W11 – Kody nadmiarowe, zastosowania w transmisji danych

Henryk Maciejewski

Marek Woda

Plan wykładu

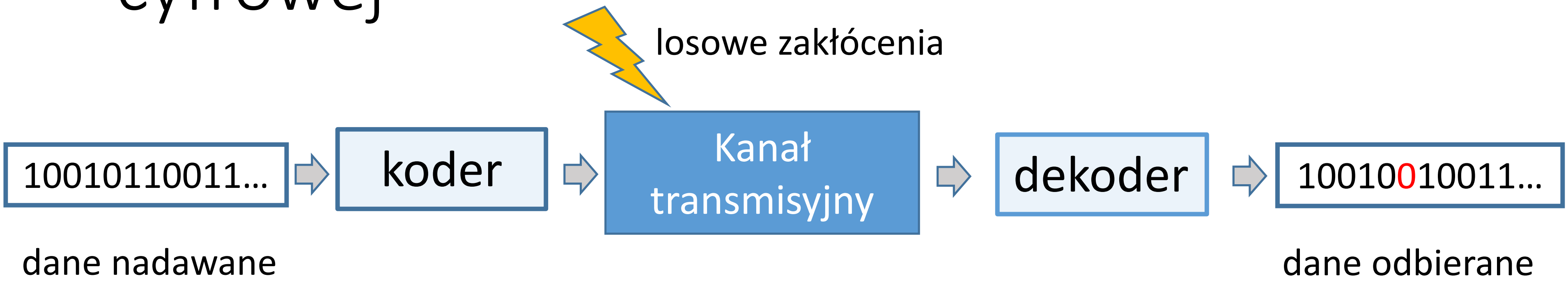
1. Kody nadmiarowe w systemach transmisji cyfrowej

2. Typy kodów, własności

3. Kody blokowe

Np. Władysław Mochnacki, Kody korekcyjne i kryptografia.

Kody nadmiarowe w systemie transmisji cyfrowej



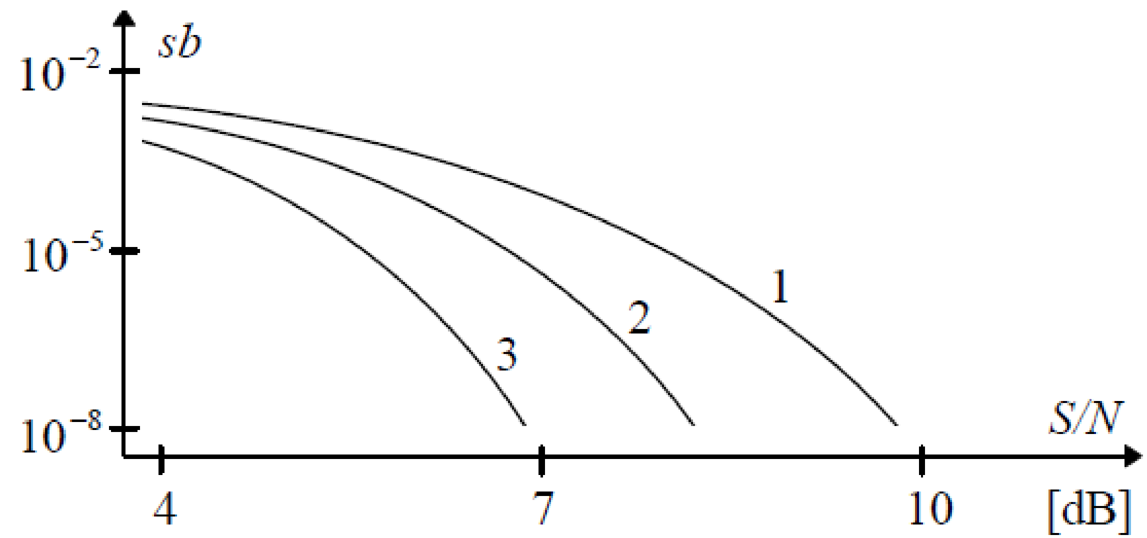
- **Elementowa stopa błędów** (**BER** - bit error rate) – prawdopodobieństwo przekłamania bitu w czasie transmisji = liczba bitów odebranych błędnie / liczba bitów przesyłanych
- BER ~ 10^{-2} – 10^{-5} w typowych kanałach;
  wymagania na system transmisji danych BER ~ 10^{-6} – 10^{-9}.
- **Kody nadmiarowe** – metoda dołączania dodatkowych bitów do danych nadawanych w celu zabezpieczenia danych przed błędami transmisji

Pale Blue Dot

Voyager telecommunication

Typowa zależność stopy błędów (sb=BER) od S/N

S – moc sygnału, N – moc szumu



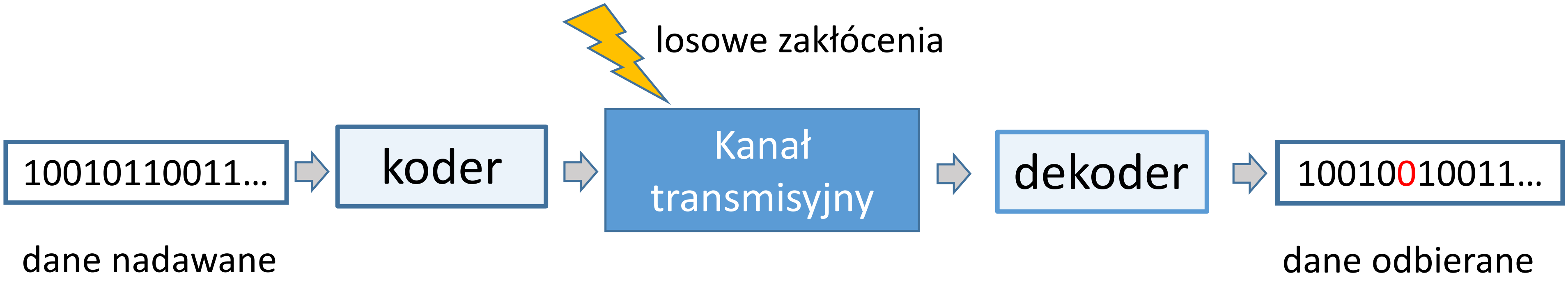
przesłano z nadmiarem
początkowo bitów

1 – kanał bez korekcji

2 – korekcja kodem Hamminga(15,11), zdolność korekcyjna t=1

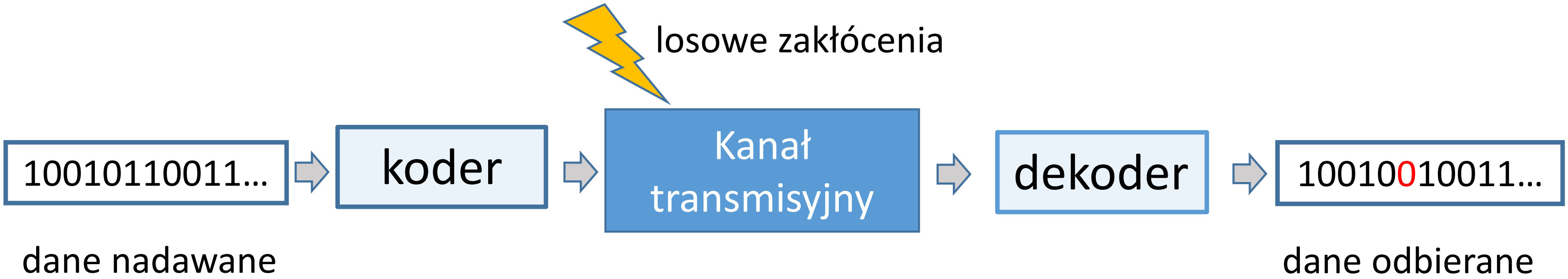
3 – korekcja kodem BCH(127,64), zdolność korekcyjna t=10

Zastosowania kodów nadmiarowych



- **ARQ** – Automatic Repeat Request
  - koder dodaje informację nadmiarową do bloku danych
  - dekoder sprawdza czy pakiet został przesłany poprawnie, jeśli nie – wysyłane jest żądanie ponownej transmisji bloku (kanał zwrotny)
  - **kod detekcyjny** – np. bit parzystości dodawany do bloku o długości n

Zastosowania kodów nadmiarowych



- **FEC** – Forward Error Correction
  - dekoder wykorzystuje informację nadmiarową do skorygowania błędów
  - kod korekcyjny np. potrajanie bitów (0 – 000, 1 – 111)
  - dekoder stosuje algorytm głosujący

- Systemy hybrydowe – ARQ z FEC w celu zmniejszenia liczby retransmisji

Typy kodów nadmiarowych

- **Kody blokowe**
  - ciąg informacyjny dzielony jest na bloki k-elementowe
  - do każdego bloku dołączana jest sekwencja kontrolna – powstaje <u>słowo kodowe</u> (<u>wektor kodowy</u>)
- **Kody splotowe** (rekurencyjne) – brak podziału na bloki, kodowanie z wykorzystaniem rejestru przesuwnego
- **Kody systematyczne** / niesystematyczne
  - Kod systematyczny - pierwsze k bitów w słowie kodowym to ciąg informacyjny
- **Kody liniowe** – suma dowolnych dwóch wektorów kodowych jest wektorem kodowym
- **Kody cykliczne** – tworzone za pomocą pierścieni wielomianów nad ciałami skończonymi (też są kodami linowymi)

CRC

Kody blokowe

Ciąg informacyjny dzielony jest na bloki k-elementowe,

Słowo kodowe n-elementowe, n > k

$$\underbrace{\boxed{\;k\;}\boxed{\;n-k\;}}_{n}$$

Kody blokowe oznaczamy symbolem **(n,k)**

Sprawność kodu R = k/n

Nadmiar kodowy = 1-R = (n-k)/n

0101... → $\boxed{\text{Koder}}$ → kanał transmisyjny → $\boxed{\text{Dekoder}}$ jeśli odszyfra każdy kod to bezbłędna transmisja

k

2^k możliwych wektorów
ale będzie 2^n bo n bitów kodujemy
więc część nie używana

2^n ciągów n bitowych

Kody blokowe

Wyjście kodera: $\mathbf{2^k}$ różnych wektorów kodowych

Wejście dekodera: $\mathbf{2^n}$ różnych wektorów (*przestrzeń wektorowa nad ciałem binarnym – GF*(2)) ← Galois
(w tym 2^k – wektorów kodowych, 2^n-2^k – niekodowych)

Po przejściu przez kanał transmisyjny wektor kodowy może:

1. zostać niezmieniony (brak zakłócenia)
2. zostać zmieniony na wektor niekodowy
3. zostać zmieniony na inny wektor kodowy

Przypadek 2 – błąd transmisji wykrywalny (korygowalny? – dekoder może znaleźć wektor kodowy różniący się od odebranego najmniejszą liczbą pozycji)

Przypadek 3 – niewykrywalny błąd transmisji

Zdolność korekcyjna i detekcyjna kodu

Odległość Hamminga: **$d_H(u,v)$** pomiędzy wektorami kodowymi u,v = liczba pozycji, na których u, v różnią się.

Np.

u = [1,1,0,1,0,0,1]

v = + [1,0,0,1,0,1,1] $d_H(u,v) = 2$

[0,1,0,0,0,1,0]

Własność: $d_H(u,v) \le d_H(u,w) + d_H(w,v)$

Waga Hamminga wektora kodowego w(u) = liczba współrzędnych niezerowych

$d_H(u,v) = w(u+v)$

u+v = [0,1,0,0,0,1,0]

Zdolność korekcyjna i detekcyjna kodu

$\cdot - 2^n$
$\cdot - 2^k$
$\cdot$ ✓ kod. w

Odległość minimalna Hamminga pomiędzy wektorami kodowymi, ozn. **d** – decyduje o zdolności detekcyjnej i korekcyjnej kodu blokowego.

(zachodzi dla kodu liniowego:
d = min waga niezerowego wektora kodowego) (dlaczego?)

$d=3$

$d_{min} = \min\{d(u,v) : u,v \in \mathbb{K} \wedge u \neq v\} = \min\{w(u+v) : u,v \in \mathbb{K} \wedge u \neq v\}$ kod liniowy $u+v=w$, w - kod liniowy

<u>Zdolność detekcyjna</u> = $d - 1$

(kod o parametrze d może wykryć ciągi błędów o wadze ≤ d-1)

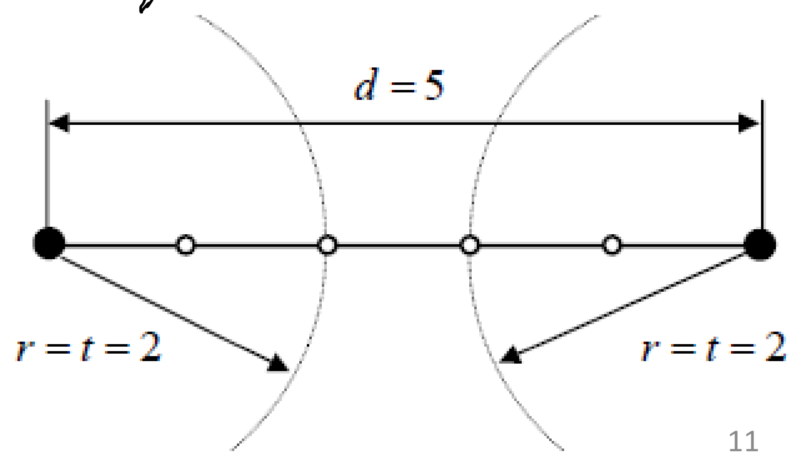
bo po zmianie d bitów trafiamy na inny wektor kodowy

<u>Zdolność korekcyjna</u>

bo korekcja do najbliższego kodu będzie korygować do dobrego

$$t = \left\lfloor \frac{d-1}{2} \right\rfloor$$

jak więcej to koryguje do kolejnego



Zapis macierzowy kodu liniowego

m_{1xk} – blok danych

$c_{X\ 1xn}$ – wektor kodowy

G_{kxn} – macierz generująca kod

$$c_X = m \cdot G$$

$G = [I_{kxk} \mid P_{kx(n-k)}]$ macierz wektorów niezmienionych /definicji kodu
aka równania kontrolne

I_{kxk} – macierz jednostkowa

G – macierz o wierszach liniowo niezależnych;
kod liniowy to zbiór wszystkich liniowych kombinacji wierszy macierzy G

Zapis macierzowy kodu liniowego

$$\boldsymbol{c}_X = \boldsymbol{m} \cdot \boldsymbol{G} = [m_1, m_2, \ldots, m_k] \cdot \begin{bmatrix} \boldsymbol{g}_1 \\ \boldsymbol{g}_2 \\ \ldots \\ \boldsymbol{g}_k \end{bmatrix} = m_1\boldsymbol{g}_1 + m_2\boldsymbol{g}_2 + \ldots + m_k\boldsymbol{g}_k$$

Wniosek – każdy wektor kodowy jest liniową kombinacją wierszy macierzy generującej kod

Wiersze macierzy generującej kod – to zbiór <u>wektorów bazowych kodu</u>.

Zapis macierzowy kodu liniowego

$$c_X = \boldsymbol{m} \cdot \boldsymbol{G} = [m_1, m_2, \ldots, m_k] \cdot \begin{bmatrix} \boldsymbol{g_1} \\ \boldsymbol{g_2} \\ \ldots \\ \boldsymbol{g_k} \end{bmatrix} = m_1\boldsymbol{g_1} + m_2\boldsymbol{g_2} + \ldots + m_k\boldsymbol{g_k}$$

Przykład kodu (5,2):

$$\boldsymbol{G} = \left[\begin{array}{cc|ccc} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{array}\right]$$

$$[a_1, a_2] \cdot G = [a_1, a_2, a_1, a_2, a_1+a_2]$$

Zapis kodu przy pomocy <u>równań kontrolnych</u> (n-k liniowo niezależnych równań):

$$a_3 = a_1$$
$$a_4 = a_2$$
$$a_5 = a_1 + a_2$$

Macierz kontrolna (macierz parzystości)

Dla każdej macierzy generującej kod istnieje G_{kxn} istnieje <u>macierz kontrolna</u> $H_{(n-k)xn}$ tż. $G \cdot H^T = 0$

$$G = [I_k \mid P_{kx(n-k)}]$$

$$H = [\, P^T_{(n-k)xk} \mid I_{n-k} \,]$$

Przykład kodu (5,2):

transponowanie

$$\boldsymbol{G} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad \boldsymbol{H}^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Macierz kontrolna – dekodowanie

c_X – wyjście kodera

$c_Y = c_X + e$ – wejście dekodera (e – wektor błędów)

s – <u>syndrom błędu</u> (zależy tylko od e)

$$\boldsymbol{s} = (\boldsymbol{c}_X + \boldsymbol{e})\boldsymbol{H}^T = \boldsymbol{c}_X\boldsymbol{H}^T + \boldsymbol{e}\boldsymbol{H}^T = \boldsymbol{m}\underbrace{\boldsymbol{G}\boldsymbol{H}^T}_{0} + \boldsymbol{e}\boldsymbol{H}^T = \boldsymbol{e}\boldsymbol{H}^T$$

$$\boldsymbol{s} = \boldsymbol{e}\boldsymbol{H}^T = [e_1\ e_2 \ldots e_n] \cdot \begin{bmatrix} \boldsymbol{h}_1^T \\ \boldsymbol{h}_2^T \\ \ldots \\ \boldsymbol{h}_n^T \end{bmatrix} = e_1\boldsymbol{h}_1^T + e_2\boldsymbol{h}_2^T + \ldots + e_n\boldsymbol{h}_n^T$$

Syndrom – liniowa kombinacja wierszy macierzy H^T

Macierz kontrolna – dekodowanie

Przykład kodu (5,2):

$c_Y = [1,1,0,1,0]$

$$H^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$s = c_Y H^T = [1\,1\,0\,1\,0] \cdot \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [1\,0\,1] + [0\,1\,1] + [0\,1\,0] = [1\,0\,0]$$

stąd $e = [0,0,1,0,0]$ (bo $s=eH^T$)

więc wektor skorygowany:

$$c_Y + e = [1,1,0,1,0] + [0,0,1,0,0] = [\boxed{1,1},\underset{a_1}{\color{red}{1}},\underset{a_2}{1},\underset{a_1+a_2}{0}]$$

Kody Hamminga

Kody (n,k) = $(2^m-1, 2^m-m-1)$,

m – liczba pozycji kontrolnych

np.

m	2	3	4	5	6
(n, k)	(3,1)	(7,4)	(15,11)	(31,26)	(63,57)

d = 3 $\rightarrow$ korygują pojedyncze błędy

(zdolność detekcyjna = 2)

Wykorzystywane np. w pamięciach komputerowych

Kody Hamminga

Przykład kodu Hamminga (7,4)

$$G = \left.\overbrace{\left[\begin{array}{ccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ \multicolumn{4}{c}{\underbrace{0 \quad 0 \quad 0 \quad 1}_{I}} & \multicolumn{3}{c}{\underbrace{0 \quad 1 \quad 1}_{P}} \end{array}\right]}^{7}\right\} 4$$

$[a_1, a_2, a_3, a_4] \cdot G = [a_1, a_2, a_3, a_4, a_5, a_6, a_7]$

$a_5 = a_1 + a_2 + a_3$

$a_6 = a_2 + a_3 + a_4$

$a_7 = a_1 + a_2 + a_4$

Prosta realizacja sprzętowa

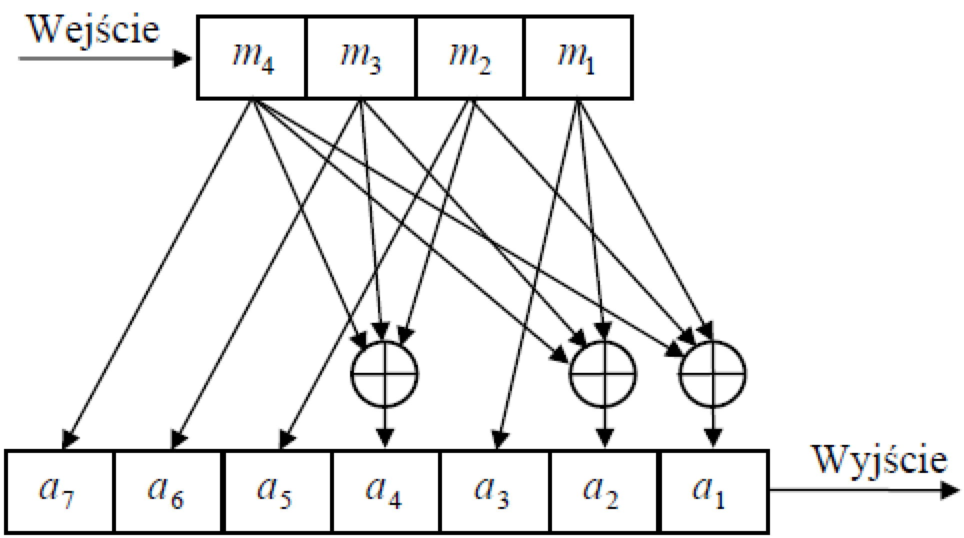
Przykład kodu Hamminga (7,4)

$[a_1, a_2, a_3, a_4] \cdot G = [a_1, a_2, a_3, a_4, a_5, a_6, a_7]$

$a_5 = a_1+a_2+a_3$

$a_6 = a_2+a_3+a_4$

$a_7 = a_1+a_2+a_4$



Kody cykliczne

Kody spełniające własność:

Jeśli $c = [a_{n-1}, a_{n-2}, \ldots, a_1, a_0]$ jest wektorem kodowym, to każde przesunięcie cykliczne c jest również wektorem kodowym.

Np.

$c_1 = [a_{n-2}, \ldots, a_1, a_0, a_{n-1}]$ - jest wektorem kodowym

Istnieją efektywne metody generacji kodów – za pomocą reprezentacji ciągów informacyjnych i wektorów kodowych za pomocą wielomianów

Realizacja koderów / dekoderów za pomocą rejestrów przesuwnych – stosunkowo prosta

Kody cykliczne

Ciągi informacyjne / kodowe zapisujemy w postaci wielomianów nad ciałem GF(2) (współczynniki ze zbioru {0,1}, dodawanie modulo 2):

$c = [a_{n-1}, a_{n-2}, \ldots, a_1, a_0]$

$c(x) = a_{n-1}x^{n-1} + a_{n-2}\,x^{n-2} + \ldots + a_1x + a_0$

Przykład:

$c = 1011$

$c(x) = x^3+x+1$

Kody cykliczne

Własność przesunięcia cyklicznego dla postaci wielomianowej definiuje się następująco:

$c(x)$ – wielomian stopnia n-1 „przesunięcie o i w lewo"

Wówczas reszta z dzielenia $x^i c(x)$ przez x^n-1 jest również wektorem kodowym (dowód – Mochnacki str. 83-84) realizuje przesunięcie

$c = [a_{n-1}, a_{n-2}, \ldots, a_1, a_0]$ $\qquad$ $c_1 = [a_{n-2}, \ldots, a_1, a_0, a_{n-1}]$

$c(x) = a_{n-1}x^{n-1} + a_{n-2}\, x^{n-2} + \ldots + a_1 x + a_0$

$c_1(x) = a_{n-2}x^{n-1} + a_{n-3}\, x^{n-2} + \ldots + a_0 x + a_{n-1}$

$c_1(x) = xc(x)(\mathrm{mod}\ x^n+1)$ $\qquad$ (dowód – Mochnacki str. 83-84)

$c_i(x) = x^i c(x)(\mathrm{mod}\ x^n+1)$

Kody cykliczne – wielomian generujący

g(x) - <u>wielomian generujący</u> kod cykliczny

stopień = n-k = liczba elementów kontrolnych w wektorze kodowym

np. dla kodu cyklicznego Hamminga $(\overset{n}{7},\overset{k}{4})$

$g(x) = \underset{\overset{\uparrow}{n-k}}{x^3} + x^2 + 1$ (podzielnik wielomianu $x^{\overset{n}{\downarrow}7}+1$)

g(x) pozwala wyznaczyć wektor kodowy dla kodu (n,k)

CRC 32 - np do transmisji ethernet

Algorytm kodowania – kod (n,k)

g(x) - wielomian generujący kod cykliczny stopnia n-k

m(x) – wielomian odpowiadający ciągowi informacyjnemu np. $1011 = x^3 + x + 1$

$\mathbf{c_X(x)}$ – wielomian odpowiadający wektorowi kodowemu

$$\mathbf{c_X(x) = x^{n-k} m(x) + r(x)}$$

przesuwamy o n-k pozycji

Gdzie r(x) jest resztą z dzielenia $x^{n-k}m(x)$ przez g(x):

$x^{n-k}m(x) = q(x)\, g(x) + r(x)$

stąd – **każdy wektor kodowy jest podzielny przez g(x)**

Algorytm kodowania – kod (n,k)

1. Przesuń ciąg informacyjny n-k pozycje w lewo: $x^{n-k}m(x)$
2. Wyznacz resztę z dzielenia $x^{n-k}m(x)$ przez g(x):

$$r(x) = r_{n-k-1}x^{n-k-1} + \ldots + r_1 x + r_0$$

3. Dopisz resztę r(x) na n-k najmłodszych pozycjach ciągu $x^{n-k}m(x)$:

$$\mathbf{c_X(x) = x^{n-k}m(x) + r(x)}$$

Dekodowanie – kod (n,k)

Otrzymany ciąg:

$c_Y(x) = c_X(x) + e(x)$

$e(x)$ – wielomian odpowiadający wektorowi błędów

Korzystamy w własności, że każdy wektor kodowy jest podzielny przez $g(x)$

Wyznaczamy resztę z dzielenia $c_Y(x)$ przez $g(x)$:

$$c_Y(x) = m(x)g(x) + r(x)$$

jeśli $r(x) = 0$ – brak błędu

jeśli $r(x) \neq 0$ – wystąpiły błędy

(algorytm korekcji – patrz np. Mochnacki)

Przykładowe kody cykliczne

Standardy CRC – np.

CRC-8 (WCDMA): $x^8+x^7+x^4+x^3+x^2+1$

CRC-16 (CCITT): $x^{16}+x^{12}+x^5+1$

CRC-40 (GSM): $(x^{23}+1)(x^{17}+x^3+1)$

CRC-64

https://en.wikipedia.org/wiki/Cyclic_redundancy_check

Kody Bose–Chaudhuri–Hocquenghema (BCH)

Parametry:

- długość wektora kodowego $n = 2^m - 1$
- liczba pozycji kontrolnych $n-k \leq mt$
- min. odległość Hamminga $d \geq 2t + 1$

Kod koryguje do t błędów i ma nie więcej niż mt elementów kontrolnych

Kody Hamminga są podzbiorem kodów BCH
(BCH z $t=1$ jest kodem Hamminga)

Kody BCH - parametry

ile dużych jakiej długości ile kodowych ile błędów znajdzie

takie coś będziemy badać!

m	*n*	*k*	*t*	*m*	*n*	*k*	*t*	*m*	*n*	*k*	*t*	*m*	*n*	*k*	*t*
3	7	4	1	6	63	10	13	7	127	15	27	8	255	123	19
4	15	11	1			7	15			8	31			115	21
		7	2	7	127	120	1	8	255	247	1			107	22
		5	3			113	2			239	2			99	23
5	31	26	1			106	3			231	3			91	25
		21	2			99	4			223	4			87	26
		16	3			92	5			215	5			79	27
		11	5			85	6			207	6			71	29
		6	7			78	7			199	7			63	30
6	63	57	1			71	9			191	8			55	31
		51	2			64	10			187	9			47	42
		45	3			57	11			179	10			45	43
		39	4			50	13			171	11			37	45
		36	5			43	14			163	12			29	47
		30	6			36	15			155	13			21	55
		24	7			29	21			147	14			13	59
		18	10			22	23			139	15			9	63
		16	11							131	18				

p. W. Mochnacki – Kody korekcyjne …

Kody Reeda-Solomona

Podklasa kodów BCH niebinarnych, nad GF(q) – ciałem rozszerzonym

np. GF(8) – rozszerzenie 3-stopnia nad GF(2)

ogólnie: $q=2^m$ rozszerzenie m-tego stopnia nad GF(2)

Kod RS(n,k) nad ciałem GF(q) ma własności:

$n=q-1$	długość wektora kodowego
$r=n-k$	pozycji kontrolnych
$d=r+1$	odległość minimalna

Kody Reeda-Solomona

Kody RS służą do korygowania błędów grupowych

nie zmienia bitu a całe słowo na „poprawne"

Idea

- Każdy element GF(q) jest ciągiem m-elementowym nad GF(2)
- Kod RS(q-1, q-1-2t) - koryguje t błędów, które pojawią się w bloku m-elementowym (czyli najprawdopodobniej błąd grupowy)

Ciało binarne GF(2)

$<\{0,1\}, +, \cdot >$

tabliczka dodawania i mnożenia

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1