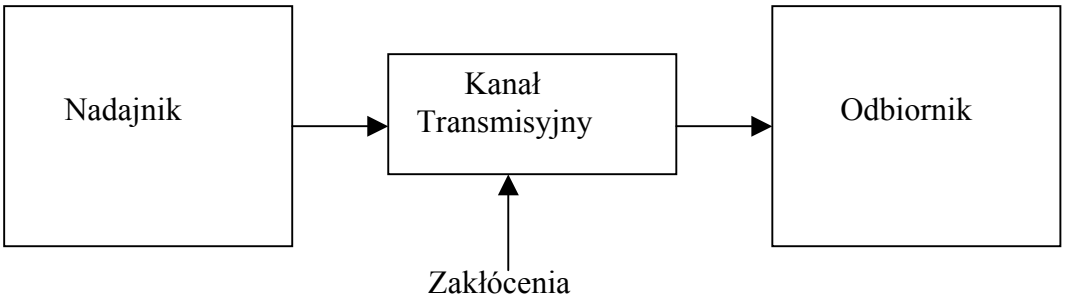
2.1 Kody wykrywające i korygujące błędy

1. Dlaczego stosujemy kody wykrywające i korygujące błędy

Całkowicie pewne, bezbłędne przesyłanie słów nad ustalonym alfabetem (z reguły słów binarnych) przez tzw. kanał komunikacyjny nie jest możliwe z uwagi na obecność w środowisku fizycznym szumów i różnego typu zakłóceń.. Podobnie przechowywanie, magazynowanie słów w pamięci nie jest pozbawione błędów. Wie o tym dobrze każdy doświadczony (przez los) użytkownik pamięci (własnej). Kody wykrywające i korygujące błędy stosowane są po to by zminimalizować błędy w interpretacji słowa kodowego przenoszącego informację pomimo ewentualnych błędów jakie mogą się pojawić na niektórych pozycjach wewnątrz odebranego słowa kodowego. Kody potrafiące wykrywać tylko błędy nazywamy kodami wykrywającymi błędy a kody potrafiące wykrywać i korygować błędy nazywamy kodami korekcyjnymi ale często dla uproszczenia obie grupy kodów nazywamy kodami korekcyjnymi.

Uwaga: Kody korekcyjne stosowane są głównie w:

- telekomunikacji i sieciach komputerowych
- pamięciach półprzewodnikowych RAM
- systemach zapisu danych na płytach CD



Rys. 1. Przesyłanie słów przez kanał komunikacyjny z zakłóceniami. Ze względu na szumy i zakłócenia musimy liczyć się z wystąpieniem błędów (przekłamań) w odebranym w odbiorniku słowie.

2. Wykrywanie błędów metodą kontroli parzystości

Wykrywanie błędów metodą kontroli parzystości jest najprostszą często stosowaną metodą wykrywania błędów.

Załóżmy, że w systemie cyfrowym chcemy przesyłać dane porcjami za pomocą słów n bitowych. Niech $x_1x_2...x_n \in \{0,1\}^n$ będzie takim n bitowym słowem. Przed wysłaniem uzupełniamy to słowo jeszcze jednym bitem x_{n+1} tzw. bitem parzystości przy czym

$x_{n+1} = 1$ jeśli liczba jedynek w słowie $x_1x_2...x_n \in \{0,1\}^n$ jest nieparzysta

$x_{n+1} = 0$ jeśli liczba jedynek w słowie $x_1x_2...x_n \in \{0,1\}^n$ jest parzysta

Zatem słowo przesyłane $x_1x_2...x_nx_{n+1}$ zawiera zawsze parzystą liczb jedynek. Łatwo sprawdzić, że układ generujący bit parzystości jest *n* wejściową sumą modulo 2.

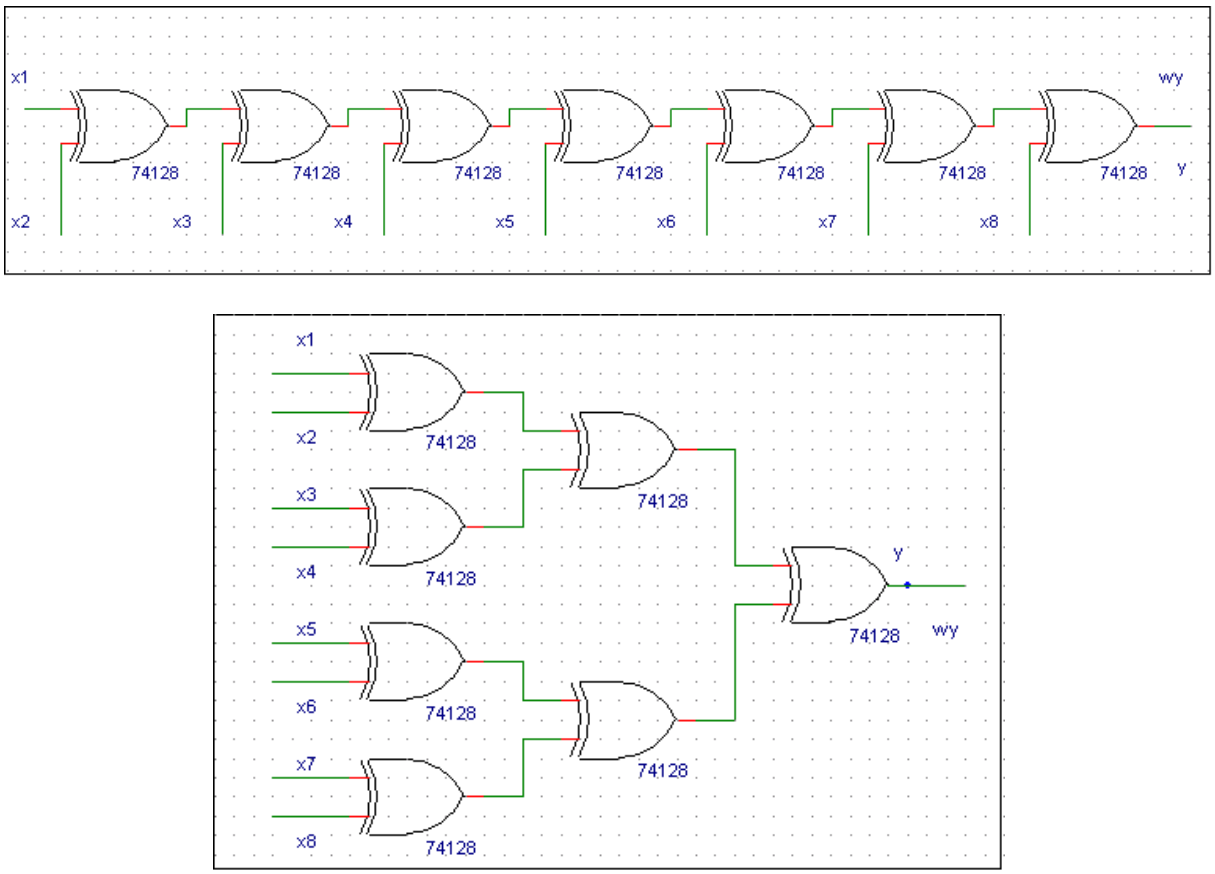
$$x_{n+1} = x_1 \oplus x_2 \oplus ... \oplus x_n$$

Nadajnik wysyła słowo *n*+1 bitowe $x_1x_2...x_nx_{n+1}$ a w odbiorniku dostajemy słowo być może z błędami (przekłamaniami) $y_1y_2...y_ny_{n+1} \in \{0,1\}^{n+1}$. Sprawdzamy parzystość liczby jedynek w słowie binarnym odebranym $y_1y_2...y_ny_{n+1}$. Układ sprawdzający czy liczba jedynek jest parzysta jest *n*+1 wejściową sumą modulo 2 (por. Rys.2). Jeśli liczba jedynek w słowie $y_1y_2...y_ny_{n+1}$ jest parzysta to na wyjściu układu mamy 0 jeśli nieparzysta to 1.

Jeśli wysłaliśmy parzystą liczbę jedynek a otrzymaliśmy w odbiorniku nieparzystą tzn., że podczas przesyłania wystąpił błąd. Sprawdzając parzystość wykryjemy każdą nieparzystą ilość błędów.

Zamiast funkcji $x_{n+1} = x_1 \oplus x_2 \oplus ... \oplus x_n$ można wykorzystać funkcję $x_{n+1} = \overline{x_1 \oplus x_2 \oplus ... \oplus x_n}$, wtedy przesyłane słowo będzie zawierało nieparzystą liczbę jedynek.

Pewną modyfikacją opisanej wyżej koncepcji są kody ze stałą liczbą jedynek *m* w słowie kodowym o stałej długości *n* tzw. „kody m z n" (por. podrozdział „kody numeryczne"). Pojedyncza zmiana 0 na 1 lub odwrotnie zmienia w takich kodach „bilans" jedynek i zer" i możemy wykryć błąd.



Rys. 2. Układy sprawdzające parzystość słowa binarnego $x_1x_2x_3x_4x_5x_6x_7x_8$

3. Podstawowe twierdzenia o kodach wykrywających i korygujących błędy

Definicja (m,n) *kodu* i *kodu liniowego*. Zakładamy, że chcemy przesyłać słowa nad alfabetem $GF(p^k)$ (alfabetem jest ciało skończone mające p^k elementów, gdzie p jest liczbą pierwszą w szczególności może to być ciało $GF(2) = Z_2 = \{0,1\}$). Słowa przesyłane mają długość m, chcemy więc przesyłać słowa z $(GF(p^k))^m$. Koncepcja kodu wykrywającego lub korygującego błędy polega na wykorzystaniu kodu redundancyjnego $f : (GF(p^k))^m \to (GF(p^k))^n$, gdzie $n > m$. Taki kod nazywamy **(m,n) *kodem*.** Obiektami kodowanymi są tu słowa o długości m nad alfabetem $GF(p^k)$ a słowami kodowymi słowa z $(GF(p^k))^m$. Zbiory $(GF(p^k))^m$ i $(GF(p^k))^n$ są przestrzeniami liniowymi nad ciałem $GF(p^k)$. Jeśli odwzorowanie $f : (GF(p^k))^m \to (GF(p^k))^n$ jest liniowe to kod nazywamy ***kodem liniowym***.

Liniowość odwzorowania oznacza, że istnieje taka macierz A o współczynnikach w ciele $GF(p^k)$ mająca n wierszy i m kolumn, że dla wektora $a \in (GF(p^k))^m$ mamy

$$f(a) = A \cdot a$$

gdzie wektor a jest traktowany jako macierz kolumnowa.

Kod wykrywający błędy powinien zachowywać się tak, że gdy odbieramy zamiast słowa kodowego $f(a)$ słowo z przekłamaniami na pewnej niewielkiej liczbie co najwyżej r pozycji (oznaczmy to słowo z przekłamaniami przez $b_r(f(a))$) to oglądając $b_r(f(a))$ będziemy mogli stwierdzić czy nastąpiły przekłamania.

Powiemy, że kod $f : (GF(p^p))^m \to (GF(p^k))^n$ wykrywa fakt popełnienia co najwyżej r błędów jeśli dla każdego $a \in (GF(p^k))^m$ zmiana cyfr na co najwyżej r pozycjach w słowie kodowym $f(a)$ nie powoduje przejścia słowa kodowego w słowo $f(b)$ dla pewnego $b \in (GF(p^k))^m, a \neq b$.

Praktycznie więc wykrywanie błędu może polegać na porównaniu $b_r(f(a))$ z wszystkimi możliwymi słowami kodującymi $f(c)$ dla $c \in (GF(p^p))^m$. Jeśli nie stwierdzamy zgodności, to stwierdzamy, że słowo $b_r(f(a))$ zawiera przekłamanie.

Kod korygujący błędy powinien zachowywać się tak, że gdy odbieramy zamiast słowa kodowego $f(a)$ słowo z przekłamaniami na pewnej niewielkiej liczbie co najwyżej r pozycji (tzn. słowo $b_r(f(a))$) to oglądając $b_r(f(a))$ będziemy mogli skorygować błędy (stosując pewien algorytm) uzyskując słowo $f(a)$ a więc w konsekwencji stwierdzamy, że zostało nadane słowo a.

Zasadnicza koncepcja na jakiej opierają się kody korekcyjne jest taka: Niech $f : (GF(p^k))^m \to (GF(p^k)^n$ będzie (m,n) kodem kodującym słowa o długości m za pomocą słów o długości n, gdzie $n > m$. Zakładamy, że podczas transmisji słowa $f(a)$ o długości m może powstać co najwyżej r błędów. Dla każdego $a \in (GF(p^k))^m$ kodujemy przy tym słowo a takim ciągiem $f(a) \in (GF(p^k))^n$, że

$$\bigvee_{a,b\in(GF(p^k))^m,\,a\neq b} K(f(a),r)\cap K(f(b),r)=\varnothing \qquad (1)$$

gdzie $K(x,r)$ oznacza kulę o promieniu r w przestrzeni metrycznej $GF(p^k))^n$ z metryką Hamminga d_H. Warunek (1) oznacza, że każde dwie kule rodziny kul $\{K(f(a),r); a\in(GF(p^k))^m\}$ są rozłączne. Jeśli tak jest to aby stwierdzić jaka informacja została nadana wystarczy zastosować do słowa odebranego $b_k(f(a))\in(GF(p^n))^m$ regułę decyzyjną $d:(GF(p^k)^n\to(GF(p^k))^m$ zdefiniowaną wzorem

$$d(b_r(f(a)))=a$$

wtedy i tylko wtedy, gdy

$$d_H(f(a),b_r(f(a)))=\min_{x\in(GF(p^k))^m}\{d_H(f(x),b_r(f(a))\} \qquad (2)$$

Innymi słowy jako informację nadaną przyjmujemy takie $a\in(GF(p^k))^m$, że odebrane słowo $b_r(f(a))\in(GF(p^k))^n$ jest najbliższe $f(a)$ tzn.

$$\bigvee_{x\in(GF(p^k))^m,x\neq a} d_H(f(a),b)<d_H(f(x),b) \qquad (3)$$

Jest oczywiste, że powyższa reguła decyzyjna przy warunku (1) jest poprawna w tym sensie, że pozwala na odtworzenie wiadomości nadanej przy ograniczonej do r ilości błędów transmisji.

Warto zauważyć, że istotą powyższego pomysłu na kod korekcyjny jest to, że wokół każdego słowa kodującego $f(a)$ (gdzie $a\in(GF(p^k))^m$) tworzymy otoczenie (w metryce Hamminga), którego elementy (oprócz $f(a)$) nie są wykorzystywane do kodowania słów z $a\in(GF(p^k))^m$. Możemy jednak odebrać (w wyniku wprowadzenia błędów podczas przesyłania słowa $f(a)$) dowolny element z tego otoczenia a mimo to nie tracimy orientacji w sytuacji, potrafimy wykryć i skorygować przekłamania.

Twierdzenie: Niech będzie dany (*m*,*n*) kod $f:(GF(p^k))^m\to(GF(p^k))^n$ i ustalona liczba $r\in<0,n>$. Kod f wykrywa fakt popełnienia co najwyżej r błędów wtedy i tylko wtedy gdy dla każdego $a,b\in(GF(p^k)^m, a\neq b$ mamy $f(a)\notin K(f(b),r)$ (czyli $d_H(f(a),f(b))>r$).

Twierdzenie: Kod $f:(GF(p^k))^m\to(GF(p^k))^n$ koryguje popełnienia co najwyżej r błędów wtedy i tylko wtedy, gdy dla każdego $a,b\in(GF(p^k))^m$, $a\neq b$ mamy $d_H(f(a),f(b))\geq 2r+1$ (lub równoważnie spełniony jest warunek (1)).

Przykład: Klasycznym przykładem kodu korekcyjnego jest kod *s* krotnie powtórzony. Powstaje on przez przyporządkowanie słowu kodowanemu $a\in(GF(p^k))^m$ słowa kodowego $\underbrace{aa...a}_{s}\in(GF(p^k))^{m\cdot s}$. Zdolności korekcyjne tak zdefiniowanego kodu są oczywiste. Widać od razu, że popełnienie co najwyżej r błędów w słowie kodowym (gdzie $r<\frac{s}{2}$) nie przeszkadza w poprawnym odczytaniu takiego słowa kodowego. Kod *s* krotnie powtórzony jest $(m,\ s\cdot m)$ kodem. ■

Kod blokowy. Niech V będzie ustalonym alfabetem. Kodem blokowym nazywamy kod $f : V^m \to V^n$, w którym słowom o długości m (obiektom kodowanym) przyporządkowujemy słowa o długości n, gdzie $n \geq m$.

Przykład: (m,n) kod jest kodem blokowym. ■

Kod blokowy nazywamy ***kodem grupowym***, jeśli słowa kodowe kodu tworzą grupę addytywną.

Różnych typów kodów korekcyjnych (ang. ECC od Error Correcting Codes) jest dosyć dużo. Z bardziej znanych warto wymienić:

- kody cykliczne (kody CRC - ang. Cyclic Redundancy Check)
- kody Hamminga
- kody Reeda-Solomona
- kody Bose-Chaudhuri-Hocquenghema (kody BCH)

Przykład: Kodowanie stosowane w płytach kompaktowych (płytach CD) to kod Reeda-Solomona, a ściślej CIRC (CIRC - ang. cross interleaved Reed-Solomon code). ■