Hammingův kód

Binární kód se nazývá Hammingův, jestliže má kontrolní matici, jejíž sloupce jsou všechna nenulová slova dané délky $n - k = r$ a žádné z nich se neopakuje.

Jedná se o speciální případ lineárních dvojkových (n, k) kódů. Tyto kódy opravují jednu chybu při vzdálenosti kódových slov $d_{min}(\boldsymbol{b}_i, \boldsymbol{b}_j) = 3$ a v rozšířené variantě $d_{min}(\boldsymbol{b}_i, \boldsymbol{b}_j) = 4$.

Algoritmus generování Hammingova kódu

1. Všechny bitové pozice jejichž číslo je rovné mocnině 2 jsou použity pro paritní bit (1, 2, 4, 8, 16, 32,...).

2. Všechny ostatní bitové pozice náleží kódovanému informačnímu slovu (3, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 17,...).

3. Každný paritní bit je vypočítán z některých bitů informačního slova. Pozice paritního bitu udává sekvenci bitů, které jsou v kódovém slově zjišťovány a které přeskočeny.

   Pro paritní bit p_1 (pozice 1) se ve zbylém kódovém slově 1 bit přeskočí, 1 zkontroluje, 1 bit přeskočí, 1 zkontroluje, atd. Pro paritní bit p_2 (pozice 2) přeskočí první bit, 2 zkontroluje, 2 přeskočí, 2 zkontroluje, atd. Pro p_3 (pozice 4) přeskočí první 3 bity, 4 zkontroluje, 4 přeskočí, 4 zkontroluje, atd.

Pro kód $(7, 4)$ platí $\boldsymbol{b} = \left(p_1^{(2^0)}, p_2^{(2^1)}, a_1^3, p_3^{(2^2)}, a_2^5, a_3^6, a_4^7\right)$:

- $p_1 \oplus a_1 \oplus a_2 \oplus a_4 = 0$ (podle bodu 3 sestaveno z b_1, b_3, b_5, b_7),
- $p_2 \oplus a_1 \oplus a_3 \oplus a_4 = 0$ (b_2, b_3, b_6, b_7),
- $p_3 \oplus a_2 \oplus a_3 \oplus a_4 = 0$ (b_4, b_5, b_6, b_7).

Generující matice $\mathbb{G}_H$ Hamming. kódu $(7, 4)$ se sestrojí tak, že se postupně zakóduje posloupnost 1000_1, 0100_2, 0010_3, 0001_4 (proto aby řádky byly lineárně nezávislé a tvořily bázi prostoru).

$$\mathbb{G}_H = \begin{pmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & p_{1_1} & p_{2_1} & 1 & p_{3_1} & 0 & 0 & 0 \\ 2 & p_{1_2} & p_{2_2} & 0 & p_{3_2} & 1 & 0 & 0 \\ 3 & p_{1_3} & p_{2_3} & 0 & p_{3_3} & 0 & 1 & 0 \\ 4 & p_{1_4} & p_{2_4} & 0 & p_{3_4} & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 3 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 4 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Kontrolní matice $\mathbb{H}_H$ Hamming. kódu $(7, 4)$ se určí následovně. Po přijetí kódového slova $\boldsymbol{b}$ víme, že bity b_3, b_5, b_6, b_7 obsahují informační slovo a zbylé redundantní bity jsou určeny tak, aby

$$\begin{array}{l} s_1 = b_4 \oplus b_5 \oplus b_6 \oplus b_7 = 0 \\ s_2 = b_2 \oplus b_3 \oplus b_6 \oplus b_7 = 0 \\ s_3 = b_1 \oplus b_3 \oplus b_5 \oplus b_7 = 0 \end{array} \Rightarrow \mathbb{H}_H = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

Vektor $\boldsymbol{s} = (s_1, s_2, s_3)$ se nazývá *syndrom* a pokud byla informace přijata bezchybně jeho hodnota je $\boldsymbol{s} = (0, 0, 0)$.

Rozšířený Hammingův kód

Rozšíření binárního Hammingova kódu vychází z toho, že přídáme na začátek každého kódového slova nový symbol určený pro kontrolu parity celého kódového slova. Bit p_0 je zvolen tak, aby $p_0 \oplus b_1 \oplus b_2 \oplus b_3 \oplus b_4 \oplus b_5 \oplus b_6 \oplus b_7$ vycházelo jako sudé číslo. Rozšířený kód dovoluje, tak jako předchozí opravit jednu chybu a navíc je schopen detekovat dvě chyby.

Generující matice $\mathbb{G}'_H$ rozšířeného Hamming. kódu $(8, 4)$ se sestrojí tak, že se postupně k zakóduje posloupnost 1000_1, 0100_2, 0010_3, 0001_4

$$\mathbb{G}'_H = \begin{pmatrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & p_{0_1} & p_{1_1} & p_{2_1} & 1 & p_{3_1} & 0 & 0 & 0 \\ 2 & p_{0_2} & p_{1_2} & p_{2_2} & 0 & p_{3_2} & 1 & 0 & 0 \\ 3 & p_{0_3} & p_{1_3} & p_{2_3} & 0 & p_{3_3} & 0 & 1 & 0 \\ 4 & p_{0_4} & p_{1_4} & p_{2_4} & 0 & p_{3_4} & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 3 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 4 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Dekódování a kontrola

Nejprve se po přijetí kódového slova $\boldsymbol{b}$ určí syndrom $\boldsymbol{s} = \mathbb{H}_H \cdot \boldsymbol{b}^T$. Například pro přijaté slovo $\boldsymbol{b} = (1010111)$ je syndrom

$$\mathbb{H}_H \cdot \boldsymbol{b}^T = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$$

Vidíme, že syndrom $\boldsymbol{s}$ je nenulový, tj. při přenosu došlo k chybě. Syndrom, který vyšel $\boldsymbol{s} = (1, 1, 0)$ odpovídá sloupci 6 kontrolní matice $\mathbb{H}_H$ a z toho vyplývá, že je třeba opravit šestý bit kódového slova $\boldsymbol{b'} = (1010\underline{1}01)$.

Návrh logiky počítačů — 3. test

A. Lineární kódy

1. Najděte generovací a kontrolní matici „koktavého“ kódu (6,2).

   Koktavý kód (jk, k) zopakuje $j\times$ počet bitů informačního slova, takže např. pro $\boldsymbol{a} = (0, 1)$, je kódové slovo $\boldsymbol{b} = (0, 0, 0, 1, 1, 1)$. Generující matice je

   $$\mathbb{G} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix},$$

pro kontrolní matici $\mathbb{H}$ platí $\mathbb{G} \cdot \mathbb{H}^T = 0$, a tato matice je typu $n \times r = n \times (n-k)$, takže v případě kódu (6,2) je kontrolní matice $\mathbb{H}$ typu 6×4.

$$\mathbb{H} = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \end{pmatrix}$$

2. Najděte generovací a kontrolní matici kódu zabezpečeného sudou paritou, jehož kódová slova mají 4 informační bity. O který kód (n, k) půjde – určete n a k. Jak se generovací matice změní pro lichou paritu?

   Jestliže kódová slova mají 4 informační bity, pak délka informačního slova je $k = 4$. Paritní kód $(k+1, k)$ funguje tak, že informační délka slova je doplněna jedním kontrolním bitem p proto, aby počet jedniček v kódovém slovu byl sudý ($p = a_1 \oplus \ldots \oplus a_k$), tudíž $n = 5$. Generující matice $\mathbb{G}$ má sudý počet jedniček v řádku, může to být například:

   $$\mathbb{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix}, \quad \text{kontrolní } \mathbb{H} = (1\ 1\ 1\ 1\ 1), \quad \mathbb{H}^T = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

   Matice $\mathbb{H}$ se získala transponováním posledního sloupce matice $\mathbb{G}$ a přidáním jednotkové matice typu 1×1.
   Pro lichou paritu ($p = a_1 \oplus \ldots \oplus a_k \oplus 1$) nelze vytvořit generující matici, protože se nejedná o lineární kód.

3. Najděte generovací matici kódu $(8, 4)$ zabezpečeného podélnou a příčnou sudou paritou. (Kódová slova mají 4 informační a 4 paritní bity.)

   Podélná parita se určí jako $p_1 = a_1 + a_2$, $p_2 = a_3 + a_4$, příčná parita $p_3 = a_1 + a_3$, $p_4 = a_2 + a_4$, generující matice $\mathbb{G}$ se určí postupným zakódováním posloupnosti 1000, 0100, 0010, 0001.

   $$\mathbb{G} = \begin{pmatrix} a_1 & a_2 & a_3 & a_4 & p_1 & p_2 & p_3 & p_4 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}.$$

4. Najděte kontrolní matici kódu, který umožňuje opravu 1 chyby (tzn. kódu SEC). Kódová slova mají obsahovat 6 informačních bitů.

   Jestliže kódová slova mají mít 6 informačních bitů, pak délka informačního slova je $k = 6$. Nejbližší vyšší Hammingův kód pro zabezpečení 6 bitů je

$(15, 11)$, z toho vidíme že je třeba – $r = n - k = 4$ zabezpečovací bity, takže hledaný kód má generující matici typu $(10, 6)$. Kontrolní matice $\mathbb{H}$ je pro tento kód typu 10×4. Kontrolní matici vytvoříme tak, že

$$\mathbb{H} = \begin{pmatrix} . & . & . & . & . & . & 1 & 0 & 0 & 0 \\ . & . & . & . & . & . & 0 & 1 & 0 & 0 \\ . & . & . & . & . & . & 0 & 0 & 1 & 0 \\ . & . & . & . & . & . & 0 & 0 & 0 & 1 \end{pmatrix}.$$

Do volných sloupců se vloží kombinace nul a jedniček tak, aby sloupce byly jedinečné a zároveň všechny řádky byly lineárně nezávislé.

5. Je dána generovací matice $\mathbb{G}$ lineárního kódu $\mathcal{K}$:

$$\mathbb{G} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Najděte generovací matici $\mathbb{G}'$ systematického lineárního kódu $\mathcal{K}'$, který má stejnou množinu kódových slov jako kód $\mathcal{K}$. Najděte kontrolní matici kódu $\mathcal{K}$ i kódu $\mathcal{K}'$.

Pro nalezení kódu $\mathcal{K}'$ ekvivalentního s kódem $\mathcal{K}$ (tzn. množina kódových slov je stejná a liší se pouze přiřazením) se nejprve provede úprava matice $\mathbb{G}$ do tvaru $\mathbb{G} = (\mathbb{E}|\mathbb{F})$ např. Gaussovou eliminací (na $\mathbb{Z}_2$):

$$\mathbb{G} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \sim \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right) = \mathbb{G}'$$

Kontrolní matice systematického kódu je $\mathbb{H}' = (\mathbb{F}^T|\mathbb{E})$.

$$\mathbb{H}' = \left(\begin{array}{ccc|ccc} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right)$$

Výsledná kontrolní matice $\mathbb{H}'$ je zároveň kontrolní maticí pro původní $\mathbb{G}$ (množina kódových slov je stejná).

6. Je dána generovací matice $\mathbb{G}$ lineárního kódu $\mathcal{K}$:

$$\mathbb{G} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Napište tabulku kódu $\mathcal{K}$ a určete jeho kódovou vzdálenost – uveďte postup. Kolik chyb je možné opravit?

Řádky generující matice $\mathbb{G}$ jsou generujícími vektory, které tvoří bázi lineárního prostoru. Tyto vektory musejí být lineárně nezávislé, což v případě výše uvedené matice $\mathbb{G}$ neplatí.

$$\mathbb{G} = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1_1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0_2 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1_3 \end{pmatrix} \sim \begin{pmatrix} 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & {\scriptstyle 1} \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & {\scriptstyle (2-1)} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & {\scriptstyle (3-2-1)} \end{pmatrix}$$

Výsledná matice $\mathbb{G}$ je generující maticí pro kód (9, 2).

n	$\boldsymbol{a}$	$\boldsymbol{b}$	Váha $\boldsymbol{b}$
0	00	000000000	0
1	01	011011011	6
2	10	101101101	6
3	11	110110110	6

Kódová vzdálenost tohoto kódu je $k_{vzd} = (\text{min. váha } \boldsymbol{b} \neq 0) = 6$ Tento kód je schopen opravit maximálně 2 chyby (2 opravit a 3 detekovat). A nebo detekovat maximálně 5 chyb.

7. Je dána generovací matice $\mathbb{G}$ lineárního kódu $\mathcal{K}$:

$$\mathbb{G} = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Napište tabulku kódu $\mathcal{K}$ a určete jeho kódovou vzdálenost – uveďte postup. Kolik chyb je možné opravit?

n	$\boldsymbol{a}$	$\boldsymbol{b}$	Váha $\boldsymbol{b}$
0	000	000000	0
1	001	011110	4
2	010	110011	4
3	011	101101	4
4	100	111100	4
5	101	100010	2
6	110	001111	4
7	111	010001	2

Kódové slovo se zjistí ze vztahu $\boldsymbol{b} = \boldsymbol{a} \cdot \mathbb{G}$.

Kódová vzdálenost tohoto kódu je $k_{vzd} = (\text{min. váha } \boldsymbol{b} \neq 0) = 2$ (minimální d např. mezi 1 a 4). Kód s $k_{vzd} = 2$ je schopen detekovat 1 chybu, je tedy typu SED (Single Error Detection), a není schopen opravit žádnou chybu.

8. Kolik kontrolních bitů budou mít kódová slova kódu pro opravu 1 chyby a detekci 2 chyb (tj. kódu SEC-DED) mají-li kódová slova 64 informačních bitů. Kolik řádků a sloupců bude mít generovací a kontrolní matice. Jaká bude kódová vzdálenost?

Jestliže kódová slova obsahují 64 informačních bitů, pak délka informačního slova je $k = 64$. Kódová vdálenost pro SEC-DED musí být $k_{vzd} = 4$. Nejbližší vyšší Hammingův kód je typu $(127, 120)$, má tedy 7 kontrolních bitů. Takže SEC-DED pro $k = 64$ je typu $(71, 64)$. Generující matice $\mathbb{G}$ je typu 71×64 a kontrolní $\mathbb{H}$ je typu $k \times r = 64 \times 7$.

B. Cyklické kódy

1. Napište polynom $G(x)$, který přísluší vektoru 1011. Tento polynom je generovacím polynomem systematického cyklického kódu (7,4). Určete kontrolní část kódového slova, jehož informační částí je $a = 1001$.

   Příslušný polynom je $(1, 0, 1, 1) \sim G(x) = x^3 + x + 1$. Pokud je $G(x)$ generovacím polynomem systematického cyklického kódu (n, k), pak obecně platí:

   $$G(x) \cdot H(x) = x^n + 1.$$

   Systematický kód má

   $$B(x) = A(x) \cdot x^m + (A(x) \cdot x^m) \,\%\, G(x),$$

   kde $m = \deg G(x)$ je stupeň generujícího polynomu.
   Pro $a = 1001$ je $A(x) = x^3 + 1$, pak $B(x) = (x^3 + 1)x^3 + ((x^3 + 1) \cdot x^3) \,\%\, (x^3 + x + 1)$.
   Kontrolní část slova je $((x^3 + 1) \cdot x^3) \,\%\, (x^3 + x + 1)$. Provedeme tedy nalezení zbytku po dělenení, těchto dvou polynomů. Všechny operace se provádějí na tělese $\mathbb{Z}_2$:

   $$\begin{array}{ll} & (x^6 + x^3) \quad : (x^3 + x + 1) = x^3 + x \\ - & \underline{(x^6 + x^4 + x^3)} \\ & x^4 \\ & \underline{-(x^4 + x^2 + x)} \\ & x^2 + x \quad \leftarrow \text{zbytek} \end{array}$$

   Polynom zbytku $x^2 + x$ odpovídá kontrolní části kódového slova $(1, 1, 0)$.

   $$\boldsymbol{b} = \boxed{\text{Informační část}} \boxed{\text{CRC}} = \boxed{1001} \boxed{110}$$

2. Napište polynom $G(x)$, který přísluší vektoru 1101. Tento polynom je generovacím polynomem systematického cyklického kódu $(7, 4)$. Určete kontrolní část kódového slova, jehož informační částí je $a = 1001$.

   Příslušný polynom je $(1, 1, 0, 1) \sim G(x) = x^3 + x^2 + 1$. Pro $a = 1001$ je $A(x) = x^3 + 1$, pak $B(x) = (x^3 + 1)x^3 + ((x^3 + 1) \cdot x^3) \,\%\, (x^3 + x^2 + 1)$.
   Kontrolní část slova je $((x^3 + 1) \cdot x^3) \,\%\, (x^3 + x^2 + 1)$. A po zjištění zbytku po dělení polynomů na $\mathbb{Z}_2$:

   $$\begin{array}{l} (x^6 + x^3) \quad : (x^3 + x^2 + 1) = x^3 + x^2 + x + 1 \\ \underline{-(x^6 + x^5 + x^3)} \\ x^5 \\ \underline{-(x^5 + x^4 + x^2)} \\ x^4 + x^2 \\ \underline{-(x^4 + x^3 + x)} \\ x^3 + x^2 + x \end{array}$$

$$\begin{array}{l} -(x^3+x^2+1) \\ \hline x+1 \end{array} \quad \leftarrow \text{zbytek}$$

$$\boldsymbol{b} = \boxed{\text{Informační část}} \boxed{\text{CRC}} = \boxed{1001} \boxed{011}$$

3. Napište polynom $G(x)$, který přísluší vektoru 11001. Tento polynom je generovacím polynomem:

   a) systematického cyklického kódu (15,11);

   b) nesystematického cyklického kódu (15,11).

   Určete, zda slovo $\boldsymbol{c} = 100100010111010$ je či není kódovým slovem některého z těchto kódů. Kterého z nich?

   Příslušný polynom je $(1,1,0,0,1) \sim G(x) = x^4 + x^3 + 1$ (pro řád generujícího polynomu platí $r = \deg(G(x)) = n - k$. Pokud je vektor $\boldsymbol{c} \sim B(x)$ kódovým slove musí platit, že $B(x) \,\%\, G(x) = 0$.
   Určíme $A(x) = B(x) : G(x)$ a přitom ověříme zda $B(x) \,\%\, G(x) = 0$ (vše na $\mathbb{Z}_2$):

   $$(x^{14}+x^{11}+x^7+x^5+x^4+x^3+x) : (x^4+x^3+1) = x^{10}+x^9+x^8+x^6+x^4+x^2,$$

   $$(x^{14}+x^{11}+x^7+x^5+x^4+x^3+x) \,\%\, (x^4+x^3+1) = x^3+x^2+x.$$

   Výsledek určuje, že $\boldsymbol{c}$ není kódovým slovem cyklického kódu a to ani jednoho z obou typů.

4. Napište polynom $G(x)$, který přísluší vektoru 1001. Tento polynom je generovacím polynomem systematického cyklického kódu $(9,6)$. Určete kontrolní část kódového slova, jehož informační částí je $\boldsymbol{a} = 110011$.

   Příslušný polynom je $(1,0,0,1) \sim G(x) = x^3 + 1$. Pro $a = 110011$ je $A(x) = x^5 + x^4 + x + 1$.
   Kontrolní část slova je $(A(x) \cdot x^m) \,\%\, G(x)$:

   $$((x^5 + x^4 + x + 1) \cdot x^3) \,\%\, (x^3 + 1) = x^2 + 1.$$

   Výsledný polynom $x^2 + 1$ odpovídá kontrolní části kódového slova $(1,0,1)$.

   $$\boxed{\text{Informační část}} \boxed{\text{CRC}} = \boxed{110011} \boxed{101}$$

5. Napište polynom $G(x)$, který přísluší vektoru 11111. Tento polynom je generovacím polynomem:

   a) systematického cyklického kódu $(10,6)$;

   b) nesystematického cyklického kódu $(10,6)$.

Určete, zda slovo $\boldsymbol{c} = 1000010000$ je či není kódovým slovem některého z těchto kódů. Kterého z nich?

Příslušný polynom je $(1,1,1,1,1) \sim G(x) = x^4 + x^3 + x^2 + x + 1$ (pro řád generujícího polynomu platí $r = \deg(G(x)) = n - k$). Pokud je vektor $\boldsymbol{c} \sim B(x)$ kódovým slovem musí platit, že $B(x) \,\%\, G(x) = 0$.

$$(x^9 + x^4) : (x^4 + x^3 + x^2 + x + 1) = x^5 + x^4$$
$$(x^9 + x^4) \,\%\, (x^4 + x^3 + x^2 + x + 1) = 0$$

Výsledek určuje, že $\boldsymbol{c}$ je kódovým slovem a to *obou* typů kódů. Jak systematický, tak i nesystematický kód mají stejnou množinu kódových slov a liší se pouze jejich přiřazení informačním slovům (u systematického by prvních 6 bitů odpovídalo informačnímu slovu a u nesystematického by byly bity různě promíchány).

6. Kódovými slovy lineárního kódu (n, k) jsou 9bitová čísla, jejichž šestnáctkové zápisy jsou: 0, 49, 92, DB, 124, 16D, 1B6, 1FF. Určete hodnoty n a k. Je tento kód cyklický? Pokud ano, najděte generovací polynom a generovací matici.

$B(x)_{16}$	$B(X)_2$	Polynomiální vyjádření
0	000000000	0
49	001001001	$x^6 + x^3 + 1$
92	010010010	$x^7 + x^4 + x = x(x^6 + x^3 + 1)$
DB	011011011	$x^7 + x^6 + x^4 + x^3 + x + 1 = (x + 1)(x^6 + x^3 + 1)$
124	100100100	$x^8 + x^5 + x^2 = x^2(x^6 + x^3 + 1)$
16D	101101101	$x^8 + x^6 + x^5 + x^3 + x^2 + 1 = (x^2 + 1)(x^6 + x^3 + 1)$
1B6	110110110	$x^8 + x^7 + x^5 + x^4 + x^2 + x = x(x + 1)(x^6 + x^3 + 1)$

Uvedený kód je lineární a kódová slova v tabulce splňují podmínku, že pro všechna slova platí, že po cyklickém posuvu vzniká nějaké jiné slovo. Z polynomiálního vyjádření vychází, že generujícím polynomem je $G(x) = x^6 + x^3 + 1$. Daný kód je typu (9,3), vidíme to z toho, že délka kódového slova je 9 bitů a ze stupně $G(x)$, který je $\deg G(x) = 6$ a udává redundanci r, $k = n - r = 9 - 6 = 3$. Z binárního vyjádření $B(x)_2$ je navíc zřejmé že se jedná o opakovací kód – informační slovo je $3\times$ zopakováno.

Příslušná generující matice $\mathbb{G}_{(9,3)}$ se získá stejně jako u všech lineárních kódů zakódováním posloupnosti informačních slov $100 \sim x^2$, $010 \sim x$, $001 \sim 1$ generujícím polynomem:

$$\mathbb{G} = \begin{pmatrix} x^{(k-1)} \cdot G(x) \\ \vdots \\ x^1 \cdot G(x) \\ x^0 \cdot G(x) \end{pmatrix} = \begin{pmatrix} g_r & g_{r-1} & \dots & 0 \\ 0 & g_r & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & g_0 \end{pmatrix}$$

$$\mathbb{G}_{(9,3)} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

7. Napište polynom $G(x)$, který přísluší vektoru 11011. Určete (a zdůvodněte), zda polynom $G(x)$ může být generovacím polynomem cyklického kódu $(12, k)$. Pokud ano, určete hodnotu k.

   Příslušný polynom je $(1,1,0,1,1) \sim G(x) = x^4+x^3+x+1$. Aby tento polynom byl generovacím polynomem cyklického kódu musí platit, že $G(x)$ dělí beze zbytku polynom $x^n + 1 = x^{12} + 1$.

   $$(x^{12} + 1) : (x^4 + x^3 + x + 1) = x^8 + x^7 + x^6 + x^2 + x + 1$$

   Polynom $G(x)$ je tedy generovacím polynomem cyklického kódu $(12, k)$. Délka informačního slova je $k = n - r = n - \deg G(x) = 12 - 4 = 8$, jedná se o kód $(12, 8)$.

8. Napište polynom $G(x)$, který přísluší vektoru 100101. Polynom $G(x)$ je generovacím polynomem cyklického kódu $(31, k)$. Určete hodnotu k. Kolik informačních bitů a kolik kontrolních bitů bude obsahovat kódové slovo a jak bude dlouhé (půjde-li o kód systematický)? Jaký bude stupeň kontrolního polynomu? Kolik bude celkem kódových slov? Jakou bude mít kód redundanci? Jaké shluky chyb lze detekovat?

   Příslušný polynom je $(1,0,0,1,0,1) \sim G(x) = x^5 + x^2 + 1$. Hodnota $k = n - \deg G(x) = 31 - 5 = 26$. Kódové slovo tohoto kódu obsahuje 26 informačních a 5 kontrolních bitů, celková délka je 31 bitů. Stupeň kontrolního polynomu bude 26. Kódových slov tohoto kódu je $2^{26} = 67\,108\,864$. Redundance kódu je $r = 5$. Obecně je cyklický kód schopen detekovat shluky chyb délky $l \leq r = \deg G(x)$, takže v našem případě je to $l \leq 5$.

C. Bezpečnostní kódy – obecně

1. Co je symetrický kanál? Co je kanál s pamětí? U kterých typů kanálů lze očekávat shluky chyb?

   Symetrický kanál je kanál, kde pravděpodobnost, že vyslaná cifra je správně přijata, nezávisí na této cifře a vyskytují se v něm symetrické chyby. *Symetrická chyba* je chyba, pří níž se přibližně stejnou pravděpodobností může v zakódované informaci změnit vlivem poruchy 0 na 1 nebo 1 na 0.

   Kanál s pamětí je to kanál u kterého se objevují *shluky chyb*, tzn. že chyba slova může být ovlivněna přenosem sousedních slov. Např. při sériové komunikaci je stejnou chybou ovlivněno několik slov za sebou nebo u poškrábaného kompaktního disku.
   U kanálu *bez paměti* vznikají se stejnou pravděpodobností jednotlivé „nezávislé“ chyby (např. náhodné chyby v pamětech RAM, ROM,... :-).

2. Co je blokový kód? Co je systematický kód? Co je samoopravný kód? Které bezpečnostní kódy nejsou samoopravné?

   Blokový kód – obsahuje kódová slova pevné délky.

Systematický kód – prvních k souřadnic kódového vektoru tvoří nezměněné informační bity a pak je bezprostředně připojena kontrolní část.
To je splněno pokud kontrolní matice je $\mathbb{H} = (\mathbb{F} \,|\, \mathbb{E}_{n-k})$. Kde $\mathbb{F}$ je binární matice typu $(n-k) \times k$ a $\mathbb{E}_{n-k}$ je čtvercová jednotková matice typu $(n-k) \times (n-k)$. Generujíc matice je $\mathbb{G} = (\mathbb{E}_k \,|\, \mathbb{F}^T)$.

Samoopravný kód – je to kód, který umožnuje opravu chyb, tzn. umožnuje zjišťovat na kterém bitu vznikla chyba a je schopen daný bit opravit na přijímací straně.

Paritní, koktavý a opakovací kód *nejsou* samoopravné.

3. Jakou informační entropii, jakou redundanci a jakou přenosovou rychlost má kód (n, k)? Kolik má kódových slov?

Informační entropie – množství přenesené informace k;

redundance – počet zabezpečovacích bitů $r = n - k$, relativní redundance udává poměr r/n;

přenosová rychlost – norma kódu, informační poměr kódu

$$R = \frac{k}{n} = \frac{n-r}{n}, \quad R \in (0, 1\rangle;$$

počet kódových slov – délka kódu, počet všech možných informačních slov

$$L = 2^k.$$

4. Co je lineární kód? Jak určíte jeho kódovou vzdálenost, máte-li k dispozici pouze seznam kódových slov?

Všechny kódová slova lineárního kódu se musejí dát zapsat jako lineární kombinace ostatních kódových slov, musí vždy obsahovat nulový vektor jako slovo. *Kódová vzdálenost* daného kódu se určí jako nejmenší počet rozdílných bitů mezi dvěma vygenerovanými slovy nebo jako minimální nenulová váha ze všech kódových slov.

Hammingova vzdálenost $d(\boldsymbol{b}_i, \boldsymbol{b}_j)$ dvou kódových vektorů $\boldsymbol{b}_i$ a $\boldsymbol{b}_j$ je počet souřadnic ve kterých se navzájem liší. *Hammingova váha* $wt(\boldsymbol{b})$ slova $\boldsymbol{b}$ je rovna počtu souřadnic rovných 1 v tomto slovu. Platí $d(\boldsymbol{b}_i, \boldsymbol{b}_j) = wt(\boldsymbol{b}_i \oplus \boldsymbol{b}_j)$. *Hammingova vzdálenost kódu* je nejmenší ze vzdáleností $k_{vzd} = d_{min} = d(\boldsymbol{b}_i, \boldsymbol{b}_j)$, kde $i \neq j$.

Věta: Je-li $\mathbb{H}$ kontrolní matice lineárního kódu $\mathcal{K}$, pak platí, že vzdálenost kódu $\mathcal{K}$ se rovná d právě tehdy když je libovolných $d-1$ řádků matice $\mathbb{H}$ lineárně nezávislých a existuje d řádků v matici $\mathbb{H}$, které jsou lineárně závislé.

5. Co jsou cyklické kódy? Patří mezi lineární kódy? Na jaké typy chyb jsou orientovány?

Cyklické kódy jsou lineární kódy, které mají tu důležitou vlastnost, že je-li

$$\boldsymbol{b} = (v_{n-1}, v_{n-2}, \ldots, v_0)$$

kódovým vektorem, je kódovým vektorem i vektor

$$\boldsymbol{b}' = (v_{n-2}, v_{n-3}, \ldots, v_0, v_{n-1}).$$

Slouží hlavně k detekci chyb, jejich význačnou vlastností je schopnost *zabezpečovat shluky chyb.*

Celý princip kódování a dekódování je následující – data určená k odeslání se považují za polynom. Ten se vydělí jiným předem dohodnutým generujícím polynomem a zbytek po dělení se připojí ke zprávě jako zabezpečující údaj. Příjemce pak opakuje stejný postup a dívá se, zda dostal stejný zbytek.

6. Kód (n, k) je rozšířený Hammingův kód. Najděte minimální n a k tak, aby k bylo alespoň jeden tisíc. Jakou má kódovou vzdálenost?

Rozšířený Hammingův kód je doplněn o paritu celého kódového slova a má k_{vzd} o jedna vyšší než nerozšířený kód, tj. $k_{vzd} = 4$. Pro počet kontrolních bitů r platí: $n = 2^{r-1}$, $k = n - r = 2^{r-1} - r$. V tabulce jsou vidět parametry pro kódy s různým r. Tento kód opravuje všechny jednoduché chyby a je schopen dvě chyby detekovat.

r	n	k
r	2^{r-1}	$2^{r-1} - r$
3	4	1
4	8	4
5	16	11
6	32	26
7	64	57
⋮	⋮	⋮
10	512	502
11	**1024**	**1013**
12	2048	2036

7. Kód (n, k) je Hammingův kód. Najděte minimální n a k tak, aby k bylo alespoň jeden tisíc. Jakou má kódovou vzdálenost?

Tento kód opravuje všechny jednoduché chyby (Single Error Correction). Jeho kódová vzdálenost je vždy $k_{vzd} = 3$.
Pro počet kontrolních bitů r platí: $n = 2^r - 1$, $k = n - r = 2^r - 1 - r$. V tabulce jsou vidět parametry pro kódy s různým r.

r	n	k
r	$2^r - 1$	$2^r - r - 1$
2	3	1
3	7	4
4	15	11
5	31	26
⋮	⋮	⋮
9	511	502
10	**1023**	**1013**
11	2047	2036

8. Co umožňuje kód s kódovou vzdáleností 5? Uveďte všechny možnosti a označení.

Detekce – 100% detekce všech detekovatelných chyb *dch* je zaručena pokud: $k_{vzd} \geq dch + 1$

Oprava – 100% korekce všech detekovatelných chyb *och* je zaručena pokud: $k_{vzd} > och + dch, k_{vzd} \geq 2 \times dch + 1$

Možnosti kódu s kódovou vzdáleností 5:

Umožuje detekovat maximálně 4 chyby (bez možnosti jejich opravy), či jednu chybu opravit a tři detekovat nebo opravit 2 chyby (platí $och \leq dch$). Viz následující tabulka.

dch	4	3	2
och	0	1	2
1 chyba	detekce	korekce	korekce
2 chyby	detekce	detekce	korekce
3 chyby	detekce	detekce	—
4 chyby	detekce	—	—
Označení	QED	TED-SEC	DED-DEC

D. Operační kód

1. Co jsou adresy řádu 0, 1, 2 a 3? Jak se jinak nazývají adresy řádu 0 a jak adresy řádu 1? Jak se souhrnně nazývají adresy řádu 2 a vyšších řádů?

   Adresa 0. řádu není adresa v pravém slova smyslu, ale je to *přímý operand*. Adresa, která má být skutečně použita, se nazývá *efektivní adresa*. Instrukce, obsahující přímo efektivní adresu se nazývá *přímá adresa*, ojediněle též *adresa prvního řádu*. Adresa druhého řádu je adresa, jejíž obsah je adresou prvního řádu. Adresa n-tého řádu je adresa, jejíž obsah je adresou řádu $n-1$. Adresy n-tého řádu, $n > 1$ se nazývají nepřímé adresy.

2. Co jsou autorelativní adresy a jakou mají výhodu? Jak se liší od skládaných (složených) adres? (Ty neztotožňujte sbázovanými adresami!)

   Autorelativní (samorelativní) adresa. Efektivní adresa se určí tak, že se sečte adresa a uvedená v instrukci s obsahem programového čítače. Přitom bývá možné, aby adresa a reprezentovala i záporné číslo. Je-li počet bitů adresy a stejný jako počet bitů efektivní adresy, stačí při sčítání ignorovat přeplnění. Adresa a však bývá kratší než efektivní adresa. Pak se pro zobrazení záporných čísel používá zpravidla doplňk. kód. Před vlastním sčítáním je pak třeba rozšířit řádovou mřížku sčítance a. Používají se v případech, kdy je žádoucí zkrátit alespoň u některých instrukcí délku adresní části a je pravděpodobné, že efektivní adresa bude dostatečne blízko instrukci.

3. Co jsou univerzální registry (angl. general registers)? Které funkce zastávají (k čemu slouží)?

   Univerzální registr je možné použít pro různé účely, např. jako střadače, indexregistry nebo registry báze.

4. Co je návratová adresa – který registr ji obsahuje? Kam se ukládá při skoku do podprogramu, pokud nemá procesor registr „ukazatel zásobníku“ (registr SP)? Uveďte aspoň dvě možnosti.

Je to adresa instrukce umístěné za instrukcí skok do podprogramu. Ukládá se do programového čítače. Pokud procesor nemá SP, tak se adresa uloží do hlavní paměti (i když v ní není simulována LIFO) nebo univerzálního registu.

5. V čem spočívá obsluha přerušení na úrovni technického vybavení (HW obsluha)?

   Přestane se provádět původní sekvence instrukcí a začne se provádět jiná – tzv. rutina přerušení, začínající na přerušovací adrese. Ukládají se informace o přerušení programu, popř. informace blíže specifikující přerušení – po zpracování přerušení se případně pokračuje v programu, jako by k přerušení nedošlo.

6. Co je maska přerušení? Jak se řeší problém současného výskytu dvou nebo více příčin přerušení?

   Je to registr, jehož každému bitu je přiřazena jedna nebo více příčin přerušení. Všechny příčiny přiřazeny nulovým bitům masky říkáme, že jsou zamaskovány, ostatní odmaskovány. V případě současného výskytu více příčín se přerušení obslouží v pořadí podle priority.

E. Řadiče

1. Co je základní cyklus počítače (instrukční cyklus) a jaké základní akce se v něm provádějí?

   Přečtení instukce (obsah programového čítače se přivede na adresový vstup hlavní paměti, vyšle se signál pro čtení, atd.) ⟶ dekódování operačního znaku ⟶ provedení instrukce ⟶ test na přerušení (může být i na začátku cyklu, v případě přerušení se začne provádět jeho obsluha).
   Vše se opakuje v cyklu, dokud nedojde k zastavení.

2. Jaké jsou základní registry řadiče počítače? Kde je uložena adresa instrukce, která se má provést? Kdy a jak se mění?

   Programový registr – obsahuje adresu instrukce, přiváděné na adresní vstup hlavní paměti, plní funkci *programového čítače*, při spuštění programu se do něj ukládá startovací adresa programu, současně se čtěním nebo ukončení provádění instrukce se ikrementuje a při instrukci skoku;

   registr instrukcí – přesouvá se po přečtení instrukce z výstupu hlavní paměti.

3. Co je klasický (obvodový) řadič? Jaký jiný typ řadiče znáte? Do které skupiny patří řadič s řídicími řetězci a řadič s čítačem?

   Klasický řadič je sestaven z logických obvodových členů. Řadiče s řídícími řetězci a řadiče s čítačem patří do skupiny klasických řadičů. Jiným typem řadiče je mikroprogramovaný řadič.

4. Co je mikroprogramovaný řadič? Jaké typy těchto řadičů znáte? Co je pro ně typické?

Řadič popsaný mikroprogramem (např. assembler hypotetického počítače). Provedení jedné operace se sestává z provedení dílčích operací, které se můžou zadat jako mikroinstrukce.

Typy řadičů:

vertikální – má krátké mikroinstrukce (16 b) obdobné instrukcím, jedna mikroinstrukce trvá několik taktů (přečtení, dekódování, provedení), používá mikroprogramového čítače, který je analogický programovému s čímž souvisí nutnost realizovat zvláštní mikroinstrukce pro skoky, řadič je složitý;

horizontální – má dlouhé mikroinstrukce (64 b), řídicí signály se získávají přímo z mikrooperačního znaku, jedna mikroinstrukce trvá jeden takt, má omezenou volbu adresy následující mikroinstrukce, nepoužívá čítač, je poměrně jednoduchý;

diagonální – má dlouhé mikroinstrukce a používá se u něj mikroprog. čítač.

5. Co je řídicí paměť? Co se do ní ukládá? Jak se česky nazývá firmware a co to je?

Paměť mikroprogramu – ukládá se do ní mikroprogram řadiče;

firmware – základní programové vybavení počítače uložené v paměti ROM. Mikroprogramové vybavení – soubor mikroprogramů, kterými je počítač vybaven.

6. Které části obsahuje horizontálně orientovaná mikroinstrukce a jaký mají význam?

Mikrooperační znak	Adresa následující instrukce	Výběr podmínky