobných chyb
  - opakovací kódy s délkou značky $n = 2 \cdot t + 1$ pro opravu t-násobných chyb

Oprava chyb u opakovacích kódů

- Předpokládejme systematický opakovací kód pro přenos čtyřprvkových informačních částí (k = 4) s pro opravu dvojitých chyb (t = 2) .
- Potřebná minimální vzdálenost: $d_0 = 2 \cdot t + 1 = 5$
- Potřebný počet opakování = 5 $\Rightarrow$ délka značky $n = 5 \cdot k = 20$
- Generující matice:

$$\boldsymbol{G} = \begin{bmatrix} 1&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&1&1&1&1&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&1&1&1&1&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1 \end{bmatrix}$$

Oprava chyb u opakovacích kódů

$$\boldsymbol{G} = \begin{bmatrix} 1\,0\,0\,0 & 1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,1\,0\,0 & 0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,1\,0 & 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0 \\ 0\,0\,0\,1 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1 \end{bmatrix}$$

- Kontrolní rovnice:

(pro prvek u_1) $v_1 = v_5$, $v_1 = v_6$, $v_1 = v_7$, $v_1 = v_8$
(pro prvek u_2) $v_2 = v_9$, $v_2 = v_{10}$, $v_2 = v_{11}$, $v_2 = v_{12}$
(pro prvek u_3) $v_3 = v_{13}$, $v_3 = v_{14}$, $v_3 = v_{15}$, $v_3 = v_{16}$
(pro prvek u_4) $v_4 = v_{17}$, $v_4 = v_{18}$, $v_4 = v_{19}$, $v_4 = v_{20}$

Oprava chyb u opakovacích kódů

- Kontrolní rovnice (v nehomogenním tvaru):

(pro prvek u_1) $v_1 = v_5$, $v_1 = v_6$, $v_1 = v_7$, $v_1 = v_8$
(pro prvek u_2) $v_2 = v_9$, $v_2 = v_{10}$, $v_2 = v_{11}$, $v_2 = v_{12}$
(pro prvek u_3) $v_3 = v_{13}$, $v_3 = v_{14}$, $v_3 = v_{15}$, $v_3 = v_{16}$
(pro prvek u_4) $v_4 = v_{17}$, $v_4 = v_{18}$, $v_4 = v_{19}$, $v_4 = v_{20}$

- Kontrolní rovnice formálně přepíšeme a doplníme ($\tilde{u}$ představuje u po opravě)

(pro prvek u_1) $\tilde{u}_1 = w_5$, $\tilde{u}_1 = w_6$, $\tilde{u}_1 = w_7$, $\tilde{u}_1 = w_8$, $\tilde{u}_1 = w_1$
(pro prvek u_2) $\tilde{u}_2 = w_9$, $\tilde{u}_2 = w_{10}$, $\tilde{u}_2 = w_{11}$, $\tilde{u}_2 = w_{12}$, $\tilde{u}_2 = w_2$
(pro prvek u_3) $\tilde{u}_3 = w_{13}$, $\tilde{u}_3 = w_{14}$, $\tilde{u}_3 = w_{15}$, $\tilde{u}_3 = w_{16}$, $\tilde{u}_3 = w_3$
(pro prvek u_4) $\tilde{u}_4 = w_{17}$, $\tilde{u}_4 = w_{18}$, $\tilde{u}_4 = w_{19}$, $\tilde{u}_4 = w_{20}$, $\tilde{u}_4 = w_4$

- Tyto rovnice již nebudeme chápat jako soustavu lineárních rovnic, u nichž hledáme řešení metodami LA. Půjde o to, určit $\tilde{u}$ i v případě chyb při přenosu.

Oprava chyb u opakovacích kódů

(pro prvek u_1) $\tilde{u}_1 = w_5$, $\tilde{u}_1 = w_6$, $\tilde{u}_1 = w_7$, $\tilde{u}_1 = w_8$, $\tilde{u}_1 = w_1$
(pro prvek u_2) $\tilde{u}_2 = w_9$, $\tilde{u}_2 = w_{10}$, $\tilde{u}_2 = w_{11}$, $\tilde{u}_2 = w_{12}$, $\tilde{u}_2 = w_2$
(pro prvek u_3) $\tilde{u}_3 = w_{13}$, $\tilde{u}_3 = w_{14}$, $\tilde{u}_3 = w_{15}$, $\tilde{u}_3 = w_{16}$, $\tilde{u}_3 = w_3$
(pro prvek u_4) $\tilde{u}_4 = w_{17}$, $\tilde{u}_4 = w_{18}$, $\tilde{u}_4 = w_{19}$, $\tilde{u}_4 = w_{20}$, $\tilde{u}_4 = w_4$

- Z pohledu LA má tato soustava řešení pouze při bezchybném přenosu celé značky, tedy jestliže platí $w_i = v_i \quad \forall\, i = 1,2,\dots,20$, kde *w* představuje přijatou *n*-tici. Pak řešení $\tilde{u}$ existuje a platí $\tilde{u} = u$

- Naše řešení: z této „soustavy“ „vypočítáme“ všechna $\tilde{u}_i$ na majoritním principu:
  - pro každé $\tilde{u}_i$ máme $2 \cdot t + 1$ rovnic
  - nedojde-li ve skupině bitů, v nichž je přenášen prvek u_i k více než *t*-násobné chybě, dá nadpoloviční většina rovnic pro $\tilde{u}_i$ vždy správný výsledek

Golayovy kódy

- *Golayův kód* $\boldsymbol{G_{23}}$ je perfektní binární kód pro opravu trojnásobných chyb.
- Délka značky je 23 znaků, generující matice má rozměr 12/23:

$$\boldsymbol{G_{23}} = \begin{bmatrix} 100000000000 & 11011100010 \\ 010000000000 & 01101110001 \\ 001000000000 & 10110111000 \\ 000100000000 & 01011011100 \\ 000010000000 & 00101101110 \\ 000001000000 & 00010110111 \\ 000000100000 & 10001011011 \\ 000000010000 & 11000101101 \\ 000000001000 & 11100010110 \\ 000000000100 & 01110001011 \\ 000000000010 & 10111000101 \\ 000000000001 & 11111111111 \end{bmatrix} = \left[\boldsymbol{I_{12}} \mid \frac{\boldsymbol{B}}{11\ldots.1}\right]$$

0123……10

čtvercová submatice $\boldsymbol{B}$ řádu 11 vznikla cyklickými posuvy první-ho řádku

jedničky v prvním řádku jsou v pozicích kvadrátů prvků tělesa $\mathbf{Z}_{11}$ (0, 1, 3, 4, 5, 9)

Golayovy kódy

- Golayův kód $\boldsymbol{G_{23}}$ má minimální Hammingovu vzdálenost $d_0 = 7$.
- Každý chybový vektor váhy $t \leq 3$ generuje jiný syndrom $\Rightarrow$ kód opravuje trojnásobné chyby.
- Počet různých syndromů v kódu $\boldsymbol{G_{23}}$ je $2^r = 2^{11}$.
- Počet různých chybových vektorů váhy $t \leq 3$ je

$$1 + 23 + \binom{23}{2} + \binom{23}{3} = 1 + 23 + \frac{23!}{21!\cdot 2!} + \frac{23!}{20!\cdot 3!} = 1 + 23 + \frac{23\cdot 22}{2\cdot 1} + \frac{23\cdot 22\cdot 21}{3\cdot 2\cdot 1} =$$

$$= 1 + 23 + 23 \cdot 11 + 23 \cdot 11 \cdot 7 = 1 + 23 + 253 + 1771 = 2048 = 2^{11}$$

- Existuje vzájemně jednoznačné přiřazení mezi syndromy a chybovými vektory váhy $t \leq 3$
- Golayův kód $\boldsymbol{G_{23}}$ je perfektní pro opravu trojnásobných chyb.

Golayovy kódy

- Golayův kód G_{24} vznikne rozšířením kódu G_{23} o celkovou kontrolu parity.
- Délka značky je 24 znaků, generující matice má rozměr 12/24:

$$G_{24} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0
\end{bmatrix}$$

Golayovy kódy

- Vlastnosti matice $\boldsymbol{G_{24}}$:
  - v každém řádku s výjimkou posledního je 8 jedniček, v posledním je jich 12
  - každá lineární kombinace řádků $\boldsymbol{G_{24}}$ má nejméně 8 jedniček
  - minimální Hammingova vzdálenost kódu je tedy $d_0 = 8$
  - pro každé dvě kódové značky u, v platí $\sum_{i=1}^{24} u_i \cdot v_i = 0$, kód je tedy ortogonálním doplňkem „sebe samého“, je to *samoduální kód*
  - jako kontrolní matici lze tedy použít matici generující $\boldsymbol{H_{24}} = \boldsymbol{G_{24}}$
- Použití kódu $\boldsymbol{G_{24}}$:
  - kódování přenosu barevných fotografií Jupitera a Saturnu (sondy Voyager 1 a 2)
  - součást současných komunikačních standardů U.S. Army i civilního letectví

Reed-Mullerovy kódy

- R – M kódy představují třídu nesystematických kódů, které umožňují opravy předem zadaného počtu chyb.
- Současně umožňují relativně jednoduché dekódování.
- Vytvoření generující matice R – M kódu:

Pro libovolná čísla $m \epsilon N$ a $r \epsilon N_0$, kdy $0 \leq r \leq m$, lze zkonstruovat matici o $n = 2^m$ sloupcích jako blokovou matici

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G_0} \\ \boldsymbol{G_1} \\ \vdots \\ \boldsymbol{G_r} \end{bmatrix}$$

$\boldsymbol{G_0} = [1\ 1\ \ldots 1]$ je matice typu $1/n$

$$\boldsymbol{G_1} = \begin{bmatrix} 0\ 0 \ldots 0\ 1 \\ \vdots \\ 0\ 0 \ldots 1\ 1 \\ 0\ 1 \ldots 1\ 1 \end{bmatrix}$$

je matice typu m/n, obsahující popořadě všechny sloupce z 0 a 1

$\boldsymbol{G_l}$, kde $2 \leq l \leq r$ jsou matice, jejichž řádky obsahují všechny možné součiny l řádků matice $\boldsymbol{G_1}$

Reed-Mullerovy kódy – ilustrační příklad

Vytvoření matice $\boldsymbol{G}$ pro $m = 3$ a $r = 3$:

$$\boldsymbol{G_0} = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$$

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G_0} \\ \boldsymbol{G_1} \\ \vdots \\ \boldsymbol{G_r} \end{bmatrix}$$

$$\boldsymbol{G_1} = \begin{bmatrix} 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1 \\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1 \\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1 \end{bmatrix}$$

$$\boldsymbol{G_2} = \begin{bmatrix} 0\ 0\ 0\ 0\ 0\ 0\ 1\ 1 \\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1 \\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 1 \end{bmatrix} \quad \begin{matrix} 1.\text{ř.}\,\boldsymbol{G_1} \cdot 2.\text{ř.}\,\boldsymbol{G_1} \\ 2.\text{ř.}\,\boldsymbol{G_1} \cdot 3.\text{ř.}\,\boldsymbol{G_1} \\ 1.\text{ř.}\,\boldsymbol{G_1} \cdot 3.\text{ř.}\,\boldsymbol{G_1} \end{matrix}$$

(součiny řádků jsou „bit po bitu“)

$$\boldsymbol{G_3} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 1] \quad 1.\text{ř.}\,\boldsymbol{G_1} \cdot 2.\text{ř.}\,\boldsymbol{G_1} \cdot 3.\text{ř.}\,\boldsymbol{G_1}$$

Reed-Mullerovy kódy – ilustrační příklad

Vytvoření matice $\boldsymbol{G}$ pro $m = 3$ a $r = 3$:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G_0} \\ \boldsymbol{G_1} \\ \boldsymbol{G_2} \\ \boldsymbol{G_3} \end{bmatrix} = \left[\begin{array}{cccccccc} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

v každé matici $\boldsymbol{G_l}$ mají všechny řádky stejný počet jedniček, a to 2^{m-l}

řádky matice $\boldsymbol{G}$ jsou lineárně nezávislé

obecné značení R-M kódů je $K_{r,m}$

Počet informačních znaků kódu je $k = 1 + m + \binom{m}{2} + \cdots + \binom{m}{r} = \sum_{l=0}^{r} \binom{m}{l}$

Reed-Mullerovy kódy – ilustrační příklad

Vytvoření matice ***G*** ($m = 3$, $r = 2$, $n = 2^m = 8$) :

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G_0} \\ \boldsymbol{G_1} \\ \boldsymbol{G_2} \end{bmatrix} = \begin{bmatrix} 1\,1\,1\,1\,1\,1\,1\,1 \\ \hline 0\,0\,0\,0\,1\,1\,1\,1 \\ 0\,0\,1\,1\,0\,0\,1\,1 \\ 0\,1\,0\,1\,0\,1\,0\,1 \\ \hline 0\,0\,0\,0\,0\,0\,1\,1 \\ 0\,0\,0\,1\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,1\,0\,1 \end{bmatrix}$$

řádkovými úpravami matice bychom zjistili, že se jedná o generující matici kódu ekvivalentního s kódem celkové kontroly parity

$$k = 1 + m + \binom{m}{2} = 1 + 3 + 3 = 7$$

Reed-Mullerovy kódy – ilustrační příklad

Vytvoření matice ***G*** ($m = 3$, $r = 1$, $n = 2^m = 8$) :

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G_0} \\ \boldsymbol{G_1} \end{bmatrix} = \begin{bmatrix} 1\,1\,1\,1\,1\,1\,1\,1 \\ \hline 0\,0\,0\,0\,1\,1\,1\,1 \\ 0\,0\,1\,1\,0\,0\,1\,1 \\ 0\,1\,0\,1\,0\,1\,0\,1 \end{bmatrix}$$

jedná o generující matici (nesystematického) kódu ekvivalentního s rozšířeným Hammingovým kódem

$$k = 1 + m = 1 + 3 = 4$$

Vytvoření matice ***G*** ($m = 3$, $r = 0$, $n = 2^m = 8$) :

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G_0} \end{bmatrix} = [1\,1\,1\,1\,1\,1\,1\,1]$$

opakovací kód s délkou značky $n = 8$, opravuje chyby s násobností $t \leq (n-2)/2$, tedy $t = 3$

$$k = 1$$

Reed-Mullerovy kódy – obecně

.

- Pro R-M kódy $K_{r,m}$ obecně platí:

  $K_{0,m}$ je opakovací kód

  $K_{m-2,m}$ je rozšířený Hammingův kód

  $K_{m-1,m}$ je paritní kód

  $K_{m,m}$ je Z_2^n , kde $n = 2^m$

- Dekódování je „relativně jednoduché“ (viz Vais: Teoretická informatika 2. část)

Ekvivalentní kódy

- Dva blokové kódy K a K' délky n jsou *ekvivalentní* , jestliže existuje permutace $[\pi_1, \pi_2, \ldots, \pi_n]$ čísel 1, 2, ….., n taková, že platí

  $\forall\, v_1, v_2, \ldots, v_n \in T^n : \quad v_1, v_2, \ldots, v_n \in K \iff v_{\pi_1}, v_{\pi_2}, \ldots, v_{\pi_n} \in K'$.
- Jinými slovy:
  - existuje vzájemně jednoznačné přiřazení mezi kódovými značkami obou kódů
  - odpovídající si značky obou kódů se navzájem liší jen záměnou pořadí prvků. U všech odpovídajících si dvojic je to pořadí zaměněno stejně.

Ekvivalentní kódy

- Ke každému lineárnímu kódu existuje ekvivalentní systematický kód.
- Jak z generující matice $\boldsymbol{G}$ zkonstruovat generující matici $\boldsymbol{G}'$ ekvivalentního systematického kódu?
  - $\boldsymbol{G}$ má *k* lineárně nezávislých řádků, *n* > *k*, matice má tedy i *k* lineárně nezávislých sloupců (vlastnosti matic, viz LA)
  - pokud je lineárně nezávislých prvních *k* sloupců matice $\boldsymbol{G}$, vytvoříme matici $\boldsymbol{G}'$ ekvivalentními řádkovými úpravami matice $\boldsymbol{G}$; získáme tak jinou bázi prostoru kódových značek generovaných maticí $\boldsymbol{G}$, tedy jinou bázi téhož kódu

Ekvivalentní kódy

- pokud prvních *k* sloupců matice $\boldsymbol{G}$ není lineárně nezávislých, provedeme takovou permutaci sloupců matice $\boldsymbol{G}$, abychom dostali nezávislé sloupce do prvních *k* pozic; pak vytvoříme systematickou matici $\boldsymbol{G}'$ ekvivalentními řádkovými úpravami (sloupcově permutované) matice $\boldsymbol{G}$; v tomto případě budou množiny značek generované maticemi $\boldsymbol{G}$ a $\boldsymbol{G}'$ různé, ale bude mezi nimi existovat vzájemně jednoznačné zobrazení dané provedenou záměnou sloupců matice $\boldsymbol{G}$, kódy tedy budou ekvivalentní