Milena Michalska, nr Albumu 39362

Sprawozdanie z lab nr 8

Kodowanie kanałowe

Zadanie 1

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

Jako strumień binarny wykorzystałam moje imie:

```
m = S2BS('Milena', 'bigEndian')
```

100110101101001011011000110010101101110011000001

```
def kodowanie(s):
    b = ''
    # wczytaj po 4 bity
    while len(s) >= 4:
        porcja = s[0:4]
        b += str(hamming(porcja))
        s = s[4:]
    return b

def hamming(bits):
    t1 = str(parzystosc(bits, [0,1,3]))
    t2 = str(parzystosc(bits, [0,2,3]))
    t3 = str(parzystosc(bits, [1,2,3]))
    #zwraca podane bity + bity parzystości
    return t1 + t2 + ''.join(bits[0]) + t3 + ''.join(bits[1:])

def parzystosc(s, i):
    return (int(s[i[0]]) + int(s[i[1]]) + int(s[i[2]])) % 2
```

Wynik kodowania:

0011001101101010101010101010101010111100000111100101101010101010111100011110 0

Zadanie 2

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

```
def negacja(stream, bit):
    stream[bit] = 0 if (stream[bit] == 1) else 1
    return stream
```

Zadanie 3

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

```
def dekodowanie(s):
    b = ''
    # wczytaj po 7 bitów
    while len(s) >= 7:
        porcja = s[0:7]
        bits = hammingDecoding(porcja)
        b += bits
        s = s[7:]
    return b

def parzystoscD(s, i):
    sum = 0
    for index in i:
        sum += int(s[index])
    return sum % 2

def hammingDecoding(bits):
    #ponowne policzenie bitów parzystości
    t1 = int(parzystoscD(bits, [0,2,4,6]))
    t2 = int(parzystoscD(bits, [1,2,5,6]))
    t3 = int(parzystoscD(bits, [3,4,5,6]))

    n = (t1 * 2**0) + (t2 * 2**1) + (t3 * 2**2)
    if (n > 0):
        bits = negacja(bits, n - 1)
        print('W transmisji nastąpił błąd!')

    return ''.join(bits[[2,4,5,6]])
```

Oryginalny strumień: 1001101011010010110110001100101011011100110000 1

Zdekodowany strumień: 100110101101001011011000110010101101110011001

Działanie funkcji jest poprawne, tylko ucina ostatnie znaki, które nie weszły do 4 bitów :/