

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

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

$$(\mathbf{v} \in \mathbf{K} \Leftrightarrow \mathbf{H} \cdot \mathbf{v} = \mathbf{0})$$

- **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 $\omega : \mathbf{T}^n \rightarrow \mathbf{T}^{n-k}$, $\omega(\mathbf{e}) = \mathbf{H} \cdot \mathbf{e}$
- Definiční obor tohoto zobrazení má více prvků než obor hodnot, zobrazení není injektivní, inverzní zobrazení ω^{-1} neexistuje \Rightarrow 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 \hat{e}_j s nejmenší Hammingovou vahou (*reprezentanta chybové třídy*).
- Zobrazení ω^{-1} pak lze jednoznačně definovat takto:

Syndrom:	\mathbf{s}_1	\mathbf{s}_2	\mathbf{s}_3	\mathbf{s}_{n-k}
Reprezentant:	\hat{e}_1	\hat{e}_2	\hat{e}_3	\hat{e}_{n-k}

Opravování chyb podle syndromu

- Opravu chyby pak provedeme takto:

$$\mathbf{s} = \mathbf{H} \cdot \mathbf{w} = \mathbf{s}_j \quad , \quad \hat{\mathbf{v}} = \mathbf{w} - \omega^{-1}(\mathbf{s}_j) = \mathbf{w} - \hat{\mathbf{e}}_j$$

$\hat{\mathbf{v}}$ 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 $H = \begin{bmatrix} 0001111 \\ 0110011 \\ 1010101 \end{bmatrix}$

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

$e = [1000000]^T$	$s = [001]^T$	$e = [0000100]^T$	$s = [101]^T$
$e = [0100000]^T$	$s = [010]^T$	$e = [0000010]^T$	$s = [110]^T$
$e = [0010000]^T$	$s = [011]^T$	$e = [0000001]^T$	$s = [111]^T$
$e = [0001000]^T$	$s = [100]^T$		

Budeme pokračovat syndromy dvojitých chyb:

$e = [1100000]^T$	$s = [011]^T$
$e = [1010000]^T$	$s = [010]^T$

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

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 $\mathbf{e}_i = [0 \dots 010 \dots 0]^T$ s jedničkou v i -tém znaku.

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

kde $\mathbf{H}_{*,i}$ představuje i -tý sloupec kontrolní matice \mathbf{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

$$\mathbf{H}_{*,i} \neq \mathbf{0} \quad \forall i = 1, 2, \dots, n$$

nesmí obsahovat stejné sloupce

$$\mathbf{H}_{*,i} \neq \mathbf{H}_{*,j} \quad \forall i, j = 1, 2, \dots, n, \quad i \neq j$$

- 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 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)

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

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

$$\mathbf{H} = \begin{bmatrix} 0111 & 100 \\ 1011 & 010 \\ 1101 & 001 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 1000 & 011 \\ 0100 & 101 \\ 0010 & 110 \\ 0001 & 111 \end{bmatrix}$$

systematický kód

$$\mathbf{H} = \begin{bmatrix} 1110 & 100 \\ 0111 & 010 \\ 1101 & 001 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} 1000 & 101 \\ 0100 & 111 \\ 0010 & 110 \\ 0001 & 011 \end{bmatrix}$$

systematický kód
(navíc ještě cyklický)

na pořadí sloupců v matici \mathbf{H} nezáleží

Hammingovy kódy - konstrukce

- Pro zadaný počet kontrolních prvků r z rovnosti $2^r = k + r + 1$ spočítáme k
- Systematická generující matice \mathbf{G} má k řádků, $k + r$ sloupců a tvar $\mathbf{G} = [\mathbf{I}_k \mid \mathbf{B}]$
- Levá submatice \mathbf{I}_k je jednotková matice řádu k
- Do řádků pravé submatice \mathbf{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 \mathbf{B} nezáleží. **Dobrý zvyk – psát je ve stoupajícím pořadí.**
- Systematická kontrolní matice \mathbf{H} má r řádků, $k + r$ sloupců
- Kontrolní matici \mathbf{H} pak vytvoříme jako $\mathbf{H} = [-\mathbf{B}^T \mid \mathbf{I}_{n-k}]$.

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 \mathbf{G} .
- Každý řádek \mathbf{G} obsahuje alespoň tři jedničky (jednu v části \mathbf{I}_k , přinejmenším dvě v části \mathbf{B}).
- Součet libovolných dvou řádků \mathbf{G} obsahuje dvě jedničky v části \mathbf{I}_k a přinejmenším jednu jedničku v části \mathbf{B} (protože v \mathbf{B} nemohou být dva stejné řádky).
- Součet libovolných tří řádků \mathbf{G} obsahuje tři jedničky v části \mathbf{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 $2^r = k + r + 1$ ($k = 4, 11, 26, 57, \dots$ 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 G a H Hammingova kódu pro r zabezpečovacích znaků
 - z matic G a H vypustíme řádky a sloupce odpovídající nepotřebným informačním prvkům (z matice G řádky a sloupce, z matice H sloupce)

Zkracování Hammingových kódů

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

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

$$G = \begin{bmatrix} 100000 & 00000 & 0011 \\ 010000 & 00000 & 0101 \\ 001000 & 00000 & 0110 \\ 000100 & 00000 & 0111 \\ 000010 & 00000 & 1001 \\ 000001 & 00000 & 1010 \\ 000000 & 10000 & 1011 \\ 0000000 & 1000 & 1100 \\ 00000000 & 100 & 1101 \\ 000000000 & 10 & 1110 \\ 0000000000 & 1 & 1111 \end{bmatrix}$$

$$H = \begin{bmatrix} 000011 & 11111 & 1000 \\ 011100 & 01111 & 0100 \\ 101101 & 10011 & 0010 \\ 110110 & 10101 & 0001 \end{bmatrix}$$

$$G = \begin{bmatrix} 100000 & 0011 \\ 010000 & 0101 \\ 001000 & 0110 \\ 000100 & 0111 \\ 000010 & 1001 \\ 000001 & 1010 \end{bmatrix}$$

$$H = \begin{bmatrix} 000011 & 1000 \\ 011100 & 0100 \\ 101101 & 0010 \\ 110110 & 0001 \end{bmatrix}$$

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 Hammingova 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:**

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

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

modře podbarvené – zabezpečovací prvky parity

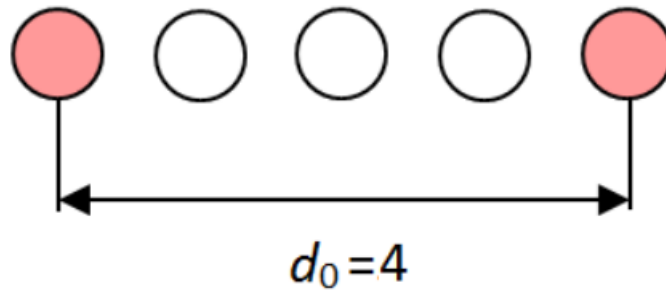
matice H_1 - kontrolní matice vytvořená na základě vztahu $H = [-B^T | I_{n-k}]$

matice 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**
 - trojitě 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

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

- 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) $v_2 = v_9$, $v_2 = v_{10}$, $v_2 = v_{11}$, $v_1 = v_{12}$

(pro prvek u_3) $v_3 = v_{13}$, $v_3 = v_{14}$, $v_3 = v_{15}$, $v_3 = v_{16}$

(pro prvek u_4) $v_4 = v_{17}$, $v_4 = v_{18}$, $v_2 = v_{19}$, $v_1 = v_{20}$

Oprava chyb u opakovacích kódů

- Kontrolní rovnice (v nehomogenním tvaru):

$$\text{(pro prvek } u_1 \text{) } v_1 = v_5 \text{ , } v_1 = v_6 \text{ , } v_1 = v_7 \text{ , } v_1 = v_8$$

$$\text{(pro prvek } u_2 \text{) } v_2 = v_9 \text{ , } v_2 = v_{10} \text{ , } v_2 = v_{11} \text{ , } v_1 = v_{12}$$

$$\text{(pro prvek } u_3 \text{) } v_3 = v_{13} \text{ , } v_3 = v_{14} \text{ , } v_3 = v_{15} \text{ , } v_3 = v_{16}$$

$$\text{(pro prvek } u_4 \text{) } v_4 = v_{17} \text{ , } v_4 = v_{18} \text{ , } v_2 = v_{19} \text{ , } v_1 = v_{20}$$

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

$$\text{(pro prvek } u_1 \text{) } \tilde{u}_1 = w_5 \text{ , } \tilde{u}_1 = w_6 \text{ , } \tilde{u}_1 = w_7 \text{ , } \tilde{u}_1 = w_8 \text{ , } \tilde{u}_1 = w_1$$

$$\text{(pro prvek } u_2 \text{) } \tilde{u}_2 = w_9 \text{ , } \tilde{u}_2 = w_{10} \text{ , } \tilde{u}_2 = w_{11} \text{ , } \tilde{u}_2 = w_{12} \text{ , } \tilde{u}_2 = w_2$$

$$\text{(pro prvek } u_3 \text{) } \tilde{u}_3 = w_{13} \text{ , } \tilde{u}_3 = w_{14} \text{ , } \tilde{u}_3 = w_{15} \text{ , } \tilde{u}_3 = w_{16} \text{ , } \tilde{u}_3 = w_3$$

$$\text{(pro prvek } u_4 \text{) } \tilde{u}_4 = w_{17} \text{ , } \tilde{u}_4 = w_{18} \text{ , } \tilde{u}_4 = w_{19} \text{ , } \tilde{u}_4 = w_{20} \text{ , } \tilde{u}_4 = w_4$$

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

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

0123.....10

čtvercová submatice B řádu 11
vznikla cyklickými posuvy prvního řádku

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

Golayovy kódy

- Golayův kód 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 G_{23} je $2^r = 2^{11}$.
- Počet různých chybových vektorů váhy $t \leq 3$ je

$$\begin{aligned} 1 + 23 + \binom{23}{2} + \binom{23}{3} &= 1 + 23 + \frac{23!}{21! \cdot 2!} + \frac{23!}{20! \cdot 3!} = 1 + 23 + \frac{23 \cdot 22}{2 \cdot 1} + \frac{23 \cdot 22 \cdot 21}{3 \cdot 2 \cdot 1} = \\ &= 1 + 23 + 23 \cdot 11 + 23 \cdot 11 \cdot 7 = 1 + 23 + 253 + 1771 = 2048 = 2^{11} \end{aligned}$$

- Existuje vzájemně jednoznačné přiřazení mezi syndromy a chybovými vektory váhy $t \leq 3$
- Golayův kód 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:

[illegible]

Golayovy kódy

- Vlastnosti matice G_{24} :
 - v každém řádku s výjimkou posledního je 8 jedniček, v posledním je jich 12
 - každá lineární kombinace řádků 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 $H_{24} = G_{24}$
- Použití kódu 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 $m \in N$ a $r \in N_0$, kdy $0 \leq r \leq m$, lze zkonstruovat matici o $n = 2^m$ sloupcích jako blokovou matici

$$G = \begin{bmatrix} G_0 \\ G_1 \\ \vdots \\ G_r \end{bmatrix} \quad \begin{array}{l} G_0 = [1 \ 1 \ \dots \ 1] \text{ je matice typu } 1/n \\ G_1 = \begin{bmatrix} 0 & 0 & \dots & 0 & 1 \\ & & & \vdots & \\ 0 & 0 & \dots & 1 & 1 \\ 0 & 1 & \dots & 1 & 1 \end{bmatrix} \text{ je matice typu } m/n, \\ \text{obsahující popořadě} \\ \text{všechny sloupce z 0 a 1} \end{array} \quad \begin{array}{l} G_l, \text{ kde } 2 \leq l \leq r \text{ jsou matice,} \\ \text{jejichž řádky obsahují} \\ \text{všechny možné součiny} \\ l \text{ řádků} \end{array}$$

Reed-Mullerovy kódy – ilustrační příklad

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

$$\mathbf{G}_0 = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

$$\mathbf{G}_1 = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \\ \vdots \\ \mathbf{G}_r \end{bmatrix}$$

$$\mathbf{G}_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \quad \begin{array}{l} 1.\text{ř. } \mathbf{G}_1 \cdot 2.\text{ř. } \mathbf{G}_1 \\ 2.\text{ř. } \mathbf{G}_1 \cdot 3.\text{ř. } \mathbf{G}_1 \\ 1.\text{ř. } \mathbf{G}_1 \cdot 3.\text{ř. } \mathbf{G}_1 \end{array} \quad (\text{součiny řádků jsou „bit po bitu“})$$

$$\mathbf{G}_3 = [0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1] \quad 1.\text{ř. } \mathbf{G}_1 \cdot 2.\text{ř. } \mathbf{G}_1 \cdot 3.\text{ř. } \mathbf{G}_1$$

Reed-Mullerovy kódy – ilustrační příklad

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

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \\ \mathbf{G}_2 \\ \mathbf{G}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

v každé matici \mathbf{G}_l mají všechny řádky stejný počet jedniček, a to 2^{m-l}

řádky matice \mathbf{G} jsou lineárně nezávislé

obecné značení R-M kódů je $K_{r,m}$

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

Reed-Mullerovy kódy – ilustrační příklad

Vytvoření matice \mathbf{G} ($m = 3$, $r = 2$, $n = 2^m = 8$) :

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \\ \mathbf{G}_2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

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 \mathbf{G} ($m = 3$, $r = 1$, $n = 2^m = 8$) :

$$\mathbf{G} = \begin{bmatrix} \mathbf{G}_0 \\ \mathbf{G}_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

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

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

Vytvoření matice \mathbf{G} ($m = 3$, $r = 0$, $n = 2^m = 8$) :

$$\mathbf{G} = [\mathbf{G}_0] = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

$$k = 1$$

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

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)