Cezary Wernik

Asystent, KAKiT, WI, ZUT

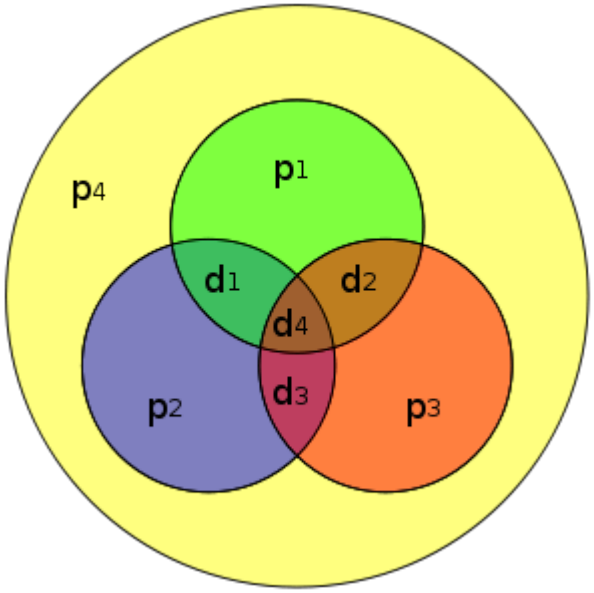
8. Kodowanie kanałowe

Uwaga: Studencie! – na koniec zajęć laboratoryjnych **bezwzględnie zaktualizuj** swoje repozytorium/e-dysk, zawierające prace z zajęć laboratoryjnych tego przedmiotu. Brak systematycznych aktualizacji repozytorium może zostać uznany za brak dokumentacji postępu w realizacji zadań laboratoryjnych, co może skutkować oceną niedostateczną.

Skrót z teorii:

Aby zabezpieczyć transmitowane dane przed potencjalnymi błędami, stosuje się kodowanie kanałowe mające uchronić strumień binarny przed przekłamaniami bitów i/lub działać prewencyjnie informując odbiorcę, że zaistniał błąd i należy powtórzyć transmisję.

Do kodów kanałowych należy protekcyjny kod Hamminga(7,4). Schemat wyznaczania bitów parzystości dla tego kodu łatwo zapamiętać dzięki poniższemu diagramowi:



[https://en.wikipedia.org/wiki/Hamming_code]

gdzie:
p_n – bity parzystości
d_n – bity danych

Ogólny algorytm tworzenia kodu:

- Numery bitów (n) indeksuj od 1 w górę;
- Bity na indeksach wyjściowych, będących potęgą dwójki będą bitami parzystości (p_n);
- Pozostałe bity (d_n) uzupełnij po kolei danymi wejściowymi;
- Każdy bit parzystości wskazuje parzystość pewnej grupy bitów wejściowych, a jego pozycja określa, które bity danych (d_n) biorą udział w wyliczeniu parzystości zgodnie z wyżej wskazanym diagramem..

Przykładowa tabela rozmieszczenia bitów danych i bitów parzystości:

Bit position		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Encoded data bits		p1	p2	d1	p4	d2	d3	d4	p8	d5	d6	d7	d8	d9	d10	d11	p16	d12	d13	d14	d15	
Parity bit coverage	p1	✗		✗		✗		✗		✗		✗		✗		✗		✗		✗		…
	p2		✗	✗			✗	✗			✗	✗			✗	✗			✗	✗		
	p4				✗	✗	✗	✗					✗	✗	✗	✗					✗	
	p8								✗	✗	✗	✗	✗	✗	✗	✗						
	p16																✗	✗	✗	✗	✗	

[https://en.wikipedia.org/wiki/Hamming_code]

Przykładowa metoda implementacji macierzowej – kod Hamminga(7,4) można zrealizować macierzowo, poprzez przemnożenie macierzowo wektora pionowego d i macierzy G:

$$\mathbf{G} := \begin{pmatrix} 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Do sprawdzenia poprawności bitów parzystości używa się dla kodu Hamminga (7,4) poniższej macierzy:

$$\mathbf{H} := \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Suplement do instrukcji znajduje się pod adresem:
http://wernik.zut.edu.pl/shared/Hamming(7,4).pdf

Procedura postępowania przy SECDED na poniższym rysunku:

SECDED

NADAWCA

B=[1 0 1 0 1 1 1 0]

$$\mathbf{d} = [d_1 \quad d_2 \quad d_3 \quad d_4]^T$$

$$\text{data}_{\text{byte n}} [b_8 \quad b_7 \quad b_6 \quad b_5 \quad b_4 \quad b_3 \quad b_2 \quad b_1]$$

$$\mathbf{d} = [d_1 \quad d_2 \quad d_3 \quad d_4]^T$$

d=[1 0 1 0] d=[1 1 1 0]

p1=(1+0+0)%2=1
p2=(1+1+0)%2=0
p3=(0+1+0)%2=1

h=[1 0 1 1 0 1 0]

p4=(1+0+1+1+0+1+0)%2=0

h=[1 0 1 1 0 1 0 0]

ODBIORCA

* w trakcie transmisji czerwone bity zostały zanegowane

h=[1 0 0 1 1 1 0 0]

Krok 1)
Weryfikacja p4

Są równe możemy iść dalej :)

p4 = (1+0+0+1+1+1+0)%2 = 0

Gdy w kroku 1 bit p4 już jest niezgodny to mamy 50% szans na naprawę pakietu w dalszych krokach.
Można na tym etapie też już zrezygnować z pakietu i oznaczyć go jako błędny ale warto podjąć próbę naprawy.

Krok 2)
Jeżeli p4 jest zgodny z otrzymanym nadawcy to sprawdź p1 p2 p3

$$p_1 = (h_1 + h_3 + h_5 + h_7) \bmod 2$$
$$p_2 = (h_2 + h_3 + h_6 + h_7) \bmod 2$$
$$p_3 = (h_4 + h_5 + h_6 + h_7) \bmod 2$$

h=[1 0 0 1 1 1 0 0]

p1 = (1+0+1+0)%2=0
p2 = (0+0+1+0)%2=1
p3 = (1+1+1+0)%2=1

Krok 3)
Wyznaczenie indeksu korekty

$$n = p_1 \cdot 2^0 + p_2 \cdot 2^1 + p_3 \cdot 2^2$$

n=0·1+1·2+1·4 = 6

Krok 4)
Negacja bitu wskazanego przez indeks korekty

h=[1 0 0 1 1 1 0 0]

h=[1 0 0 1 1 0 0 0]

Krok 5)
Ponowna weryfikacja p4

Są różne, decyzja do podjęcia w kroku 6 :(

p4 = (1+0+0+1+1+0+0)%2 = 1

Krok 6) Decyzja:

Gdy po korekcie bitu wskazanego przez n oraz ponownej weryfikacji bitu p4, bit wyliczony ponownie jest zgodny z otrzymanym w transmisji to znaczy, że były tylko 1 błąd, można przepisać bity danych do wyjścia.

W przeciwnym przypadku, gdy w kroku 5 weryfikowany bit p4 będzie niezgodny z otrzymanym, to znaczy że w transmisji były conajmniej 2 błędne bity.
Pakiet bezwzględnie odrzucamy, prosimy

nadawcę o ponowną transmisję.

Zadanie:

Wykonaj w formie programistycznej implementacji poniżej przedstawione zadania.

1) Zaimplementuj funkcję kodującą kodem Hamming (7,4) zadany strumień binarny. Do generowania strumienia binarnego użyj funkcji **S2BS** napisanej na laboratoriach „5. Modulacja dyskretna”.

2) Napisz funkcję negującą wskazany numer bitu w strumieniu binarnym z zadania pierwszego.

3) Zaimplementuj funkcję dekodującą kod Hamminga (7,4), sprawdź poprawność działania.

4) Zaimplementuj rozwinięcie (koder i dekoder) kodu Hamming (7,4) jako kod SECDED, posiadający dodatkowy bit parzystości. Zweryfikuj działanie poprzez zanegowanie dwóch bitów w strumieniu binarnym.

Łącznie w wyniku działania twojego kodu powinno zostać wygenerowanych 6 strumieni binarnych, zanotuj je w formie komentarza w kodzie programu.

Kody spakuj w katalog i umieść na swoim repozytorium.

< Poprzedni temat

Wydrukuj instrukcję