

Telekomunikacja - laboratorium			Studia dzienne - inżynierskie		
Nazwa zadania		Kody wykrywające i korygujące błędy transmisji.			
Dzień	poniedziałek	Godzina	12:15 – 13:45	Rok akademicki	2019/2020
Imię i Nazwisko		Konrad Chojnacki 224274			
Imię i Nazwisko		Krzysztof Szcześniak 224434			
Imię i Nazwisko		-----			
Opis programu, rozwiązania problemu.					
<div>1. Program odczytuje wiadomość zawartą w pliku tekstowym, a następnie zamienia odczytane znaki na system binarny.</div> <div>2. W celu zwiększenia odległości pomiędzy słowami program dodaje do każdego znaku (tj. 8 bitów) 8 bitów parzystości. Do wyznaczenia bitów parzystości program korzysta z zależności $HT = 0$, gdzie H jest zdefiniowaną macierzą 8×16 – podana w kolejnym podpunkcie sprawozdania, a T jest to wektor 8 bitowej wiadomości rozszerzony o 8 bitów parzystości.</div> <div>$c_1 = (h_{11}a_{11} + h_{12}a_{12} + \dots + h_{18}a_{18}) \bmod 2$$c_2 = (h_{21}a_{21} + h_{22}a_{22} + \dots + h_{28}a_{28}) \bmod 2$</div> <div>Analogicznie wyznaczono c3, c4, ..., c8.</div> <div>3. Przed weryfikacją można dokonać modyfikacji pliku z zakodowaną postacią w celu wywołania błędu.</div> <div>4. Etap weryfikacji polega na obliczeniu iloczynu macierzy H z każdym wektorem 16 bitowym otrzymując wektor HE.</div> <div>5. Jeżeli otrzymany wektor HE jest wektorem składającym się z samych 0 to oznacza to, że odebrane słowo jest prawidłowe.</div> <div>6. Jeżeli otrzymany wektor HE nie jest wektorem zerowym oznacza to, że wystąpił błąd transmisji i należy go skorygować.</div> <div>7. Etap korygowania błędu polega na porównaniu wektora HE z kolumnami macierzy H. Numer kolumny, która jest taka sama jak w wektorze HE, wskazuje pozycje bitu na którym nastąpiło przekłamanie. Należy więc zanegować ten bit w celu poprawienia słowa. Tak wygląda etap korygowania dla pojedynczego błędu, trochę inaczej jest w przypadku podwójnego błędu. W tym przypadku należy porównywać otrzymany wektor HE z sumą dwóch kolumn macierzy H. Numery tych kolumn których suma jest równa wektorowi HE oznacza to, że bity na pozycjach równych numerom tych kolumn są przekłamate i również jak w poprzednim przypadku należy zanegować te dwa bity w celu skorygowania słowa.</div>					
Najważniejsze elementy kodu programu z opisem.					
<div>1. Zadeklarowanie macierzy H:</div> <div><pre>bool H[8][16] = {{1,1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0}, {1, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0}, {1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0}, {1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0}, {0, 1, 0, 1, 0, 1, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0}, {0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0}, {0, 0, 1, 1, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0}, {0, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1}};</pre></div> <div>Dobór tej macierzy nie jest przypadkowy. Musi ona spełniać warunki:</div> <div><ul style="list-style-type: none">nie może mieć identycznych kolumnnie może mieć kolumny zerowejw przypadku korygowania 2 błędów wymagane jest aby suma każdych dwóch kolumn była różna.</div> <div>2. Funkcja służąca do obliczania bitów parzystości:</div> <div><pre>vector<bool>addParityBits(vector<bool>message, boolcalculateHE=false) { vector<bool>parityBits; int k = 1; if (calculateHE == true) { k = 2; } for (int i = 0; i < 8; i++) { int c = 0; for (int j = 0; j < k*8; j++) { c += H[i][j]*message[j]; } c=c%2; parityBits.push_back(c); } }</pre></div>					

```
return parityBits;
}
```

W funkcji tej zaimplementowany został wzór podany w poprzednim punkcie sprawozdania. Dodatkowo funkcja ta wykorzystywana jest przez program w etapie weryfikowania do obliczenia iloczynu HE.

3. Funkcja realizująca etap weryfikacji:

```
void check(vector<bool>&message, int choice) {
    vector<bool> E;
    bool requiredCorrection = false;

    E = addParityBits(message, true);
    for (int i = 0; i < E.size(); i++) {
        if (E[i] == 1) {
            requiredCorrection = true;
        }
    }
    if (requiredCorrection == true) {
        if (choice == 2) {
            correctionv1(E, message);
        }
        else {
            correctionv2(E, message);
        }
    }
}
```

Jeśli wyznaczony wektor nazwany w tym przypadku E nie będzie zerowy, to konieczne jest przejście do etapu korygowania błędów w zależności od wybranej opcji jednego błędu lub dwóch.

4. Funkcja, która koryguje błędy:

```
void correctionv1(vector<bool>E, vector<bool>&message) {
    bool correction=false;

    for (int i = 0; i < 16; i++) {
        for (int j = 0; j < 8; j++) {
            if (E[j]==H[j][i]) {
                correction = true;
            }
            else {
                correction = false;
                break;
            }
        }
        if (correction == true) {
            message[i]= (~message[i])%2;
        }
    }
}
```

Funkcja ta służy do poprawienia pojedynczego błędu. Porównuje ona kolumny przekazanego wektora E (czyli iloczynu macierzy H z wektorem słowa 16 bitowego) z kolumną H i jeżeli znajdzie równość tych kolumn to zmienia wartość zmiennej na true i wskazuje wówczas pozycje przekłamanego bitu. Aby go poprawić należy go zanegować.

```
void correctionv2(vector<bool>E, vector<bool>&message) {
    bool correction = false;

    for (int i = 0; i < 16; i++) {
        for (int j = i+1; j < 16; j++) {
            for (int k = 0; k < 8; k++) {
                if (E[k] == (H[k][i] + H[k][j]) % 2) {
                    correction = true;
                }
                else {
                    correction = false;
                    break;
                }
            }
        }
    }
}
```

```
    }  
    if (correction == true) {  
        message[i]= (~message[i]) % 2;  
        message[j]= (~message[j]) % 2;  
    }  
}  
}
```

Powyższa funkcja działa podobnie jak poprzednia tylko ta służy do poprawy dwóch błędów, czyli porównywany jest wektor E (czyli iloczyn macierzy H i 16 bitowego słowa) z sumą dwóch kolumn macierzy H . Jeżeli zachodzi taka równość to zmienne „i” i „j” wskazują pozycję przekłamanych bitów, które należy zanegować w celu poprawienia słowa.

Podsumowanie wnioski.

Zaimplementowany przez nas kod umożliwia detekcję oraz korektę do dwóch błędów transmisji.

W celu umożliwienia tego zadania, kluczowe jest zadeklarowanie odpowiedniej macierzy H , dzięki której jednoznaczne jest określenie miejsca przekłamania, jak również dodawanie do kodowanych słów bitów parzystości.