Wyższa Szkoła Informatyki Stosowanej i Zarządzania

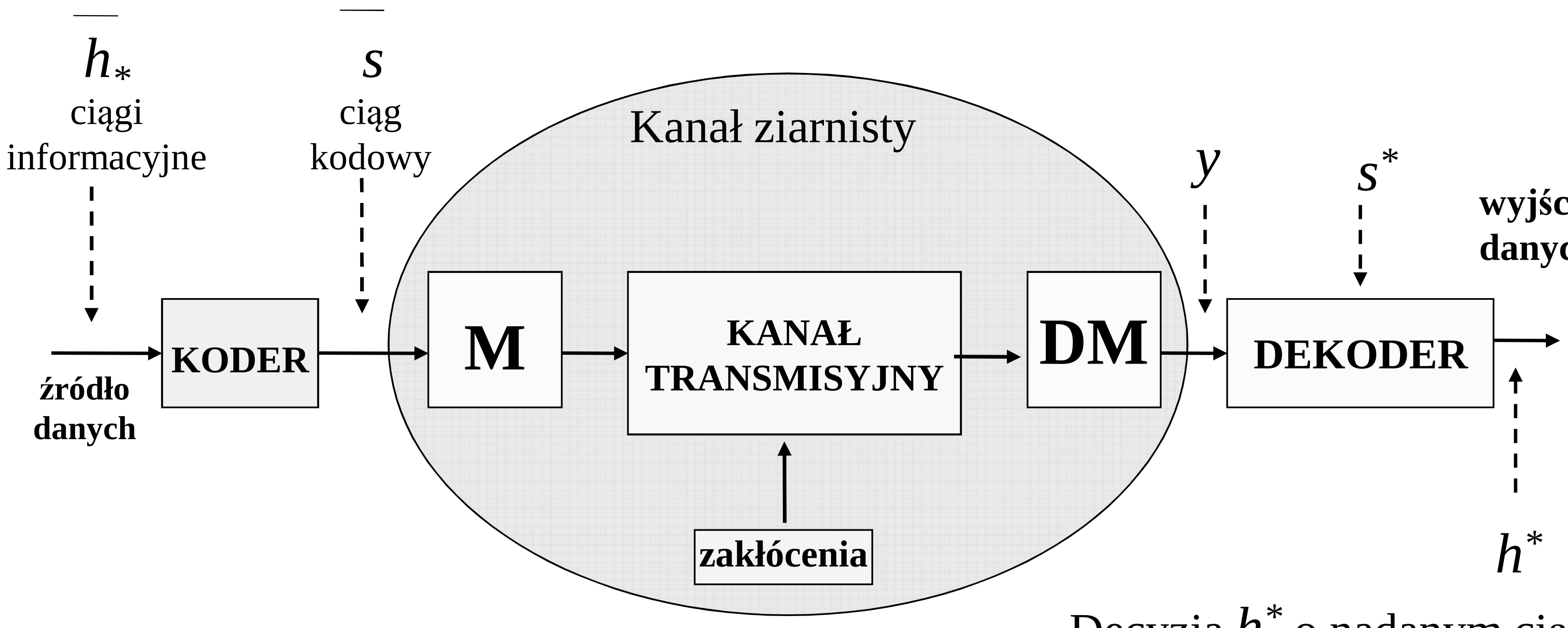
KODOWY SYSTEM TRANSMISJI DANYCH

dr inż. Janusz DUDCZYK

ZAGADNIENIA

- Schemat blokowy, terminologia;
- Parametry kanału ziarnistego, ciągi błędów;
- Kody blokowe, splotowe, liniowe;
- Dekodowanie korekcyjne;
- Koder kodu rozdzielnego;
- Koder kodu nierozdzielnego.

Schemat blokowy kodowego systemu transmisji danych



Decyzja h^* o nadanym ciągu informacyjnym h_*

Pojęcia podstawowe

- Kodowanie - zabezpieczenie transmisji przed błędami;
- Źródło danych - generuje wiadomość dyskretną, której przyporządkowane są ciągi informacyjne. Jest to kodowanie dla źródła;
- Koder - przypisuje ciągom informacyjnym odpowiednie ciągi kodowe /zawierające nadmiar/;
- Modulator - zamienia sygnał cyfrowy na analogowy w zależność od kanału.

Dzięki zakodowaniu danych nadawczych, po stronie odbiorczej na podstawie odebranego ciągu, zostaje podjęta decyzja o nadanym ciągu kodowym.

Pojęcia podstawowe

Szybkość modulacji $\upsilon_m = \frac{1}{\varepsilon}, \quad [\text{bod}]$

czas trwania sygnału elementarnego

Szybkość transmisji $\upsilon_t = \upsilon_m \cdot \lg_2 q \,, \quad [\text{bit/s}]$ q - wartość modulacji

Elementowa stopa błędu $p_e = \frac{N_b}{N_w}$

Stopa błędu:

1. Nie powinna przekraczać, 10^{-10} - 10^{-12};
2. Dla kanału radiowego 10^{-2} (konieczność zabezpieczenia kodowania, co setny bit może zostać przekłamany);
3. Dla linii telefonicznej 10^{-3} - 10^{-4}.

Pojęcia podstawowe

Źródło danych:

- Bezpamięciowe (ciągi od siebie niezależne);
- Generuje *k*-pozycyjne *q*-narne ciągi informacyjne h_i;
- Ciągi h_i są ciągami kodowymi równodostępnego kodu dla źródła;

$$P(\overrightarrow{h_i}) = const = \frac{1}{L}; \quad L = q^k$$

- W dalszych rozważaniach $q = 2$.

Parametry kanału ziarnistego

1. Zbiór sygnałów wejściowych kanału $\{\vec{s_i}\} \ : \quad i = 1,2,\ldots,L;\ L = 2^k$
2. Zbiór sygnałów wyjściowych kanału $\{\vec{y_j}\} \ : \quad j = 1,2,\ldots,k;\ k = 2^n$

$$\left.\begin{matrix}\vec{s_1}\\ \vec{s_2}\\ \cdot\\ \cdot\\ \cdot\\ \vec{s_L}\end{matrix}\right\} \Rightarrow \boxed{\begin{matrix}\text{Kanał}\\ \text{ziarnisty}\end{matrix}} \Rightarrow \left.\begin{matrix}\vec{y_1}\\ \vec{y_2}\\ \cdot\\ \cdot\\ \cdot\\ \vec{y_k}\end{matrix}\right\}$$

k - długość ciągu informacyjnego

n - długość ciągu kodowego

Parametry kanału ziarnistego

Kanał ziarnisty jest całkowicie określony poprzez:

Zbiór możliwych ciągów kodowych $\left\{\overrightarrow{s_i}\right\}$

Zbiór możliwych ciągów odebranych $\left\{\overrightarrow{y_j}\right\}$

Matematyczne związki pomiędzy $\left\{\overrightarrow{s_i}\right\}$ i $\left\{\overrightarrow{y_j}\right\}$

Parametry kanału ziarnistego

Macierz prawdopodobieństw:

$$m = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix} \quad \mathbf{M} = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

Prawdopodobieństwa przejść kanału: $p_{ij} = P(\overrightarrow{y_j} / \overrightarrow{s_i})$

Jeżeli $p_{ij} = p_{ji} = p$ to kanał jest symetryczny i bezpamięciowy (BSK).

Kanał bezpamięciowy – prawdopodobieństwo błędnego odebrania informacji nie zależy od wcześniejszego i późniejszego nadania informacji.

$$m_{\mathrm{BSK}} = \begin{bmatrix} 1-p & p \\ p & 1-p \end{bmatrix}$$

Ciągi błędów

Błąd – zmiana wartości logicznej elementu sygnału $\boldsymbol{s}_{\mathbf{i}}$ powstała w czasie przesyłania tego sygnału przez kanał ziarnisty.

Ciąg błędów:

$$\vec{z} = \vec{y} \oplus \vec{s} \quad \text{gdzie}: z_l = y_l \oplus s_j$$

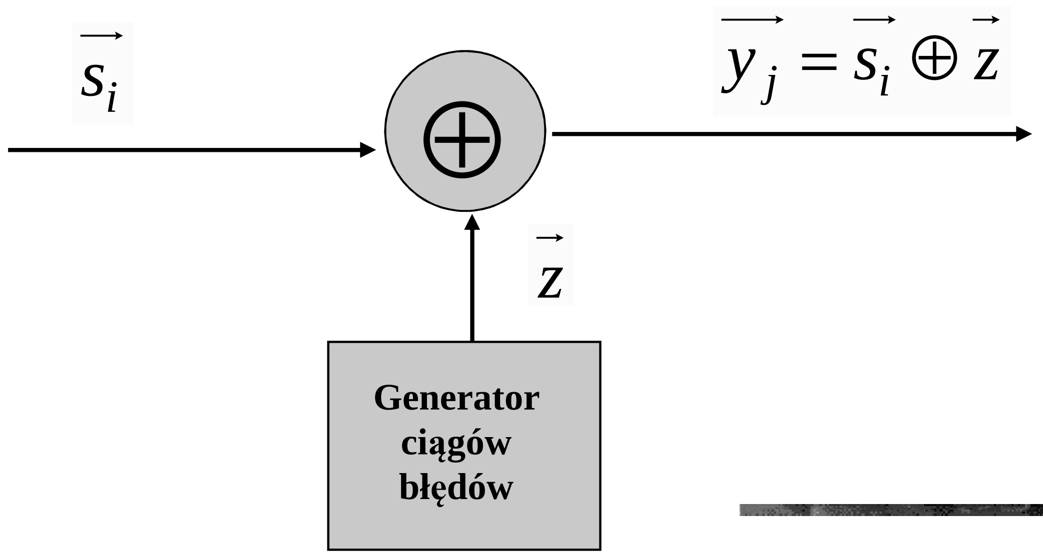
$$\text{dla} \quad l = 1,2,3,\ldots,n$$

$\vec{s_i}$	1	0	1	1	0	1	1	1	1	1	0	1
$\vec{y_j}$	1	0	1	0	0	1	0	0	1	1	0	1
$\vec{z_i}$	0	0	0	1	0	0	1	1	0	0	0	0

Ciągi błędów

Waga ciągu binarnego – liczba jedynek w ciągu. $\upsilon(\vec{s}) = \sum_{i=1}^{n} s_i$

Waga ciągu błędów – liczba błędów (jedynek) jakie wystąpiły podczas transmisji.



Ciągi błędów

Kodowanie nadmiarowe – przyporządkowanie sekwencji ***k***-pozycyjnych ciągów przeprowadzonych według reguł kodowania.

Kody nadmiarowe wzajemnie jednoznaczne – przyporządkowanie ciągów do ciągów kodowych jest wzajemnie jednoznaczne.

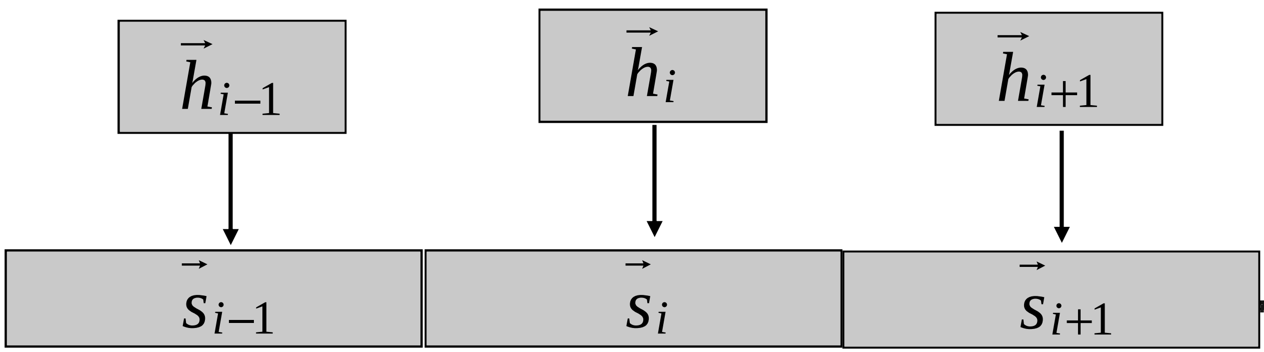
Dekodowanie – przyporządkowanie odebranym ciągom wyjściowym decyzji o nadanych ciągach kodowych w oparciu o reguły dekodowania oraz odtworzenie ciągu informacyjnego.

Kody blokowe

Kodowanie blokowe – to przekształcenie ***k***-pozycyjnych ciągów informacyjnych na ***n***-pozycyjne ciągi kodowe.

$$\vec{h} = (h_1, h_2, \ldots, h_k) \quad \rightarrow \quad \vec{s} = (s_1, s_2, \ldots, s_n)$$

Kodowanie ***i***-tego ciągu informacyjnego zamyka się w całości w czasie trwania ***i***-tego ciągu kodowego.



Kody blokowe

Właściwości:

Uzyskany kod blokowy to kod (n,k)

Sprawność kodu blokowego: $\eta = \frac{k}{n}$

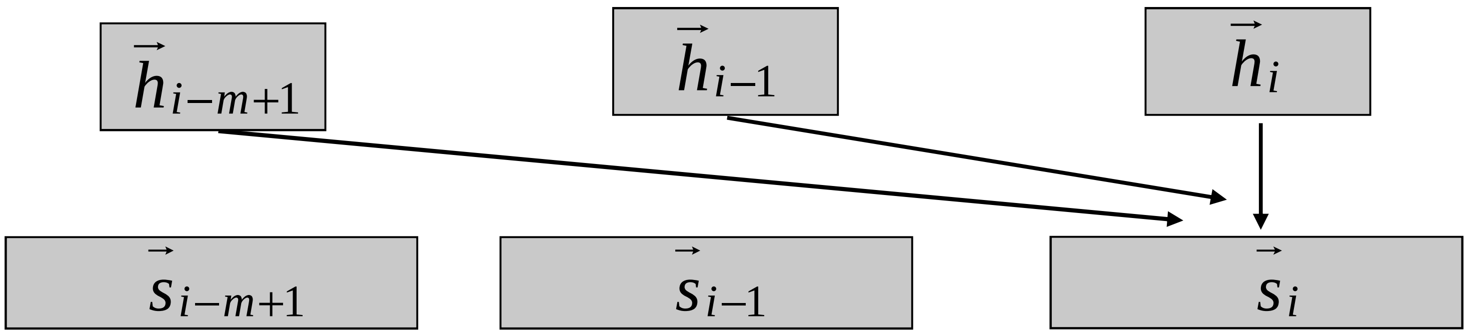
Nadmiar kodowy kodu blokowego: $p_k = 1 - \eta = 1 - \frac{k}{n}$

Efektywna szybkość transmisji: $\upsilon_{\mathrm{ef}} = \eta \cdot \upsilon_t; \quad [\mathrm{bit}/s]$

Kody splotowe

Kodowanie splotowe – ***i***-ty ciąg kodowy uzależniony jest od ***i***-tego ciągu informacyjnego oraz ***m***-1 poprzednich ciągów informacyjnych.

$$\overrightarrow{s_i} = f\left(h_{i-m+1}, h_{i-m+2}, \ldots, h_i\right)$$



Kody splotowe

Właściwości:

Uzyskany kod wymuszony $N = n \cdot m$

Sprawność kodu splotowego: $\eta = \dfrac{k}{n}$

Kod wymuszony – sekwencje złożone z ***m***-kolejnych ciągów kodowych:

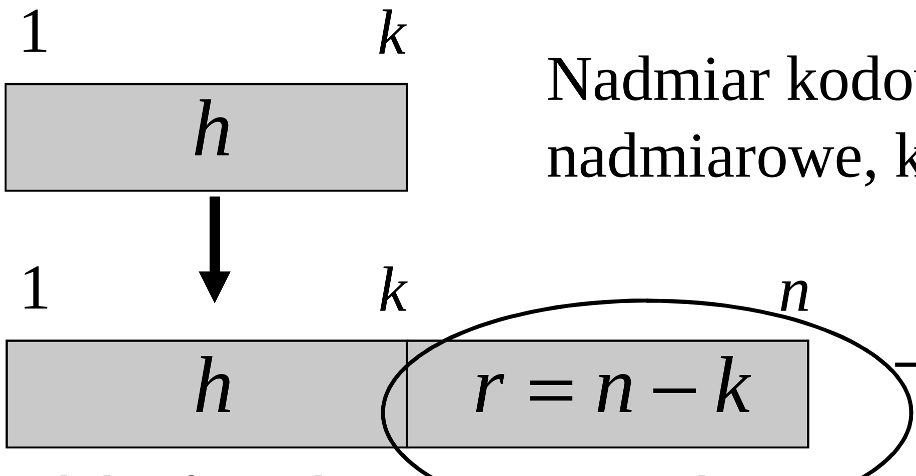
Lepiej dostosowany do kanału, w którym występuje szereg błędów (na podstawie całości możliwość odtworzenia przekłamanego lub zanikłego fragmentu).

Kody liniowe (blokowe)

Założenie: ciągi informacyjne ***h*** oraz kodowe ***s*** są binarne.

Kod rozdzielczy: $s_i = h_j; \quad \text{dla} \quad j = 1,2,3,\ldots k$

k – długość ciągu informacyjnego.



Nadmiar kodowy: $r = n - k$ pozycje nadmiarowe, kontrolne

Kody liniowe

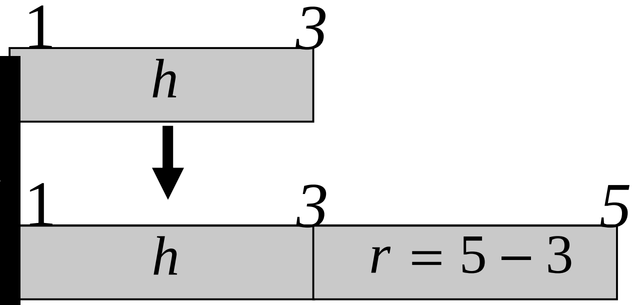
Kod liniowy: jest zadawany układem ***r*** liniowo niezależnych równań – „testów parzystości kodu” o znanych współczynnikach $\boldsymbol{T_{jl}}$.

$\boldsymbol{T_{jl}}$ przyjmują wartości ze zbioru {0,1}

$$\oplus \sum_{i=1}^{n} T_{jl} s_i = 0; \quad j = 1,2,3,\ldots,r$$

$$T_{jl} = \begin{cases} 1 & \text{- jeżeli } \boldsymbol{l}\text{-ta pozycja ciągu kodowego wchodzi w skład } \boldsymbol{j}\text{-tego równania.} \\ 0 & \text{- w przypadku przeciwnym.} \end{cases}$$

Kody liniowy (5,3) - przykład



1. Dla kodu rozdzielczego w każdym zespole kontrolnym występuje tylko jedna pozycja kontrolna.
2. Podzbiór pozycji ciągu kodowego objęty *j*-tym równaniem kontrolnym, to *j*-ty zespół kontrolny.

$$\begin{cases} s_1 \oplus s_2 \oplus s_4 = 0 \\ s_1 \oplus s_3 \oplus s_5 = 0 \end{cases}$$

dla utworzonego zespołu kontrolnego suma „mod 2”=0

$$\begin{cases} 1 \oplus 1 \oplus 0 = 0 \\ 1 \oplus 0 \oplus 1 = 0 \end{cases}$$

			$\vec{s}$				
0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	1
0	1	0	0	1	0	1	0
0	1	1	0	1	1	1	1
1	0	0	1	0	0	1	1
1	0	1	1	0	1	1	0
1	1	0	1	1	0	0	1
1	1	1	1	1	1	0	0

$d_{\min} = 2$

Odległość minimalna kodu $d_{\min}$

Odległość Hamminga: pomiędzy dwoma ***n***-pozycyjnymi ciągami, jest to liczba pozycji na których owe ciągi różnią się między sobą, lub waga sumy modulo tych ciągów.

Odległość minimalna kodu liniowego: to minimalna odległość pomiędzy dwoma ciągami kodowymi.

$$d_{\min} = \min\left[d\left(\vec{s}_i, \vec{s}_j\right)\right]$$

$i, j \in \{1,2,3,\ldots,2^k\}$ $\quad i \neq j \quad$ $\vec{s}_i, \vec{s}_j$ - ciągi kodowe

Dekodowanie detekcyjne

Dekodowanie detekcyjne: nie umożliwia odtworzenia sygnału (informacji), informuje o wystąpieniu błędów.

Nadmiar kodowy wykorzystywany jest do wykrywania błędów.

Reguła decyzyjna dekodera: „punktowa reguła decyzyjna z wymazywaniem”.

$$\vec{s} = \begin{cases} \vec{y} & \text{dla} \quad \vec{y} \in \{\vec{s}\} \\ ? & \text{w przypadku przeciwnym} \end{cases}$$

Decyzja wymijająca

Dekodowanie detekcyjne

Syndrom: ciąg o długości ***r*** uzyskany poprzez podstawienie $\boldsymbol{y_i}$ w miejsce $\boldsymbol{s_i}$ i wyliczenie testów parzystości.

$$Y_j(\vec{y}) = \oplus\sum_{i=1}^{n} T_{jl}\, y_i$$

Jeżeli odebrany ciąg jest ciągiem kodowym, to w wyniku syndrom składa się z samych zer.

$$\vec{Y}(\vec{y}) = \vec{0}_r$$

ciąg o długości ***r*** złożony z samych zer.

Reguła decyzyjna detektora:

$$\vec{s}^{\,*} = \begin{cases} \vec{y} & \text{dla} \quad Y\left(\vec{y}\right) = \vec{0}_r \\ ? & \text{w przypadku przeciwnym} \end{cases}$$

decyzja wymijająca

Dekodowanie detekcyjne - przykład

Sygnał odebrany ma postać: $y = 01010$

Wyliczenie testów parzystości: $Y_j(\vec{y}) = \oplus \sum_{i=1}^{n} T_{jl} y_i$

$$y_1 \oplus y_2 \oplus y_4 = 0 \oplus 1 \oplus 1 = 0$$
$$y_1 \oplus y_3 \oplus y_5 = 0 \oplus 0 \oplus 0 = 0$$

$$\Rightarrow \quad \vec{Y} = \{0,0\}$$

Przekłamanie informacji

Z powodu zakłóceń odebrano: $y = 01011$

$$y_1 \oplus y_2 \oplus y_4 = 0 \oplus 1 \oplus 1 = 0$$
$$y_1 \oplus y_3 \oplus y_5 = 0 \oplus 0 \oplus 1 = 1$$

$$\Rightarrow \quad \vec{Y} = \{0,1\}$$

Dekodowanie detekcyjne

Należy zwiększyć odległość minimalną poprzez zwiększenie nadmiaru. Powoduje to jednak zmniejszenie sprawności i spadek szybkości transmisji.

Kod liniowy wykrywa wszystkie ciągi błędów dla których zachodzi związek:

$$V(\vec{z}) < d_{\min}$$

Waga zakłóceń

Dekodowanie detekcyjne - przykład

Sygnał odebrany ma postać: $y = 01010$

Z powodu zakłóceń odebrano: $y = 01111$

$$y_1 \oplus y_2 \oplus y_4 = 0 \oplus 1 \oplus 1 = 0$$

$$y_1 \oplus y_3 \oplus y_5 = 0 \oplus 1 \oplus 1 = 0$$

$$\Rightarrow \quad \vec{Y} = \{0,0\}$$

Kod nie jest w stanie poprawnie wykryć błędu transmisji.

Dekodowanie korekcyjne

Dekodowanie korekcyjne: nie informuje o wystąpieniu błędów, lecz stara się je usunąć.

Nadmiar kodowy wykorzystywany jest do wykrywania błędów oraz lokalizacji wykrytych błędów i skorygowania pozycji błędów.

Reguła decyzyjna dekodera: „punktowa reguła decyzyjna brak odpowiedzi wymijającej.

$$\vec{s} = \vec{y} \quad \text{dla} \quad \vec{y} \in \{\vec{s}\}$$

Dekodowanie korekcyjne

Syndrom: ciąg o długości ***r*** uzyskany poprzez podstawienie $\boldsymbol{y_i}$ w miejsce $\boldsymbol{s_i}$ i wyliczenie testów parzystości.

jeśli $\vec{Y}(\vec{y}) = \vec{0}_r$ podejmowana jest decyzja $\vec{s}^* = \vec{y}$

jeśli $\vec{Y}(\vec{y}) \neq \vec{0}_r$ to na podstawie syndromu określenie jaki ciąg błędów wystąpił i podejmowana jest decyzja $\vec{s}^* = \vec{y} \oplus \vec{z}'$

gdzie: $\vec{z}'$ - ciąg błędów określony na podstawie syndromu

Dekoder optymalny działa zgodnie z zasadą maksymalnego prawdopodobieństwa.

Korekcja błędów

Niech $\vec{s}_i$ oraz $\vec{s}_j$ będą ciągami kodowymi o odległości: $d(s_i;s_j) = d_{\min}$

Jeśli w odebranym ciągu $\vec{s}_i$ wystąpił pojedynczy błąd, to $d(\vec{y};\vec{s}_i) = 1$

oraz $d(\vec{y};\vec{s}_j) = d_{\min} \pm 1$

Dla ***t***-błędów, w najgorszym przypadku $d(\vec{y};\vec{s}_j) = d_{\min} - t$

Dekoder skoryguje błąd ***t***-krotny, gdy $d(\vec{y};\vec{s}_j) > d(\vec{y};\vec{s}_i); \quad d(\vec{y};\vec{s}_i) = t$

W najgorszym przypadku, $d_{\min} - t > t$ więc: $d_{\min} > 2t$

Kod może korygować błędy o krotności mniejszej niż połowa odległości minimalnej kodu.

Kod z kontrolą parzystości (detekcyjny)

Kod liniowy, detekcyjny dany równaniem: $\oplus \sum_{l=1}^{n} s_l = 0$

Jest to kod typu $(n, n-1)$

Wyznaczanie pozycji kontrolnej $s_n = \oplus \sum_{l=1}^{n-1} s_l$

Oznacza to, że zbiór ciągów kodowych to zbiór wszystkich ciągów ***n***-pozycyjnych o parzystej liczbie jedynek.

Odległość minimalna tego kodu: $d_{\text{min}}=2$

Kod wykrywa wszystkie błędy o krotności nieparzystej, nie wykrywa żadnych błędów o krotności parzystej.

Korekcyjny kod Hamminga

Kod o $\boldsymbol{d}_{\mathbf{min}} = 3$ i może korygować błędy pojedyncze.

Syndrom – ciągu odebranego traktowany jest jako liczba binarna, która wskazuje na numer pozycji błędu.

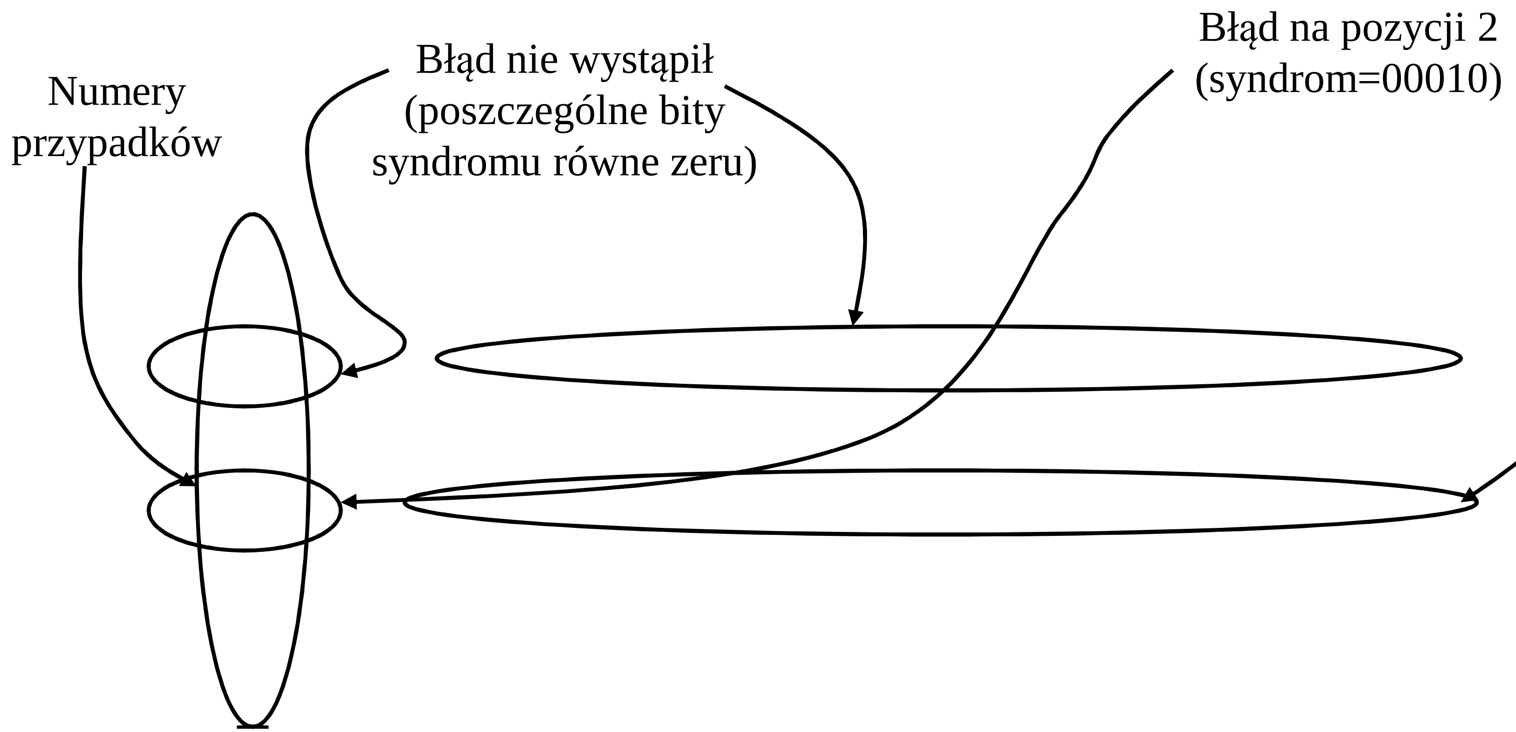
Należy rozpatrzyć n +1 sytuacji (pojedynczy błąd może wystąpić na „n” sposobów, bądź może nie wystąpić).

Należy stworzyć n +1 różnych syndromów.

Dla kodu liniowego można stworzyć 2^r różnych syndromów więc:

$$2^r \geq n+1$$

Korekcyjny kod Hamminga



Poszczególne wiersze zawierają kolejne zespoły kontrolne.
Kod Hamminga jest kodem nierozdzielnym.
Pozycje kontrolne występują w jednym zespole kontrolnym.

Kod Hamminga (7,4)

$r = 3$

$k = 4$

Kod (7,4) posiada trzy bity nadmiarowe: $n = 7$

2^r syndromów = 2^3 = 8

deklarowanie wartości syndromu

	Y_3	Y_2	Y_1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

I zespół kontrolny **K1** - $\boxed{s_1} \oplus s_3 \oplus s_5 \oplus s_7 = 0$

II zespół kontrolny **K2** - $\boxed{s_2} \oplus s_3 \oplus s_6 \oplus s_7 = 0$

III zespół kontrolny **K3** - $\boxed{s_4} \oplus s_5 \oplus s_6 \oplus s_7 = 0$

K1, K2, K3 – stanowią pozycje kontrolne bowiem występują tylko raz w jednym zespole kontrolnym !!!.

Kod Hamminga (7,4)

proces kodowania

Kod (7,4) posiada trzy bity nadmiarowe:

$r = 3$

$k = 4$

$n = 7$

I zespół kontrolny **K1 -** $\boxed{s_1} \oplus s_3 \oplus s_5 \oplus s_7 = 0$

II zespół kontrolny **K2 -** $\boxed{s_2} \oplus s_3 \oplus s_6 \oplus s_7 = 0$

III zespół kontrolny **K3 -** $\boxed{s_4} \oplus s_5 \oplus s_6 \oplus s_7 = 0$

4 pozycje informacyjne

	Y_3	Y_2	Y_1
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

h_i

0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0

1	2	3	4	5	6	7
$\boxed{0}$	$\boxed{0}$	0	$\boxed{0}$	0	0	0
$\boxed{0}$	$\boxed{0}$	0	$\boxed{1}$	0	0	1

jest to ciąg $\vec{h_i}$

$\boxed{1}$	$\boxed{1}$	0	$\boxed{0}$	1	1	0

Kod Hamminga (7,4)

Obliczanie syndromu

Obliczanie syndromu dla ciągu kodowego:

$$\boxed{1}\ \boxed{1}\ \boxed{0}\ \boxed{0}\ 1\ 1\ 0$$

I zespół kontrolny **K1** - $\boxed{s_1} \oplus s_3 \oplus s_5 \oplus s_7 = 0$

II zespół kontrolny **K2** - $\boxed{s_2} \oplus s_3 \oplus s_6 \oplus s_7 = 0$

III zespół kontrolny **K3** - $\boxed{s_4} \oplus s_5 \oplus s_6 \oplus s_7 = 0$

$$Y_1 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$Y_2 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$Y_3 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$\vec{Y} = \{000\}_b = 0$$

błąd na pozycji „3”

$$\boxed{1}\ \boxed{1}\ \boxed{1}\ \boxed{0}\ 1\ 1\ 0$$

$$Y_1 = 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$Y_2 = 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$Y_3 = 0 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$\vec{Y} = \{011\}_b = 3$$

Kod Hamminga (7,4)
Obliczanie syndromu

Obliczanie syndromu dla ciągu kodowego:

$\boxed{1}\ \boxed{1}\ \boxed{0}\ \boxed{0}\ 1\ \boxed{1}\ 0$

błąd na pozycji „3" i „6"

$1\ 1\ \boxed{1}\ 0\ 1\ \boxed{0}\ 0$

$$Y_1 = 1 \oplus 1 \oplus 1 \oplus 0 = 1$$

$$Y_2 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$$

$$Y_3 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

BRAK KOREKCJI

$$\vec{Y} = \{101\}_b = 5$$

$\boxed{1}\ \boxed{1}\ 0\ \boxed{0}\ 1\ \boxed{1}\ 0$

błąd na pozycji „6"

$1\ 1\ 0\ 0\ 1\ \boxed{0}\ 0$

$$Y_1 = 1 \oplus 0 \oplus 1 \oplus 0 = 0$$

$$Y_2 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$$

$$Y_3 = 0 \oplus 1 \oplus 0 \oplus 0 = 1$$

$$\vec{Y} = \{110\}_b = 6$$

Wydłużony kod Hamminga

Do kodu Hamminga (n, k) i $d_{min} = 3$ dodana zostaje jedna pozycja kontrolna, uzyskany zostaje: $(n + 1, k)$ i $d_{min} = 4$.

Kod ten koryguje wszystkie błędy pojedyncze oraz wykrywa błędy podwójne (kod korekcyjno-detekcyjny).

Dodatkowa pozycja kontrolna to test parzystości:

Dekoder podejmuje decyzję na podstawie dwóch decyzji:

$\vec{s}_1$ - wytworzona na podstawie syndromu, jak w kodzie Hamminga.

$\vec{s}_2$ - wytworzona na podstawie testu parzystości.

Wydłużony kod Hamminga

$$\vec{s}_2 = \oplus \sum_{l=1}^{n+1} y_l$$ - wytworzona na podstawie testu parzystości.

Możliwe przypadki:

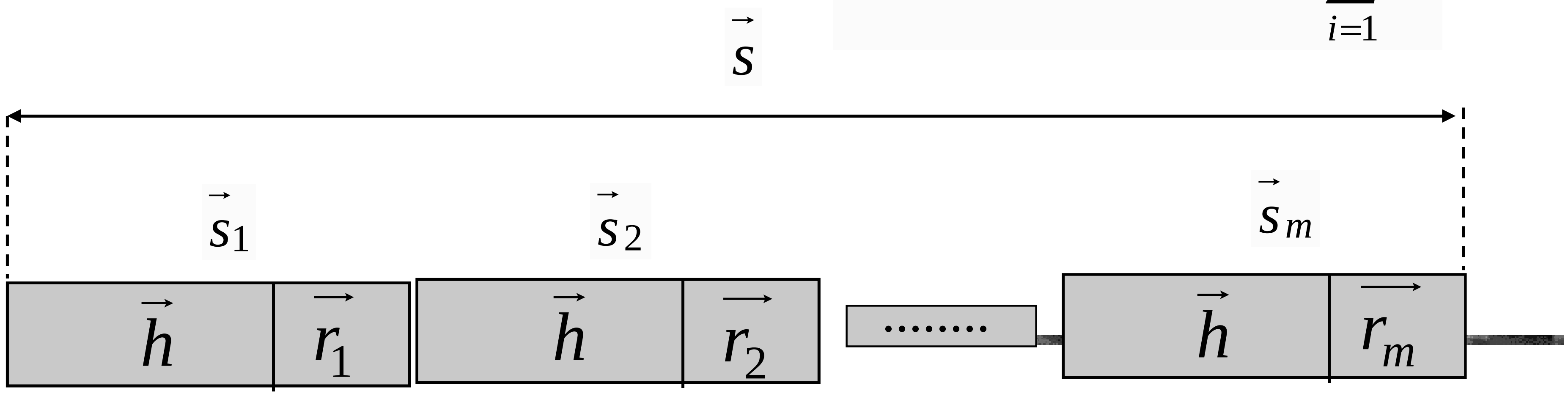
1. Brak błędu $\overrightarrow{s_1} = \vec{0},\ \overrightarrow{s_2} = 0$; podjęcie decyzji;
2. Błąd pojedynczy $\overrightarrow{s_1} \neq \vec{0},\ \overrightarrow{s_2} = 1$;korekcja na podstawie syndromu;
3. Błąd na pozycji kontrolnej $\overrightarrow{s_1} = \vec{0},\ \overrightarrow{s_2} = 1$; podjęcie decyzji;
4. Błąd podwójny $\overrightarrow{s_1} \neq \vec{0},\ \overrightarrow{s_2} = 0$; decyzja wymijająca.

Kody łączne

Dany jest zbiór kodów blokowych $\{C_i\}$ o jednakowej długości ciągu informacyjnego k.

Kod *m*-razy łączony – kod dla którego ciąg kodowy $\vec{s}$ powstaje z szeregu połączeń m-ciągów.

$$d_{\min} = \sum_{i=1}^{m} d_{\min i}; \quad \eta = \frac{k}{\sum_{i=1}^{m} n_i}$$



Kod iterowany

Kod dwukrotnie iterowany:

Długość ciągu informacyjnego: $k = k_1 k_2$, który wpisuje się w macierz o wymiarach „k x k”;

Wiersze koduje się kodem (n_1, k_1)

Kolumny koduje się kodem (n_2, k_2)

Otrzymany kod to kod: (n_1, n_2, k_1, k_2)

$$d_{\min} = d_{\min 1} \cdot d_{\min 2}; \quad \eta = \frac{k_1 \cdot k_2}{n_1 \cdot n_2} = \eta_1 \cdot \eta_2$$

Sprawność wyższa niż dla kodu łącznego

Zapis wielomianowy kodu

Binarny *n*-pozycyjny ciąg traktowany jest jako wielomian postaci:

$$\vec{b} = \vec{b}_{n-1} \cdot x^{n-1} \oplus \vec{b}_{n-2} \cdot x^{n-2} \oplus,\ldots,\oplus b_1 \cdot x \oplus b_0$$

$$b_j \in \{0,1\} \;\; : j = 0,1,\ldots,n-1$$

Przykład: $\vec{a} = 01011$ ⇒ $a(x) = x^4 \oplus x^2 \oplus x$

Kodowanie rozdzielne

Pierwsze *k*-pozycji to pozycje informacyjne.

Dzieląc przesunięty wielomian informacyjny przez wielomian generacyjny otrzymano :

$$\frac{x^r h(x)}{g(x)} = A(x) \oplus \frac{r(x)}{g(x)}$$

Mnożąc obustronnie otrzymano:

$$A(x) \cdot g(x) = x^r h(x) \oplus r(x)$$

Wielomian $A(x)g(x)$ należy do zbioru {s(x)}

$$s(x) = A(x) \cdot g(x)$$

Reguła kodowania ma postać:

$$s(x) = x^r \cdot h(x) \oplus r(x)$$

Kodowanie rozdzielne

$$\frac{x^r h(x)}{g(x)} = A(x) \oplus \frac{r(x)}{g(x)}$$

Przykład: Kod (7,3) generowany przez wielomian $g(x) = x^4 \oplus x^2 \oplus x \oplus 1$

$$g(x) = \{10111\}$$

h			s							$V(s)$
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	1	1	1	4
0	1	0	0	1	0	1	1	1	0	4
0	1	1	0	1	1	1	0	0	1	4
1	0	0	1	0	1	1	1	0	0	4
1	0	1	1	0	0	1	0	1	1	4
1	1	0	1	1	1	0	0	1	0	4
1	1	1	1	1	0	0	1	0	1	4

$$d_{\min} = 4$$

$$h(x) = \{011\} \xrightarrow{\cdot x^4} \{0110000\}$$

$$\{0110000\} \xrightarrow{\div g(x)} \{0111001\}$$

Kodowanie rozdzielne

$$\frac{x^r h(x)}{g(x)} = A(x) \oplus \frac{r(x)}{g(x)}$$

Przykład: Kod (7,3) generowany przez wielomian $g(x) = x^4 \oplus x^2 \oplus x \oplus 1$

$$g(x) = \{10111\}$$

$$\{0110000\} \xrightarrow{\div g(x)} \{0111001\}$$

$$A(x): \boxed{1\ 1}$$

$$\begin{array}{cccccccc}
 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
\oplus & 1 & 0 & 1 & 1 & 1 & & \\
\hline
 & & 0 & 1 & 1 & 1 & 1 & 0 \\
\oplus & & & 1 & 0 & 1 & 1 & 1 \\
 & & 0 & 0, & 1 & 0 & 0 & 1
\end{array}
\quad \div \boxed{1\ 0\ 1\ 1\ 1}$$

$g(x)$

$x^r \cdot h(x)$

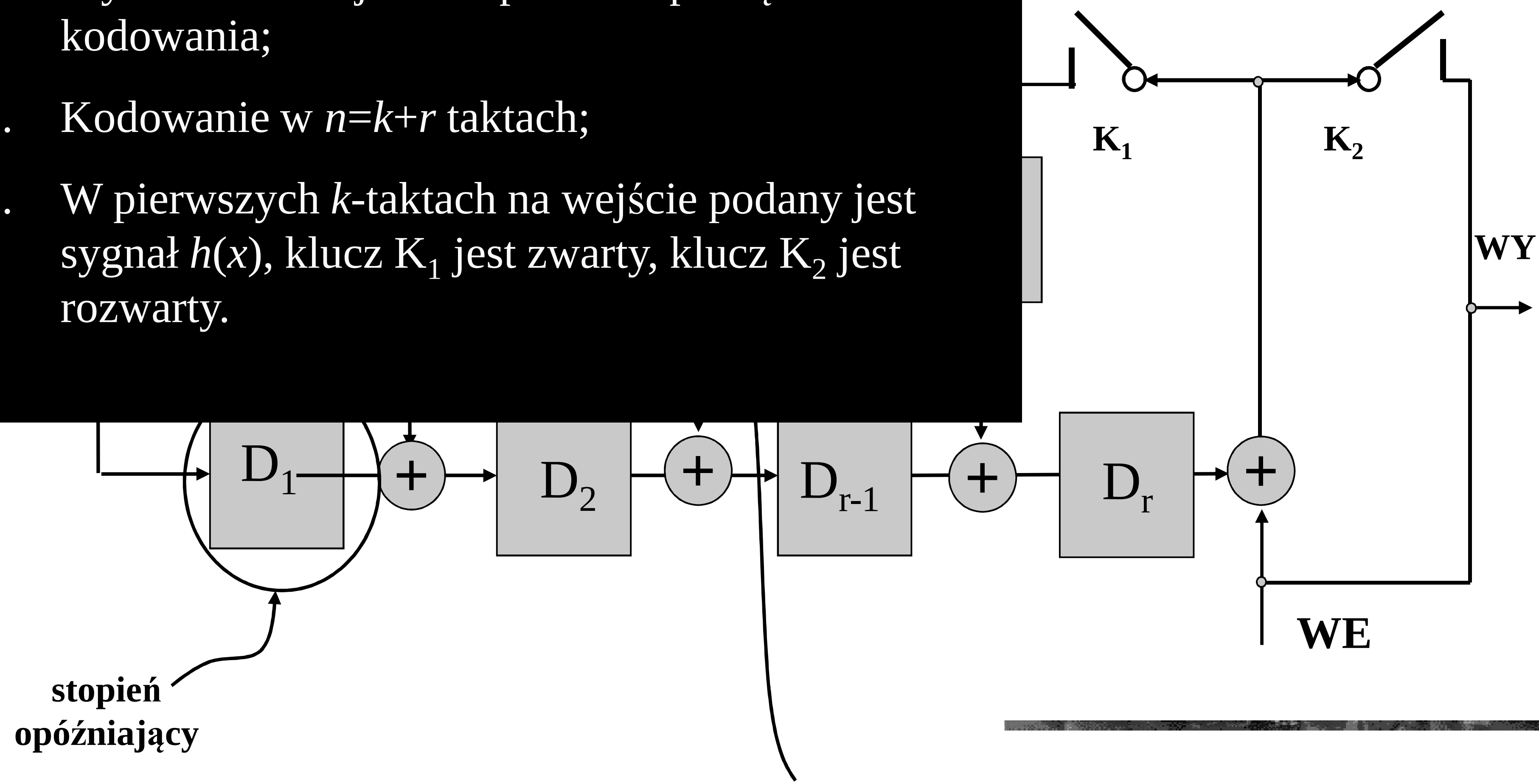
$r(x)$

$$s(x) = A(x) \cdot g(x) = x^r h(x) \oplus r(x) = \{0111001\}$$

Koder kodu rozdzielnego

realizuje dzielenie wielomianów

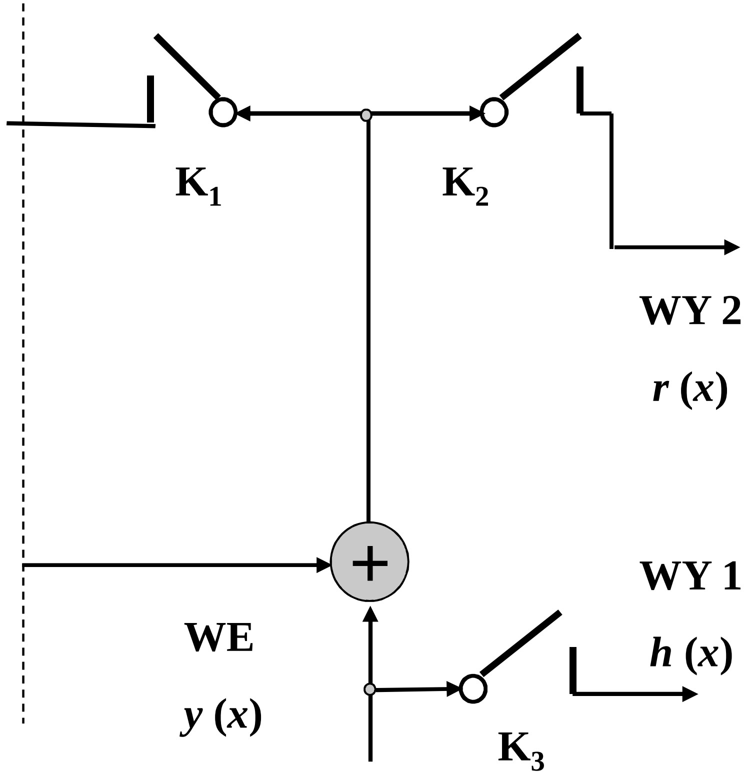
1. Wyzerowanie rejestrów przed rozpoczęciem kodowania;
2. Kodowanie w $n=k+r$ taktach;
3. W pierwszych k-taktach na wejście podany jest sygnał $h(x)$, klucz K_1 jest zwarty, klucz K_2 jest rozwarty.



Dekoder kodu rozdzielnego

realizuje dzielenie wielomianów

1. Ciąg informacyjny jest znany zanim zostanie podjęta decyzja o jego poprawności;
2. Na wejściu odebrany $y(x)$;
3. Wyzerowanie rejestrów przed rozpoczęciem dekodowania w $n=k+r$ taktach;
4. W pierwszych k-taktach odbywa się kluczowanie.



Kodowanie nierozdzielne

Ciągi informacyjne są mnożone przez wielomian generujący $g(x)$.

$$s(x) = w(x) \cdot g(x)$$

$w(x)$ - wielomian stopnia większego niż k-1.

Przykład: Kod (7,3) generowany przez wielomian $g(x) = x^4 \oplus x^2 \oplus x \oplus 1$

$$g(x) = \{10111\}$$

h			s							$V(s)$
0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	1	0	1	1	1	4
0	1	0	0	1	0	1	1	1	0	4
0	1	1	0	1	1	1	0	0	1	4
1	0	0	1	0	1	1	1	0	0	4
1	0	1	1	0	0	1	0	1	1	4
1	1	0	1	1	1	0	0	1	0	4
1	1	1	1	1	0	0	1	0	1	4

$$\begin{array}{ccccccccc}
 & & & 1 & 0 & 1 & 1 & 1 \\
 & & & & * & 0 & 1 & 1 \\
\hline
 & & & 1 & 0 & 1 & 1 & 1 \\
\oplus & & 1 & 0 & 1 & 1 & 1 & \\
 & 0 & 0 & 0 & 0 & 0 & & \\
\hline
 & 0 & 1 & 1 & 1 & 0 & 0 & 1
\end{array}$$

Koder kodu nierozdzielnego

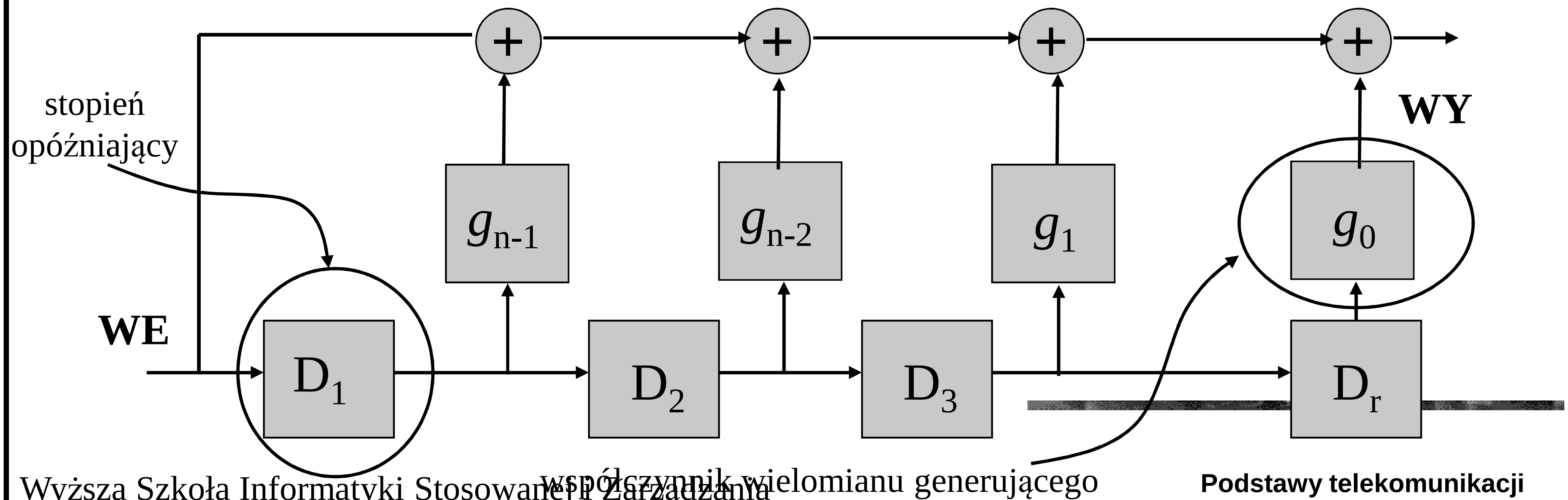
realizuje mnożenie wielomianów

Dla cyklicznego kodowania nierozdzielnego zachodzi:

Mnożenie wielomianów jest równoważne sumowaniu przesuniętych iloczynów cząstkowych.

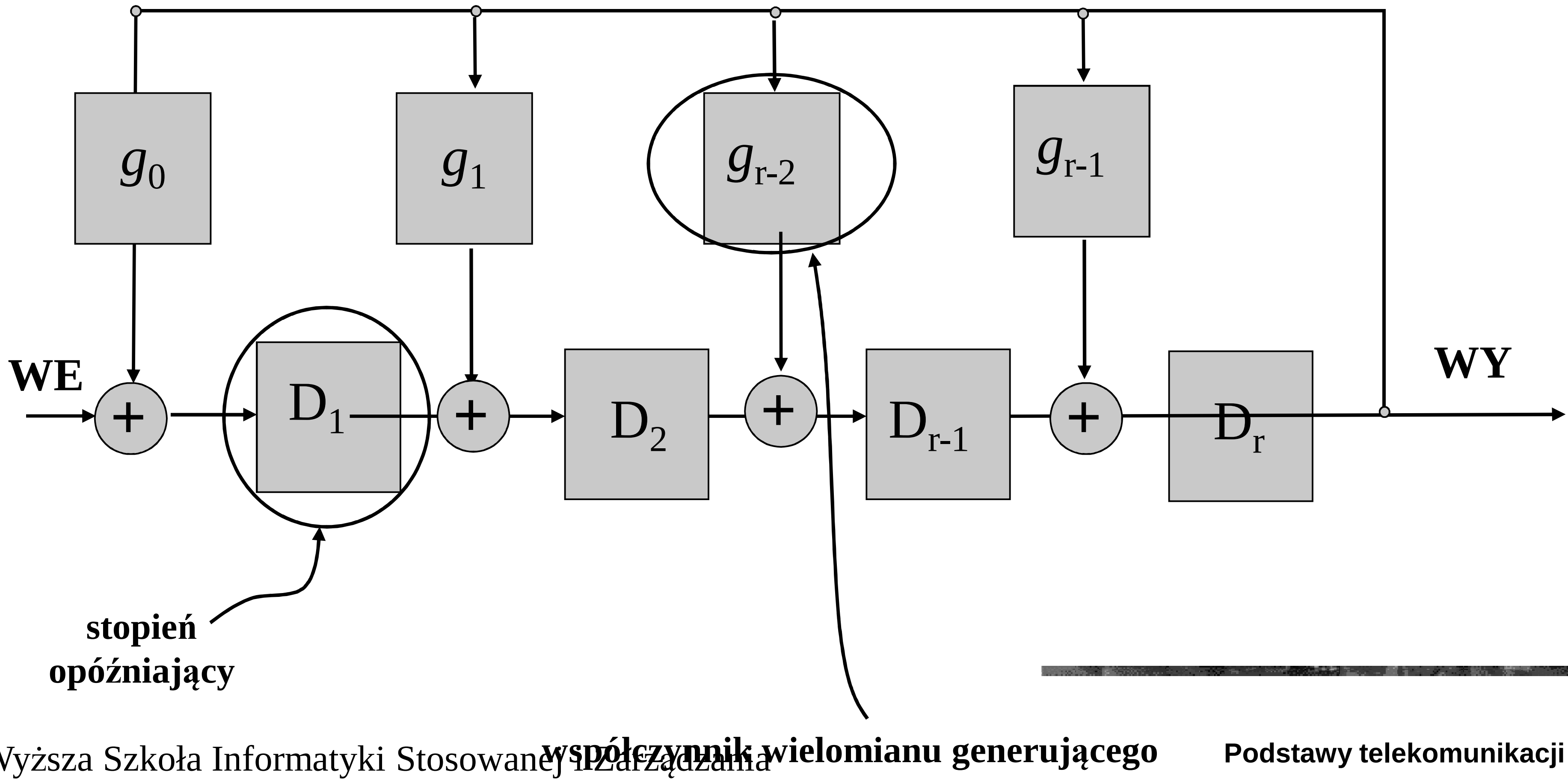
$$s(x) = h(x) \oplus \sum_{i=0}^{r} g_i \cdot x^i$$

Przykład: $g = 1011;\ h = 1001;\ g * h = 1010011$



Dekoder kodu nierozdzielnego

realizuje dzielenie wielomianów



DZIĘKUJĘ
ZA UWAGĘ