

1. Расчёт начальных условий.

Пусть объем кода (количество передаваемых сообщений) составит $N = 32$. Для систематического кода необходимо рассчитать необходимое число информационных и контрольных разрядов. Также определим, сколько ошибок должен обнаружить систематический код, и сколько должен исправить.

Расчеты произведены программно:

```
import numpy as np
from math import log2, ceil
import pandas as pd

d = 3
W_i = 1
N = 32
n_u = ceil(log2(N))
n_k = ceil(log2((n_u + 1) + log2(n_u + 1)))
n = n_k + n_u
W_p = d - W_i
print('Число информационных разрядов ----> ', n_u, '\nЧисло контрольных разрядов ----> ', n_k)
print()
```

На скриншоте выше: d – минимальное кодовое расстояние, при котором возможно исправить одну ошибку, W_p – минимальный вес каждой строки проверочной матрицы.

2. Порождающие и проверочные матрицы, уравнения проверки

Порождающая матрица состоит из информационной и проверочной. Сгенерируем проверочную матрицу P:

```
# ----  
  
P = []  
for i in range(2 ** n_k):  
    P.append(list(map(int, f'{i:0{n_k}b}')))  
P = list(filter(lambda x: sum(x) >= W_p, P))[:n_u]
```

Встроенными средствами библиотеки numpy сгенерируем единичную матрицу I, и, объединив, получим искомую порождающую матрицу C:

```
I = np.eye(n_u, dtype=int)  
P = np.array(P, dtype=int)  
# pprint(I)  
# pprint(P)  
C = np.concatenate([I, P], axis=1)  
print('Образующая матрица: -----> ')  
print(pd.DataFrame(C, columns=[f'_{i}' for i in range(n)]).to_csv(sep='\t', index=False))
```

Составим проверочную матрицу: $H = \|P^T I_{n_k}\|$:

```
H = np.concatenate([C[:, n_u:].T, np.eye(n_k, dtype=int)], axis=1)  
print('Контрольная матрица -----> ')  
col_name_1 = [f'a{i + 1}' for i in range(n_u)]  
col_name_2 = [f'p{i + 1}' for i in range(n_k)]  
col_name_1.extend(col_name_2)  
H_df = pd.DataFrame(H, index=None, columns=col_name_1)  
print(H_df.to_csv(sep='\t', index=False))
```

Дополнительно для некоторого исходного сообщения 'с' можно получить линейный групповой код:

```
c = np.array([1, 1, 0, 0, 1])  
linear_code = c.dot(C) % 2  
print(f'\nЛинейный код для ----> ', end=' ')  
print(''.join(list(map(str, map(int, list(c)))))), end=' -----> ')  
print(''.join(list(map(str, map(int, list(linear_code)))))), end=' # 110011111|'
```

Далее представлен код для ввода искаженного закодированного слова, код для системы проверок $P_j \oplus \sum_{i=1}^{n_u} P_{ij} a_i = S_j, j = 1..n_k$ и расчёта синдрома S:

```
p = correct_word[-n_k:]
lst = []
print('\nСистема проверок:')
for i in range(n_k):
    mask = H[i, :n_u]
    s = (np.sum(correct_word[:n_u][mask == 1]) + p[i]) % 2
    lst.append(s)
    print(' ⊕ '.join([H_df.columns[-n_k:][i]] + H_df.columns[:n_u][mask == 1].to_list()), end=' = ')
    print(' ⊕ '.join(list(map(str, map(int, [p[i]] + list(correct_word[:n_u][mask == 1]))))), ' = ', int(s))

print()
ind = np.array(lst)
print('Синдром ---->', ''.join(list(map(str, map(int, ind)))))
for i in range(H.shape[1]):
    if all(ind == H[:, i]):
        print('Инвертируемый разряд = ', i + 1)
```

3. Программный контроль пунктов 1 и 2.

Продemonстрируем работу программы на примере $N = 32$.

```
E:\PythonYandex\work\venv\Scripts\pytho
Число информационных разрядов ----> 5
Число контрольных разрядов ----> 4

Образующая матрица: ----->

1  0  0  0  0  0  0  1  1
0  1  0  0  0  0  1  0  1
0  0  1  0  0  0  1  1  0
0  0  0  1  0  0  1  1  1
0  0  0  0  1  1  0  0  1

Контрольная матрица ----->
a1 a2 a3 a4 a5 p1 p2 p3 p4
0  0  0  0  1  1  0  0  0
0  1  1  1  0  0  1  0  0
1  0  1  1  0  0  0  1  0
1  1  0  1  1  0  0  0  1

Линейный код для ----> 11001 -----> 110011111
Искаженное слово ----> 100011111

Система проверок:
p1 ⊕ a5 = 1 ⊕ 1 = 0
p2 ⊕ a2 ⊕ a3 ⊕ a4 = 1 ⊕ 0 ⊕ 0 ⊕ 0 = 1
p3 ⊕ a1 ⊕ a3 ⊕ a4 = 1 ⊕ 1 ⊕ 0 ⊕ 0 = 0
p4 ⊕ a1 ⊕ a2 ⊕ a4 ⊕ a5 = 1 ⊕ 1 ⊕ 0 ⊕ 0 ⊕ 1 = 1

Синдром ----> 0101
Инвертируемый разряд = 2
```

Также для примера положим $N = 16$:

```

E:\PythonYandex\work\venv\Scripts\python.exe E:/
Число информационных разрядов ----> 4
Число контрольных разрядов ----> 3

Образующая матрица: ----->

1  0  0  0  0  1  1
0  1  0  0  1  0  1
0  0  1  0  1  1  0
0  0  0  1  1  1  1

Контрольная матрица ----->
a1 a2 a3 a4 p1 p2 p3
0  1  1  1  1  0  0
1  0  1  1  0  1  0
1  1  0  1  0  0  1

Линейный код для ----> 1001 -----> 1001100
Искаженное слово ----> 0001100

Система проверок:
p1 ⊕ a2 ⊕ a3 ⊕ a4 = 1 ⊕ 0 ⊕ 0 ⊕ 1 = 0
p2 ⊕ a1 ⊕ a3 ⊕ a4 = 0 ⊕ 0 ⊕ 0 ⊕ 1 = 1
p3 ⊕ a1 ⊕ a2 ⊕ a4 = 0 ⊕ 0 ⊕ 0 ⊕ 1 = 1

Синдром ----> 011
Инвертируемый разряд = 1

Process finished with exit code 0

```