# Laboratorium ochrony danych

**Ćwiczenie nr 3**

## Temat ćwiczenia: Kod BCH

*Cel dydaktyczny*: Zapoznanie się z metodami detekcji i korekcji błędów transmisyjnych za pomocą binarnych kodów cyklicznych, na przykładzie kodu Bose-Chaudhuri-Hocquenghema.

## Wprowadzenie teoretyczne

### Kody cykliczne

Kody cykliczne są podklasą kodów liniowych i znalazły największe zastosowania praktyczne. Popularność kodów cyklicznych wynika z następujących ich zalet:

- istnieją efektywne algebraiczne metody konstrukcji kodów cyklicznych o wymaganych właściwościach,
- realizacja koderów i dekoderów kodów cyklicznych za pomocą rejestrów przesuwnych ze sprzężeniem zwrotnym jest stosunkowo prosta.

W algebrze kodów cyklicznych ciągi informacyjne i kodowe zapisuje się w postaci wielomianów, a właściwości kodów opisuje się za pomocą pojęć z zakresu pierścieni wielomianów i ciał Galois.

Nazwa kodów cyklicznych pochodzi od właściwości przesunięcia cyklicznego, którą spełniają wektory kodowe. Stąd wywodzi się też definicja kodu cyklicznego.

Kod $(n,k)$ jest kodem cyklicznym, jeśli każdy wektor kodowy

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

po *i*-tym przesunięciu cyklicznym daje wektor

$$\boldsymbol{c}_i = \left[a_{n-1-i}, a_{n-2-i}, \ldots, a_1, a_0, a_{n-1}, a_{n-2}, \ldots, a_{n-i}\right]$$

będący również wektorem kodowym tego kodu.

Wektor kodowy $\boldsymbol{c}$ można zapisać w postaci wielomianu

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

W teorii blokowych kodów cyklicznych wykorzystuje się pojęcie pierścienia klasy reszt modulo $x^n - 1$. W dalszych rozważaniach ograniczymy się do kodów nad $GF(2)$. Pierścień klas reszt modulo $x^n + 1$ jest pierścieniem wielomianów stopnia nie większego niż $n-1$, które odpowiadają ciągom binarnym o długości *n*.

### Wielomiany generujące kody BCH

Ideałem jest podzbiór wielomianów pierścienia generowany przez pewien wielomian $g(x)$, który jest dzielnikiem $x^n + 1$. Ideał ten stanowi kod, a wielomian $g(x)$ nazywamy *wielomianem generującym kod*. Wielomian $g(x)$ dzieli bez reszty każdy wielomian odpowiadający wektorowi kodowemu. Stopień wielomianu generującego kod określa liczbę elementów kontrolnych wektora kodowego.

Z powyższych rozważań wynika, że wielomianem generującym kod cykliczny może być każdy wielomian, który jest podzielnikiem $x^n + 1$, gdzie $n = q^m - 1$, a *m* jest liczbą naturalną.

Kody Bose-Chaudhuri-Hocquenghema (BCH) należą do kodów korygujących błędy losowe i mają duże znaczenie praktyczne. Zostały one niezależnie skonstruowane przez Hocquenghema w 1959 r. oraz przez Bose z Chaudhurim w 1960 r. Kody BCH swoją popularność zawdzięczają następującym zaletom:

- Istnieją efektywne metody konstruowania kodów BCH o zadanych właściwościach detekcyjnych i korekcyjnych.
- Konstrukcja koderów i dekoderów kodów BCH jest prostsza niż dla innych kodów.

Kody BCH można konstruować nad ciałem binarnym i ciałami rozszerzonymi. Największe znaczenie mają kody binarne. Udowodniono, że dla każdej liczby całkowitej $m$ i $t < 2^{m-1}$ istnieje kod BCH o długości $n = 2^m - 1$. Może on korygować do $t$ błędów i ma nie więcej niż $mt$ elementów kontrolnych. Kody te mają następujące parametry:

- długość wektora kodowego $n = 2^m - 1$,
- liczba pozycji kontrolnych $n - k \le mt$,
- odległość minimalna $d \ge 2t + 1$.

Wielomiany generujące kody BCH wyznacza się w następujący sposób. Niech $\alpha$ będzie elementem pierwotnym ciała $GF(2^m)$. Zbiór $\{f(x)\}$ jest zbiorem ciągów kodowych kodu BCH, jeśli pierwiastkami dowolnie wybranego wielomianu $f(x)$ są elementy ciała

$$\alpha, \alpha^2, \alpha^3, \ldots, \alpha^{2t}.$$

Każdy element ciała o parzystym wykładniku ma w tej sekwencji taką samą funkcję minimalną jak któryś z poprzedzających go elementów o wykładniku nieparzystym. Na przykład $\alpha^2$ i $\alpha^4$ są pierwiastkami $m_1(x)$, $\alpha^6$ jest pierwiastkiem $m_3(x)$ itd. Uwzględniając ten fakt podczas wyznaczania wielomianu generującego kod BCH, wystarczy wziąć pod uwagę elementy ciała z wykładnikami nieparzystymi.

Wielomian generujący kod BCH o zdolności korekcyjnej $t$ jest najmniejszą wspólną wielokrotnością funkcji minimalnych $m_1(x), m_3(x), \ldots, m_{2t-1}(x)$

$$g(x) = NWW\left(m_1(x), m_3(x), \ldots, m_{2t-1}(x)\right).$$

P r z y k ł a d

Wyznaczanie wielomianów generujących kody BCH.

Dla $m = 4$ dwumian $x^{q^m - 1} - 1$ ma następujący rozkład

$$x^{15} + 1 = (x+1)\left(x^4 + x + 1\right)\left(x^4 + x^3 + x^2 + x + 1\right)\left(x^2 + x + 1\right)\left(x^4 + x^3 + 1\right).$$

Wielomiany nierozkładalne z prawej strony znaku równości są wielomianami minimalnymi elementów ciała $GF(2^4)$. Podstawiając symbole wielomianów minimalnych, otrzymamy

$$x^{15} + 1 = m_0(x)\, m_1(x)\, m_3(x)\, m_5(x)\, m_7(x).$$

Korzystając z tego wyrażenia, dla zadanych wartości $t$ można wyznaczyć wielomiany generujące kody BCH.

$t = 1, \quad g(x) = m_1(x),$ kod Hamminga (15,11);

$t = 2, \quad g(x) = m_1(x)\, m_3(x),$ kod (15,7);

$t = 3, \quad g(x) = m_1(x)\, m_3(x)\, m_5(x),$ kod (15,5).

Po prawej stronie wielomianów generujących podano parametry kodów $(n,k)$. Na przykład dla $t = 2, \ (n,k) = (15,7)$. Aby obliczyć liczbę pozycji informacyjnych $k$ wektora

kodowego, należy wyznaczyć stopień wielomianu generującego. Dla kodu $(n,k)=(15,7)$ otrzymamy wielomian generujący ósmego stopnia

$$g(x)=\left(x^4+x+1\right)\left(x^4+x^3+x^2+x+1\right)=x^8+x^7+x^6+x^4+1.$$

Wektor kodowy będzie zatem zawierał osiem pozycji kontrolnych i siedem informacyjnych. Odległość minimalna tego kodu jest $d\geq 5$ i może on korygować dwa błędy.

Dla $t=1$ kod BCH ma wielomian generujący

$$g(x)=m_1(x)=x^4+x+1\,.$$

Kod BCH korygujący jeden błąd jest jednocześnie kodem Hamminga. Generalnie kody Hamminga są podzbiorem kodów BCH.

## Tabela kodów cyklicznych

Problem konstrukcji kodów cyklicznych sprowadza się do syntezy wielomianu generującego kod i wyznaczenia jego odległości minimalnej. Zadania te są dość trudne, dlatego też często posługujemy się tablicami zawierającymi wielomiany generujące kody cykliczne i ich parametry.

Wybrany zestaw wielomianów kodów cyklicznych podano w tabeli. Kolejne kolumny tej tabeli zawierają: długość wektora kodowego *n*, liczbę pozycji informacyjnych *k*, odległość minimalną $d$ i wielomian generujący kod $g(x)$.

Wielomian generujący kod podano w postaci współczynników w systemie ósemkowym. Aby na podstawie tabeli wyznaczyć wielomian generujący, należy liczbę ósemkową zamienić na liczbę dwójkową. Na przykład $2467_8=10\ 100\ 110\ 111_2$, gdzie współczynnik zmiennej o najwyższej potędze znajduje się z lewej strony ciągu. Stąd wielomianem generującym kod cykliczny $(15,5)$, korygującym trzy błędy, będzie wielomian 10 stopnia:

$$g(x)=x^{10}+x^8+x^5+x^4+x^2+x+1.$$

Tabela. Wielomiany generujące binarnych kodów cyklicznych

| $n$ | $k$ | $d$ | $g(x)$ | $n$ | $k$ | $d$ | $g(x)$ | $n$ | $k$ | $d$ | $g(x)$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 4 | 3 | 13 | 33 | 20 | 6 | 20741 | 45 | 22 | 8 | 63335065 |
| | 3 | 4 | 35 | | 13 | 10 | 4172741 | | 21 | 3 | 110111011 |
| 15 | 11 | 3 | 23 | | 12 | 10 | 14217043 | | 20 | 6 | 330333033 |
| | 10 | 4 | 65 | | 11 | 11 | 25456465 | 47 | 24 | 11 | 43073357 |
| | 7 | 5 | 721 | | 10 | 12 | 76563537 | | 23 | 12 | 145115461 |
| | 6 | 6 | 1163 | 35 | 25 | 4 | 2565 | 49 | 28 | 3 | 10040001 |
| | 5 | 7 | 2467 | | 24 | 4 | 7637 | | 27 | 4 | 30140003 |
| | 4 | 8 | 7531 | | 23 | 3 | 13627 | | 21 | 4 | 2010040001 |
| 17 | 9 | 5 | 727 | | 20 | 6 | 147257 | 51 | 43 | 3 | 433 |
| | 8 | 6 | 1171 | | 19 | 6 | 251761 | | 35 | 5 | 266251 |
| 21 | 16 | 3 | 61 | | 18 | 4 | 735235 | | 33 | 6 | 1403537 |
| | 15 | 4 | 123 | | 17 | 6 | 1532051 | | 32 | 6 | 2020213 |
| | 12 | 5 | 1663 | | 16 | 7 | 2433361 | | 27 | 9 | 134531443 |
| | 11 | 6 | 2531 | | 15 | 8 | 7455423 | | 26 | 10 | 242245105 |
| | 10 | 5 | 5031 | | 11 | 5 | 143676743 | | 24 | 10 | 1762776477 |
| | 9 | 8 | 17053 | | 10 | 10 | 244303045 | | 19 | 14 | 50112257553 |
| | 5 | 10 | 214537 | 39 | 27 | 3 | 13617 | | 18 | 14 | 170336760675 |
| 23 | 12 | 7 | 5343 | | 26 | 6 | 34221 | | 10 | 18 | 62066722733023 |
| | 11 | 8 | 17445 | | 25 | 3 | 55263 | 55 | 35 | 5 | 7164555 |
| 25 | 5 | 5 | 4102041 | | 24 | 6 | 167725 | | 34 | 8 | 11235667 |
| | 4 | 10 | 14306143 | | 15 | 10 | 153651205 | 57 | 39 | 3 | 1341035 |
| 27 | 9 | 3 | 1001001 | | 14 | 10 | 274373617 | | 38 | 6 | 3443047 |
| | 8 | 6 | 3003003 | | 13 | 12 | 423136633 | 63 | 57 | 3 | 103 |
| 31 | 26 | 3 | 45 | 41 | 21 | 9 | 6647133 | | 56 | 4 | 305 |
| | 25 | 4 | 157 | | 20 | 10 | 13351355 | | 51 | 5 | 12471 |
| | 21 | 5 | 3551 | 43 | 29 | 6 | 64213 | | 45 | 7 | 1701317 |
| | 16 | 7 | 107657 | | 28 | 6 | 134635 | | 39 | 9 | 166623567 |
| | 15 | 8 | 310361 | | 15 | 13 | 2607043415 | | 36 | 11 | 1033500423 |
| | 11 | 11 | 5423325 | | 14 | 14 | 7211144427 | | 30 | 13 | 157464165547 |
| | 6 | 15 | 313365047 | 45 | 35 | 4 | 2113 | | 24 | 15 | 17323260404441 |
| | 5 | 16 | 535437151 | | 29 | 5 | 230213 | | 18 | 21 | 1363026512351725 |
| 33 | 23 | 3 | 3043 | | 25 | 5 | 7217531 | | 16 | 23 | 6331141367235453 |
| | 22 | 6 | 5145 | | 24 | 6 | 11620753 | | 10 | 27 | 472622305527250155 |
| | 21 | 3 | 17537 | | 23 | 7 | 21113023 | | 7 | 31 | 523104543503271737 |

## Algorytm kodowania

Do kodowania informacji za pomocą kodów cyklicznych można wykorzystać wielomian generujący kod lub macierz generującą kod. Tutaj rozważymy kodowanie za pomocą wielomianu generującego.

Wektor kodowy cyklicznego kodu systematycznego ma formę:

$$\boldsymbol{c}_X = [m_{n-1}, \ldots, m_{n-k}, r_{n-k-1}, \ldots, r_0],$$

gdzie współrzędne $m_i$ są elementami informacyjnymi, a współrzędne $r_i$ – elementami kontrolnymi.

Gdy mamy wielomian generujący $g(x)$ stopnia $n-k$, to aby obliczyć wektor kodowy systematycznego kodu cyklicznego $(n,k)$, należy wykonać następujące czynności:

Wielomian odpowiadający informacji $m(x)$ pomnożyć przez $x^{n-k}$, tj.

$$x^{n-k} m(x).$$

2. Otrzymany iloczyn $x^{n-k}m(x)$ podzielić przez wielomian generujący kod $g(x)$ i wyznaczyć resztę $r(x)$ z tego dzielenia

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

3. Obliczyć wielomian $c_X(x)$, odpowiadający wektorowi kodowemu, dodając $x^{n-k}m(x)$ i resztę $r(x)$ (dodajemy modulo 2 i otrzymamy wektor kodowy podzielny bez reszty przez $g(x)$):

$$c_X(x) = x^{n-k}m_N(x) + r(x).$$

Napiszmy ciąg informacyjny w postaci wielomianu

$$m(x) = m_{k-1}x^{k-1} + m_{k-2}x^{k-2} + \ldots + m_1 x + m_0.$$

Pomnożenie tego wielomianu przez $x^{n-k}$ jest równoważne z przesunięciem wektora kodowego w lewo o $n-k$ pozycji

$$m(x)\,x^{n-k} = m_{k-1}x^{n-1} + m_{k-2}x^{n-2} + \ldots + m_1 x^{n-k+1} + m_0 x^{n-k}.$$

Dzielenie uzyskanego wyrażenia przez $g(x)$ można zapisać w formie algorytmu Euklidesa

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

gdzie $q(x)$ jest częścią całkowitą, a $r(x)$ resztą z dzielenia w postaci

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

Wzór umożliwiający obliczenia części kontrolnej wektora kodowego możemy również zapisać w formie kongruencji

$$\sum_{i=1}^{k} m_{n-i}x^{n-i} \equiv \sum_{i=1}^{n-k} r_{n-k-i}x^{n-k-i} \left(\mathrm{mod}\ g(x)\right).$$

Wielomian z lewej strony kongruencji odpowiada części informacyjnej wektora kodowego, a wielomian z prawej strony – części kontrolnej wektora kodowego, która jest równa reszcie z dzielenia wielomianu informacyjnego przez wielomian generujący kod $g(x)$. Otrzymany wektor kodowy $c_X(x)$ jest podzielny bez reszty przez $g(x)$.

## Uproszczony algorytm dekodowania

W czasie transmisji wektorów kodowych kanałem transmisyjnym powstają błędy transmisyjne. Zadaniem dekodera jest wykrycie lub wykrycie i usunięcie tych błędów.

Każdy kod cykliczny ma swój algorytm dekodowania, który pozwala skorygować wszystkie błędy korygowalne przez dany kod, tj. nie przekraczające jego zdolności korekcyjnej $t$. W praktyce często używa się algorytmu uproszczonego, wspólnego dla wszystkich kodów cyklicznych. Algorytm ten umożliwia wykrycie i korekcję wszystkich błędów znajdujących się na $n-k$ pozycjach kontrolnych wektora kodowego, o ile ich liczba nie przekracza zdolności korekcyjnej kodu. W przypadku, gdy błędy znajdują się również w części informacyjnej wektora kodowego możliwość korekcji błędów zależy od ich rozłożenia w wektorze kodowym, nawet jeśli ich liczba nie przekracza zdolności korekcyjnej kodu. Algorytm ten omówimy szczegółowo.

W procesie dekodowania oblicza się syndrom wektora odebranego. Dla kodów cyklicznych syndrom oblicza się, dzieląc wielomian $c_Y(x)$, odpowiadający wektorowi odebranemu $\boldsymbol{c_Y}$, przez wielomian generujący kod $g(x)$. Syndrom $s(x)$ jest równy reszcie z tego dzielenia

$$c_Y(x) = q(x)\,g(x) + s(x).$$

Syndrom $s(x)$ jest wielomianem stopnia $\leq n-k-1$. Jeśli syndrom ma wartość zerową, oznacza to, że wektor odebrany jest wektorem kodowym i w czasie transmisji nie wystąpiły żadne błędy wykrywalne przez kod (może się zdarzyć, że błędy rozłożą się w taki sposób, że wektor odebrany będzie innym wektorem danego kodu – wtedy nie zostaną one wykryte). Niezerowa wartość syndromu świadczy o tym, że odebrany wektor nie jest wektorem kodowym i zostały wykryte błędy transmisyjne.

Wektor odebrany $\boldsymbol{c_Y}$, jest sumą wektora nadanego $\boldsymbol{c_X}$ i wektora błędów $\boldsymbol{e}$. Wzór ten zapisujemy w postaci wielomianów

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

Wielomian odpowiadający wektorowi kodowemu $\boldsymbol{c_X}$ dzieli się bez reszty przez wielomian generujący kod $g(x)$, można zatem napisać

$$c_X(x) = m(x)\,g(x).$$

Podstawiając tę zależność do wzoru poprzedniego, otrzymamy

$$c_Y(x) = m(x)\,g(x) + e(x).$$

Porównujemy prawą stronę tego wzoru z prawą stroną wzoru pierwszego. Po przekształceniach mamy

$$e(x) = \big(m(x) + q(x)\big)g(x) + s(x).$$

Syndrom jest resztą z dzielenia wielomianu odpowiadającego wektorowi błędów $e(x)$ przez wielomian generujący kod $g(x)$. Syndrom zawiera informację o położeniu błędów transmisyjnych, co jest wykorzystywane w trakcie korekcji błędów.

Schemat blokowy algorytmu dekodowania z korekcją błędów pokazano na rysunku. Na rysunku tym zastosowano takie same oznaczenia jak w opisie algorytmu. Proces dekodowania ma następujący przebieg.

Po wyznaczeniu syndromu oblicza się jego wagę Hamminga $w(s)$. Mogą wówczas wystąpić następujące przypadki:

1. Waga syndromu jest mniejsza lub równa zdolności korekcyjnej kodu, $w(s) \leq t$. Oznacza to, że błędy są położone w części kontrolnej wektora kodowego (poprawną część informacyjną można otrzymać bezpośrednio z wektora odebranego bez korygowania). Wektor odebrany $c_Y$ może być wtedy skorygowany dzięki dodaniu syndromu do wektora odebranego. W wyniku tego działania otrzymamy wektor wyjściowy dekodera $c_D$:

$$c_D = c_Y + s\,.$$

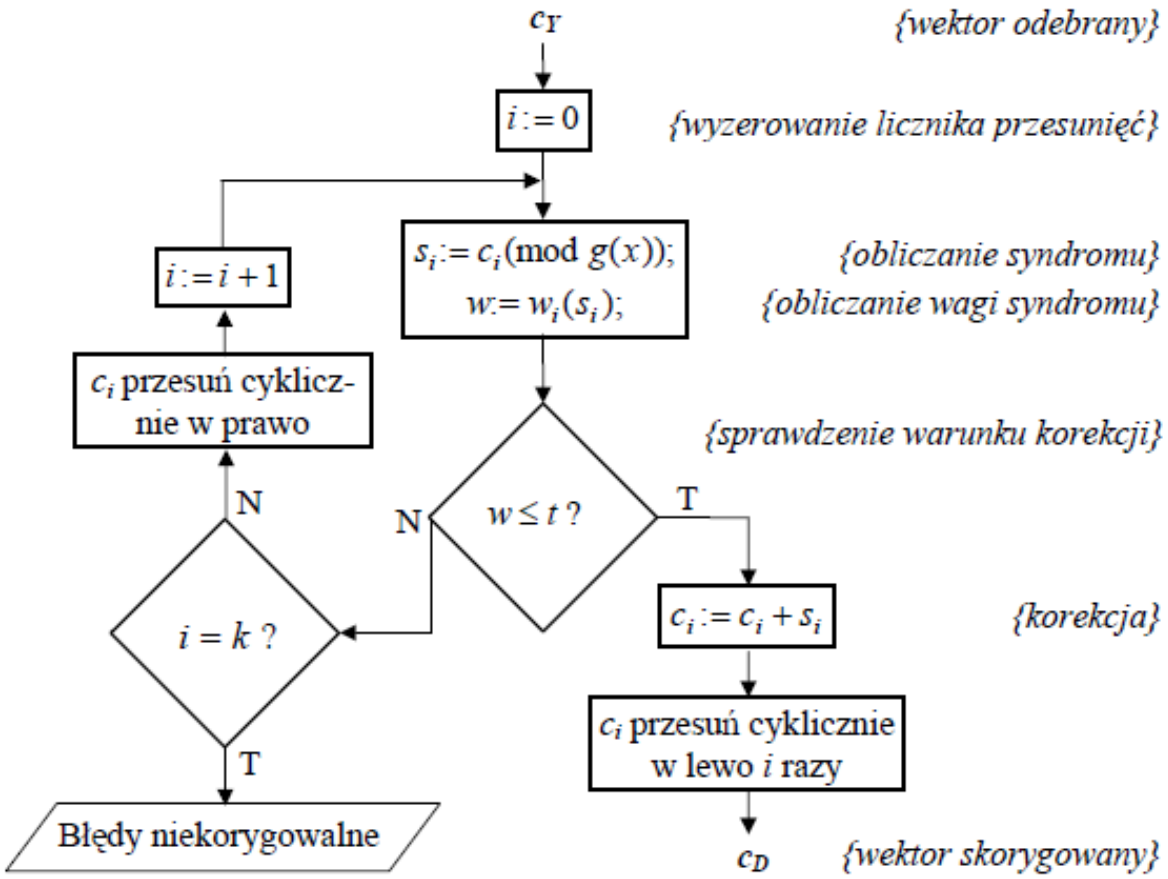
Na podstawie tego wektora można wyznaczyć informację odebraną $m^*$. Będzie ona równa części informacyjnej wektora $c_D$.

2. Waga syndromu jest większa od zdolności korekcyjnej kodu, $w(s) > t$. Przypadek ten oznacza, że błędy obejmują część informacyjną wektora kodowego. Należy wówczas przesunąć cyklicznie wektor odebrany tak, aby błędy znalazły się w części kontrolnej, a potem go skorygować. W tym celu wykonujemy następujące czynności. Przesuwamy wektor odebrany cyklicznie o jedną pozycję w dowolnym kierunku (np. w prawo), obliczamy syndrom i jego wagę oraz sprawdzamy, czy został spełniony warunek podany w p. 1, czy też warunek podany w p. 2.

- Jeżeli $w(s) \leq t$, należy skorygować wektor odebrany zgodnie z p. 1, a następnie przesunąć go cyklicznie w odwrotną stronę (w lewo), aby odtworzyć jego pierwotną postać.

- Jeżeli $w(s) > t$, trzeba ponownie przesuwać cyklicznie wektor odebrany w tę samą stronę, obliczając po każdym przesunięciu syndrom i jego wagę aż do momentu, kiedy $w(s) \le t$. Wtedy należy skorygować wektor odebrany i przesunąć go w odwrotną stronę o taką samą liczbę pozycji.

- W przypadku gdy po $k$ przesunięciach cyklicznych nie uda się skorygować wektora odebranego oznacza to, że wystąpiły błędy niekorygowalne, np. istnieją bity błędne odległe o więcej pozycji w wektorze odebranym niż wynosi liczba bitów części kontrolnej (nie dało się przesunąć bitów błędnych do części kontrolnej).




Rysunek. Schemat blokowy uproszczonego algorytmu dekodowania

Uproszczony algorytm dekodowania nie zapewnia wykorzystania pełnych zdolności korekcyjnych kodów korygujących błędy wielokrotne. Wynika to z faktu, że w przypadku błędów wielokrotnych nie zawsze jest możliwe przesunięcie błędnych elementów do części kontrolnej wektora kodowego.

Algorytm może również nie zadziałać w przypadku, gdy liczba błędów w wektorze kodowym przekracza zdolność korekcyjną kodu $t$.

Kody BCH mają własny algorytm dekodowania, który zapewnia korekcję wszystkich błędów nie przewyższających zdolności korekcyjnej danego kodu. Algorytm ten jest jednak bardzo złożony. Jego opis można znaleźć w literaturze.

Algorytm kodowania i obliczania syndromu wykorzystują dzielenie wielomianów. W realizacji programowej tych algorytmów można zastosować symulacje układu do dzielenia wielomianów.

## Opis oprogramowania

Dla zademonstrowania właściwości kodu BCH opracowano program komputerowy BCH_P.CPP w języku C++, który udostępniono w wersji wykonywalnej. Fragmenty tego programu, bez funkcji obliczania syndromu (Calculate_Syndrom), podano w postaci źródłowej. Funkcja obliczania syndromu ma postać:

```
void Calculate_Syndrom(int slowo[], int syndrom[])
{
  /* wg[r+1]    - wielomian generujący          */
  /* slowo[n]   - wektor kodowy                 */
  /* syndrom[r] - obliczony syndrom             */

  //  dzielenie wielomianu zapisanego w slowo[ ] przez wg[ ]
  //  slowo[0],  wg[0]   - współczynniki przy najwyższej potędze wielomianów
  //  np.  slowo[n] = {1,0,0,0,0,0,0}
  //         wg[r+1] = { 1,0,1,1 }
  //         syndrom[] = {1, 0, 1} = (slowo) mod (wg) reszta z dzielenia;

  //  wstawić obliczanie syndromu }
```

Dodatkowo w zbiorze BCCGPOL.TXT podano zestaw wielomianów generujących kody cykliczne (w tym kody BCH). Kolejne kolumny zawierają:

$n$ – długość wektora kodowego, $k$ – liczba symboli informacyjnych,
$r$ – liczba symboli kontrolnych, $d$ – minimalna odległość Hamminga,
$g(x)$ – wielomian generujący kod, zapisany w systemie ósemkowym.

## Przebieg ćwiczenia

1. Uruchomić program BCH_P i prześledzić jego działanie. Zwrócić uwagę na możliwości korekcyjne kodu dla wybranych wektorów wiadomości, a także dla różnych postaci wektora błędów (różnej liczby błędów oraz różnego ich rozkładu w wektorze). Wersja wykonywalna implementuje kod BCH o parametrach $(n, k) = (15, 7)$; $r = 8$; $d = 5$; $t = 2$; $g(x) = 7\ 2\ 1$.

2. W programie BCH_P.CPP do obliczania syndromu używa się funkcji Calculate_Syndrom (realizuje obliczanie reszty z dzielenia wielomianu slowo przez wielomian wg). Funkcja nie została zamieszczona w kodzie źródłowym BCH_ST. Napisać własną funkcję obliczania syndromu wektora kodowego i dołączyć ją do programu BCH_ST. Uruchomić program.

3. W zbiorze BCCGPOL.TXT znaleźć wielomiany generujące kody BCH: (7, 4), (15, 7), (31, 21), (51, 24). Zmodyfikować program tak aby realizował te kody. Sprawdzić działanie kodów dla przypadków, gdy liczba błędów w wektorze kodowym przekracza zdolność korekcyjną kodu, lub gdy błędy wielokrotne są odlegle od siebie o więcej niż wynosi długość części kontrolnej wektora kodowego, lub gdy błędy rozkładają się w taki sposób, że wektor odebrany jest innym wektorem danego kodu.

Zademonstrować działanie programu podczas laboratorium. Przysłać krótkie sprawozdanie zawierające wyniki testów funkcjonalnych opracowanego programu. Dopisać wnioski z działania uproszczonego algorytmu dekodowania.