

W11 – Kody nadmiarowe, zastosowania w transmisji danych

Henryk Maciejewski

Marek Woda

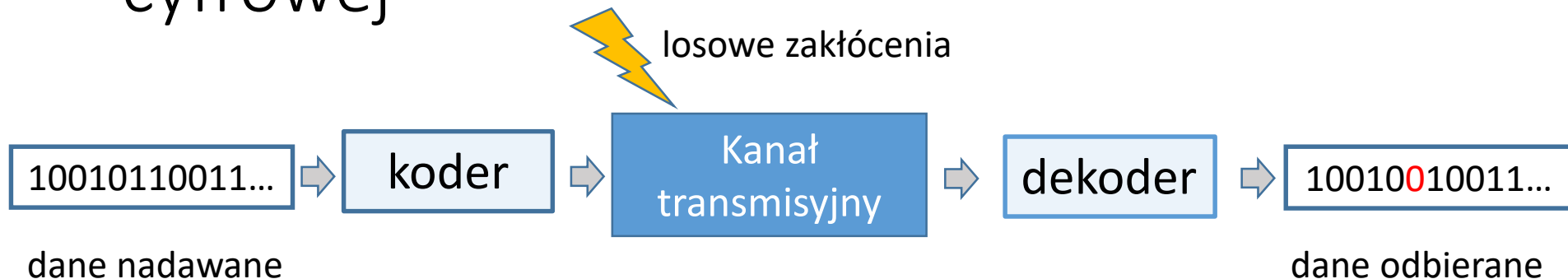
Plan wykładu

1. Kody nadmiarowe w systemach transmisji cyfrowej
2. Typy kodów, własności
3. Kody blokowe

Np. Władysław Mochnacki, Kody korekcyjne i kryptografia.

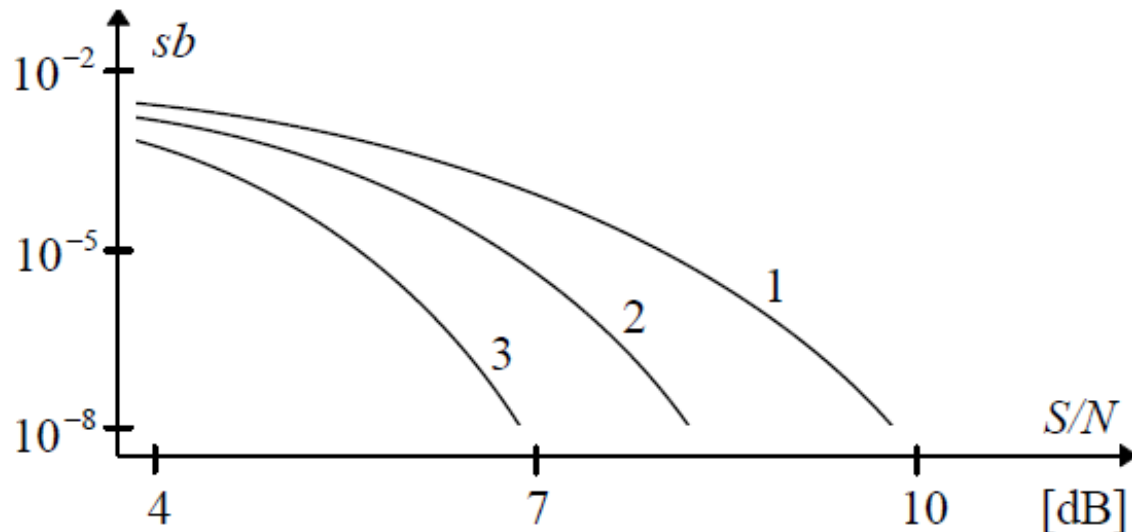


Kody nadmiarowe w systemie transmisji cyfrowej



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

Typowa zależność stopy błędów ($s_b = \text{BER}$) od S/N
 S – moc sygnału, N – moc szumu



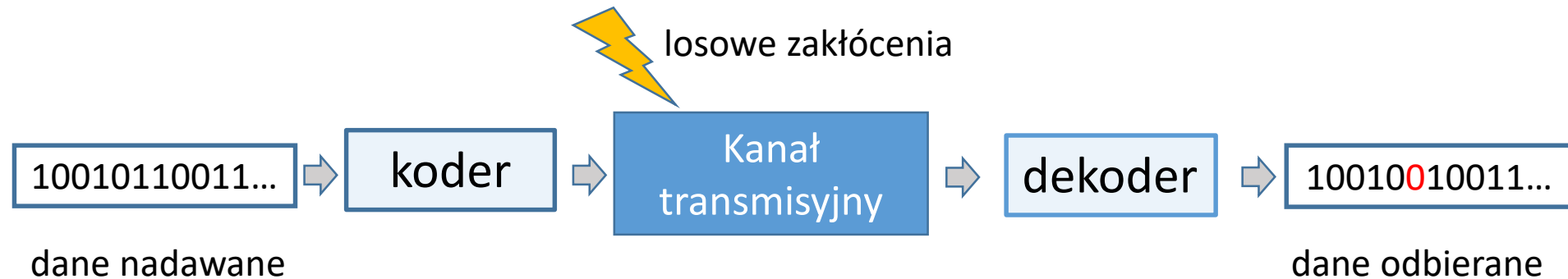
1 – kanał bez korekcji

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

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

przekład z nadniewem
pozwolawo bitów

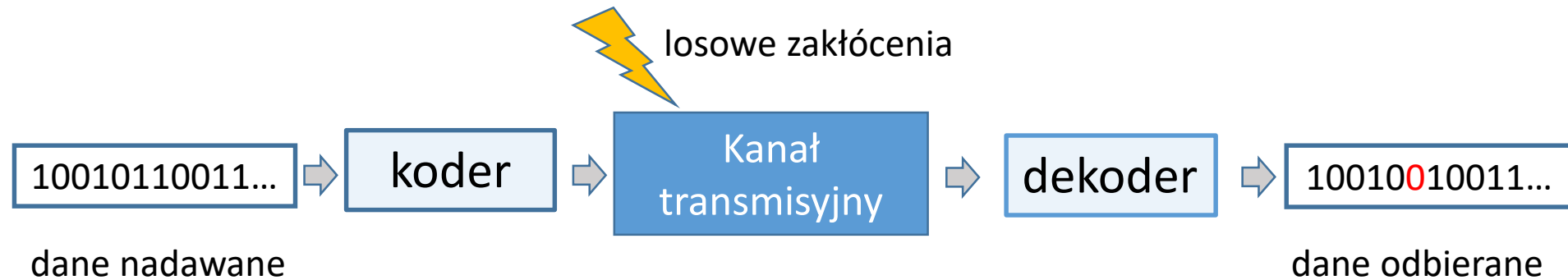
Zastosowania kodów nadmiarowych



- **ARQ – Automatic Repeat Request**

- koder dodaje informację nadmiarową do bloku danych
- dekodek 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



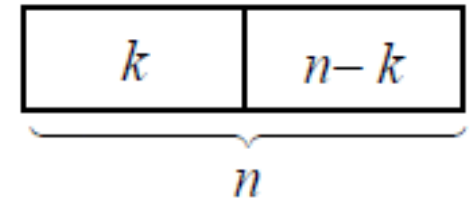
- **FEC** – Forward Error Correction
 - dekodek wykorzystuje informację nadmiarową do skorygowania błędów
 - kod korekcyjny np. potrójanie bitów (0 – 000, 1 – 111)
 - dekodek stosuje algorytm głosujący
- Systemy hybrydowe – ARQ z FEC w celu zmniejszenia liczby retransmisji

Typy kodów nadmiarowych

- **Kody blokowe**
 - ciąg informacyjny dzielony jest na bloki k-elementowe
 - do każdego bloku dołączana jest sekwencja kontrolna – powstaje słowo kodowe (wektor kodowy)
- **Kody splotowe** (rekurencyjne) – brak podziału na bloki, kodowanie z wykorzystaniem rejestru przesuwne
- **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)

Kody blokowe

Ciąg informacyjny dzielony jest na bloki k -elementowe,
Słowo kodowe n -elementowe, $n > k$



Kody blokowe oznaczamy symbolem (n,k)

Sprawność kodu $R = k/n$

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

0101... \rightarrow **Koder** $\xrightarrow{\text{kanal transmisyjny}}$ **Dehoder**

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

2^n ciągów
 n bitów

jeśli odejśću logarytmy kod to
bierzećna transmisja

Kody blokowe

Wyjście koder: 2^k różnych wektorów kodowych

Wejście dekoder: 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? – dekodery 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.

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

$$v = [1, 0, 0, 1, 0, 1, 1]$$

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

$$d_H(u,v) = 2$$

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

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

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

$$u+v = [0, 1, 0, 0, 0, 1, 0]_{10}$$

Zdolność korekcyjna i detekcyjna kodu

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?)

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

Zdolność detekcyjna = **d - 1**

(kod o parametrze d może wykryć ciągi błędów o wadze $\leq d-1$)

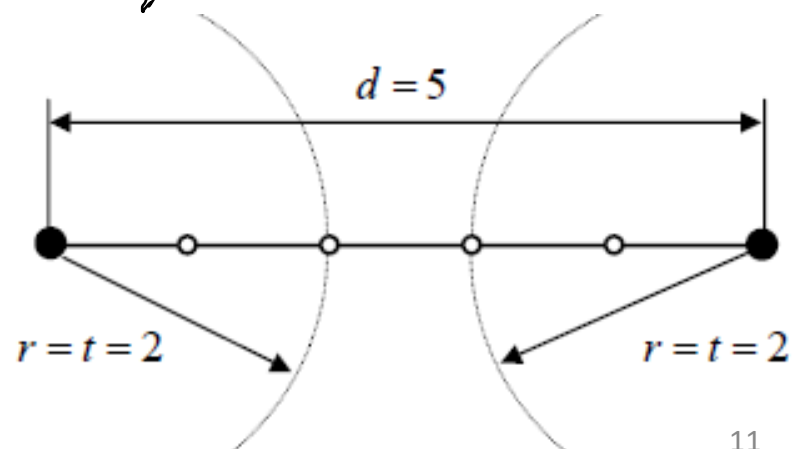
bo po zmianie d bitów trafiamy na inny wektor kodowy

Zdolność korekcyjna

bo korekują do najbliższego kodu będąc korygowani do dobrego

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

jak więcej to koryguje do kolejnego



Zapis macierzowy kodu liniowego

$m_{1 \times k}$ – blok danych

$c_{1 \times n}$ – wektor kodowy

$G_{k \times n}$ – macierz generująca kod

$$c_X = m \cdot G$$

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

$I_{k \times k}$ – macierz jednostkowa

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

Zapis macierzowy kodu liniowego

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

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

Wiersze macierzy generującej kod – to zbiór wektorów bazowych kodu.

Zapis macierzowy kodu liniowego

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

Przykład kodu (5,2):

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

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

Zapis kodu przy pomocy równań kontrolnych (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 $G_{k \times n}$ istnieje macierz kontrolna $H_{(n-k) \times n}$ tż. $G \cdot H^T = 0$

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

$$H = [P_{(n-k) \times k}^T \mid I_{n-k}]$$

Przykład kodu (5,2):

$$G = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}, \quad H = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{bmatrix}, \quad H^T = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0^{15} & 1 \end{bmatrix}$$

transponowanie

Macierz kontrolna – dekodowanie

c_X – wyjście koder

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

s – syndrom błędu (zależy tylko od e)

$$s = (c_X + e)H^T = c_X H^T + eH^T = m \underbrace{GH^T}_{\emptyset} + eH^T = eH^T$$

$$s = eH^T = [e_1 \ e_2 \ \dots \ e_n] \cdot \begin{bmatrix} h_1^T \\ h_2^T \\ \dots \\ h_n^T \end{bmatrix} = e_1 h_1^T + e_2 h_2^T + \dots + e_n h_n^T$$

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:

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

$\uparrow \quad \uparrow \quad \uparrow$
 $a_1 \quad a_2 \quad a_1 + a_2$

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)

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

Handwritten annotations: A brace above the matrix spans all 7 columns and is labeled '7'. A brace to the right of the matrix spans all 4 rows and is labeled '4'. A brace below the first four columns is labeled 'k' (information part), and a brace below the last three columns is labeled 'p' (parity part).

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

$$a_5 = a_1 + a_2 + a_3$$

$$a_6 = a_2 + a_3 + a_4$$

$$a_7 = a_1 + a_2 + a_4$$

Prosta realizacja sprzętowa

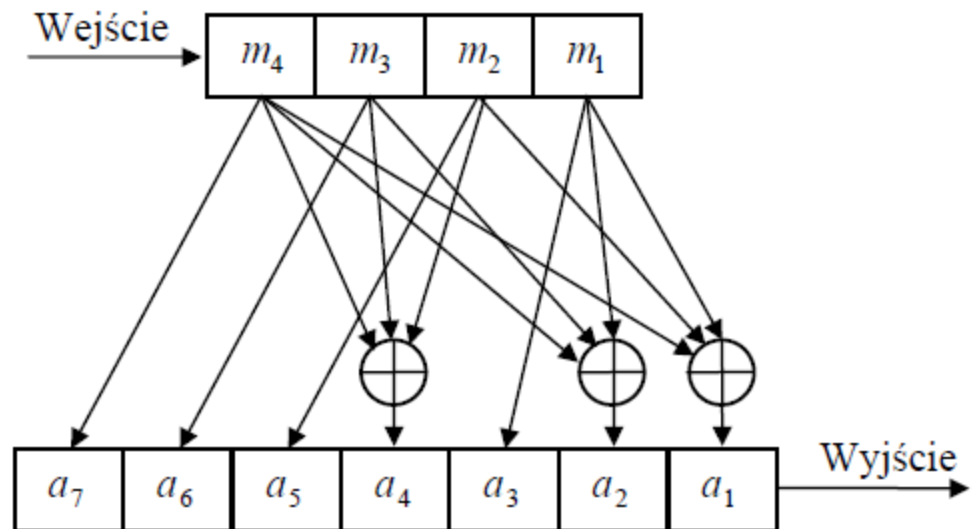
Przykład kodu Hamminga (7,4)

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

$$a_5 = a_1 + a_2 + a_3$$

$$a_6 = a_2 + a_3 + a_4$$

$$a_7 = a_1 + a_2 + a_4$$



Kody cykliczne

Kody spełniające własność:

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

Np.

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

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

Realizacja koderów / dekodekoderów 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):

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

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

Przykład:

$$c = 1011$$

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

Kody cykliczne

Własność przesunięcia cyklicznego dla postaci wielomianowej definiuje się następująco:

$c(x)$ – wielomian stopnia $n-1$

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

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

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

$$c_1(x) = a_{n-2}x^{n-1} + a_{n-3}x^{n-2} + \dots + a_0x + a_{n-1}$$

$$c_1(x) = xc(x) \pmod{x^n + 1} \quad (\text{dowód – Mochnacki str. 83-84})$$

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

Kody cykliczne – wielomian generujący

$g(x)$ - wielomian generujący kod cykliczny

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

np. dla kodu cyklicznego Hamminga (7,4)

$$g(x) = x^3 + x^2 + 1 \quad (\text{podzielnik wielomianu } x^7+1)$$

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

Algorytm kodowania – kod (n,k)

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

$m(x)$ – wielomian odpowiadający ciągowi informacyjnemu

$c_x(x)$ – wielomian odpowiadający wektorowi kodowemu

$$c_x(x) = x^{n-k}m(x) + r(x)$$

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 $g(x)$:

$$r(x) = r_{n-k-1}x^{n-k-1} + \dots + r_1x + r_0$$

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

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

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)

$\langle \{0,1\}, +, \cdot \rangle$

tabliczka dodawania i mnożenia

+	0	1
0	0	1
1	1	0

·	0	1
0	0	0
1	0	1