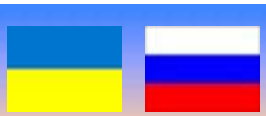
DonNTU
Computer
Department



КОНТРОЛЬНАЯ РАБОТА

по курсу

”ТЕОРІЯ КОРЕКТУЮЧИХ КОДІВ”

DonNTU
Computer
Department

TEL. (0622) 301-07-58
301-08-04
FAX. (062) 335-45-89
mailto:do@cs.dgtu.donetsk.ua

83000 Донецьк
вул. Артема 58
корпус 4, ауд. 4.14
кафедра “Комп’ютерна інженерія”
Дяченко О.М.

Web design by Dyachenko Oleg

УДК 681.3

Методичні вказівки щодо організації самостійної роботи студентів при виконанні індивідуальних завдань з курсу “Теорія коректуючих кодів” (для студентів спеціальності 7.091502 “Системне програмування”)/ Скл.: О.М.Дяченко - Донецьк: ДонНТУ, 2011. – 38 с.(на електронному носії № 208, прот. № 2 від 21.03.11)

Розглядаються питання розробки групового коду, коду Хеммінга, циклічного коду Хеммінга, а також проектування кодерів і декодерів на основі цих кодів. Наведені порядок і приклади виконання завдань контрольної роботи.

Укладач: О.М.Дяченко

Рецензент: Ю.Є.Зінченко

1. СИСТЕМАТИЧНІ ГРУПОВІ КОДИ

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Широкий клас коригувальних кодів складають систематичні групові коди. Ці коди відносяться до групи роздільних блокових кодів. Для систематичного групового коду сума по модулю два двох дозволених комбінацій також дає дозволену комбінацію.

У теорії кодування широко використовується матричне представлення кодів.

Усі дозволені кодові комбінації систематичного групового (n,k)-коду можна одержати, маючи k вихідних дозволених кодових слів.

Вихідні кодові комбінації повинні задовольняти наступним умовам:

1. У число вихідних комбінацій не повинна входити нульова.

2. Кодова відстань між будь-якими парами вихідних комбінацій не повинна бути менше d_{min}.

3. Кожна вихідна комбінація, як і будь-яка ненульова дозволена комбінація, повинна містити кількість одиниць не менш d_{min}.

4. Усі вихідні комбінації повинні бути лінійно незалежні, тобто жодна з них не може бути отримана шляхом підсумовування інших.

Вихідні комбінації можуть бути отримані з матриці, що має k рядків і n стовпців:

$$P(n,k)=\begin{vmatrix} a11 & a12 & \dots & a1k & b11 & b12 & \dots & b1p \\ a21 & a22 & \dots & a2k & b21 & b22 & \dots & b2p \\ . & . & \dots & . & . & . & \dots & . \\ . & . & \dots & . & . & . & \dots & . \\ . & . & \dots & . & . & . & \dots & . \\ ak1 & ak2 & \dots & akk & bk1 & bk2 & \dots & bkp \end{vmatrix}$$

Тут символи перших k стовпців є інформаційними й останні p стовпців - перевірочними.

Матрицю $P_{(n,\,k)}$ називають утворюючою.

Матриця $P_{(n,\,k)}$ може бути представлена двома підматрицями: інформаційною U_k і перевірочною H_p:

$$P(n,k)=\left|\begin{array}{cccc} a11 & a12 & \dots & a1k \\ a21 & a22 & \dots & a2k \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ ak1 & ak2 & \dots & akk \end{array}\right| \left|\begin{array}{cccc} b11 & b12 & \dots & b1p \\ b21 & b22 & \dots & b2p \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ bk1 & bk2 & \dots & bkp \end{array}\right|,$$

де

$$Uk = \begin{vmatrix} a11 & a12 & \dots & a1k \\ a21 & a22 & \dots & a2k \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ ak1 & ak2 & \dots & akk \end{vmatrix}; \quad Hp = \begin{vmatrix} b11 & b12 & \dots & b1p \\ b21 & b22 & \dots & b2p \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ bk1 & bk2 & \dots & bkp \end{vmatrix}.$$

Для побудови утворюючої матриці зручно інформаційну матрицю U_k брати у виді квадратної одиничної матриці:

$$Uk = \begin{vmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 0 & 1 \end{vmatrix}.$$

При цьому перевірочна підматриця повинна задовольняти наступним умовам:

1. Кількість одиниць у рядку повинна бути не менш d_{min} - 1.

2. Сума по модулю два двох будь-які рядків повинна містити не менш d_{min} - 2 одиниць.

Для кодів з $d_{min} = 2$ утворююча матриця має вид:

$$P(n,k) = \begin{vmatrix} 1 & 0 & 0 & \dots & 0 & 1 \\ 0 & 1 & 0 & \dots & 0 & 1 \\ 0 & 0 & 1 & \dots & 0 & 1 \\ . & . & . & \dots & . & . \\ 0 & 0 & 0 & \dots & 1 & 1 \end{vmatrix} = \underbrace{\begin{vmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ . & . & . & \dots & . \\ 0 & 0 & 0 & \dots & 1 \end{vmatrix}}_{Uk} \underbrace{\begin{vmatrix} 1 \\ 1 \\ 1 \\ . \\ 1 \end{vmatrix}}_{Hp}$$

В усіх комбінаціях коду, побудованого за допомогою такої матриці, парне число одиниць.

Для кодів з $d_{min} = > 3$ параметри утворюючої матриці визначаються виходячи з кількості інформаційних розрядів і заданих коригувальних здібностей коду. Порядок побудови і вибір параметрів утворюючої матриці для $d_{min} = > 3$ розглянемо на прикладі.

Приклад.

Побудувати матрицю для групового коду, здатного виправляти одиночну помилку при передачі 16 символів первинного алфавіту.

Рішення.

1. Тому що число інформаційних розрядів коду k = 4 (16=2^4), то число рядків утворюючої матриці дорівнює 4.

2. Число коригувальних розрядів для кодів з $d_{min} = 3$ дорівнює

$p = [\log_2 \{(k + 1) + [\log_2 (k + 1)]\}] = \log_2 8 = 3,$

тоді довжина коду $n = p + k = 3 + 4 = 7$, і загальне число стовпців дорівнює 7.

3. Як інформаційну підматрицю U_k вибираємо одиничну, тому, оскільки вага кожного рядка перевірочної підматриці повинниа бути не менш $d_{min} - 1 = 3 - 1 = 2$, то як рядки перевірочної підматриці H_p можуть бути обрані тризначні двійкові комбінації з числом одиниць, більшим чи рівним двом: 011, 101, 110, 111.

4. Остаточний вид $P_{(n,\ k)}$ матриці може бути одним з трьох варіантів:

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{vmatrix};$$

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{vmatrix};$$

$$\begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Як видно з приклада, основним вимогам можуть задовольняти кілька матриць. Вибір тієї чи іншої матриці з числа матриць, можливих для даних k і d_{min} визначається по додаткових вимогах: мінімум коригувальних розрядів чи максимальна простота апаратури кодера і декодера.

Коригувальні коди з мінімальною кількістю надлишкових розрядів називають щільноупакованими кодами.

Приклади щільноупакованих кодів для $d_{min} = 3$: (3, 1); (7, 4); (15, 11); (31, 26); (63, 57) і т.д.

Максимальна простота апаратури кодера і декодера досягається в нещільно упакованих кодах з малою щільністю перевірок на парність. Такі коди містять мінімальне число одиниць у коригувальних (перевірочних) розрядах породжуючої матриці, (тобто в перевірочній матриці). При побудові кодів з максимально простими шифраторами і дешифраторами для забезпечення умови $d_{min} = 3$ послідовно вибираються вектори вагою W = 2, 3, ... p.

Приклад.

Побудувати код для k = 20, $d_{min} = 3$, за умови максимальної простоти апаратури кодера і декодера.

Рішення.

$$p = [\log_2 \{(k + 1) + [\log_2 (k + 1)]\}] = 5.$$

Мінімальна кількість коригувальних розрядів, що забезпечують $d_{min} = 3$, p = 5.

Утворююча матриця коду складається з одиничної інформаційної підматриці з 20 рядків і стовпців і перевірочної підматриці. Мінімальна вага рядка перевірочної підматриці дорівнює $d_{min} - 1 = 2$. У перевірочної підматриці повинно бути 20 рядків і мінімальна кількість одиниць.

Якщо вибрати p = 5, одержимо 10 рядків з вагою 2 і 10 рядків з вагою 3, тобто 10 * 2 + 10 * 3 = 50 одиниць.

Якщо вибрати p = 6, одержимо 15 рядків з вагою 2 і 5 рядків з вагою 3, тобто 15 * 2 + 5 * 3 = 45 одиниць.

Якщо вибрати p = 7, одержимо 20 рядків з вагою 2, тобто 20 * 2 = 40 одиниць.

Таким чином, код з k = 20 і p = 7 має максимальну простоту апаратури кодера і декодера. Перевірочна підматриця буде складатися з 20 рядків наступних комбінацій: 0000011, 0000101, 0000110, 0001001, 0001010, ..., 1001000, 1010000.

Перевірочні символи утворюються за рахунок лінійних операцій над інформаційними символами. Для кожної кодової комбінації повинне бути складено p незалежних сум по модулю два.

Дуже зручно перевірочні суми складати за допомогою перевірочної матриці H, що будується в такий спосіб. На початку будується підматриця H1, що є транспонованою стосовно підматриці H_p:

$$H1 = \begin{vmatrix} b11 & b21 & \dots & bk1 \\ b12 & b22 & \dots & bk2 \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ b1p & b2p & \dots & bkp \end{vmatrix}.$$

Потім до неї праворуч приписується одинична матриця:

$$H = \begin{vmatrix} b11 & b21 & \dots & bk1 & 1 & 0 & 0 & \dots & 0 & 0 \\ b12 & b22 & \dots & bk2 & 0 & 1 & 0 & \dots & 0 & 0 \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \cdot & \dots & \cdot & \cdot \\ b1p & b2p & \dots & bkp & 0 & 0 & 0 & \dots & 0 & 1 \end{vmatrix}.$$

Алгоритм визначення перевірочних символів по інформаційним за допомогою матриці H наступний. Позиції, займані одиницями в першому рядку підматриці H1, визначають інформаційні розряди, що повинні брати участь у формуванні першого перевірочного розряду кодової комбінації. Позиції одиниць у другому рядку підматриці H1 визначають інформаційні розряди, що беруть участь у формуванні другого перевірочного розряду і т.д.

Приклад. Нехай утворююча матриця коду (7,4) має вид:

$$P7,4 = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Перевірочна підматриця:

$$Hp = \begin{vmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{vmatrix}.$$

Транспонована підматриця стосовно H_p:

$$H1 = \begin{vmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \end{vmatrix}.$$

Приписуючи праворуч одиничну матрицю, одержимо перевірочну матрицю:

$$H = \begin{vmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{vmatrix}.$$

Кодові комбінації повинні містити p = n - k = 7 - 4 = 3 перевірочних символів. Підматриця H1 указує на те, що перевірочні символи повинні визначатися рівностями:

```
b1 = a2 + a3 + a4;
b2 = a1 + a3 + a4;
b3 = a1 + a2 + a4.
```

Тоді для повідомлення, що представляється, наприклад, простою кодовою комбінацією 0011, перевірочні символи будуть:

```
b1 = 0 + 1 + 1 = 0;
b2 = 0 + 1 + 1 = 0;
b3 = 0 + 0 + 1 = 1.
```

Отже, повна кодова комбінація буде мати вид 0011001.

Перевірочна матриця H дуже зручна для визначення місця помилки в кодовій комбінації, а, отже, виправлення помилок. Перевірка кодових комбінацій при цьому виконується шляхом підсумовування по модулю два перевірочних символів кодових комбінацій і перевірочних символів, обчислених по прийнятим інформаційним. У результаті буде отримана сукупність контрольних рівностей, кожна з якої представляє суму по модулю два одного з контрольних розрядів і визначеної кількості інформаційних.

Склад контрольних рівностей легко визначається з перевірочної матриці H. До складу першої контрольної рівності повинні входити символи, позиції яких зайняті одиницями в першому рядку матриці H. До складу другої контрольної рівності повинні входити символи, позиції яких зайняті одиницями в другому рядку матриці H і т.д.

Так, для розглянутого раніше приклада з кодом (7, 4) ці рівності будуть мати вид:

```
S1 = b1 + a2 + a3 + a4;
S2 = b2 + a1 + a3 + a4;
S3 = b3 + a1 + a2 + a4.
```

У результаті p таких перевірок буде отримане p - розрядне двійкове число S (синдром), що буде дорівнювати нулю при відсутності помилок і відмінне від нуля у випадку наявності помилок (визначеної кратності!).

Якщо код призначений для виправлення помилок, то повинна бути заздалегідь визначена відповідність між видом синдрому і видом помилки, що виправляється.

Нехай у розглянутому прикладі (7, 4) з перевірочною матрицею H

$$H = \left|\begin{array}{cccccccc} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{array}\right|$$

відбулася помилка в першому розряді кодової комбінації (перекручений символ a1). Тоді перевірочні операції дадуть наступний результат: S1=0; S2=1; S3=1. Таким чином, буде отриманий синдром S = 011, що відповідає першому стовпцю матриці H.

При помилці в другому розряді a2 кодової комбінації буде отриманий синдром S = 101, що відповідає другому стовпцю матриці H і т.д.

При помилці в шостому розряді кодової комбінації, що відповідає перекручуванню другого перевірочного символу b2, буде отриманий синдром S = 010, що відповідає шостому стовпцю матриці H чи другому стовпцю одиничної підматриці.

Якщо для виправлення однократних помилок у кодових комбінаціях одержати синдроми досить просто, то для виправлення дворазових, триразових і т.д. помилок, а також для виправлення пачок помилок побудова синдромів досить важка й у цих випадках удаються до допомоги ЕОМ.

Завдання № 1

ПРОЕКТУВАННЯ КОДЕРА І ДЕКОДЕРА ГРУПОВОГО КОДУ

Завдання: розробити кодер і декодер для групового коду, що виправляє одиночну помилку.

Варіанти завдання:

- 1. Довжина кодового слова n

$$n = [(35 - N) / 2],$$

де квадратні дужки означають округлення до найближчого більшого цілого, N – номер варіанта.

- 2. Варіант А - груповий код оптимальний з погляду мінімуму коригувальних розрядів (N - парне). Варіант В - груповий код оптимальний з погляду мінімуму апаратних витрат реалізації кодера й декодера (N - непарне).

Порядок виконання роботи

- 1. Визначити мінімальну кількість перевірочних розрядів. Побудувати породжувальну матрицю групового коду за варіантом А або В.
- 2. Побудувати перевірочну матрицю групового коду. Визначити рівності для перевірочних розрядів і рівності для визначення розрядів синдрому.
- 3. Синтезувати кодер і декодер. Для виправлення одиночної помилки в декодері синтезувати дешифратор.
- 4. Розробити функціональні схеми кодера й декодера.

Зміст звіту

- 1. Титульний аркуш.
- 2. Завдання.
- 3. Вихідні дані.
- 4. Породжувальна матриця групового коду.
- 5. Перевірочна матриця групового коду.
- 6. Синтез декодера.
- 7. Функціональна схема кодера і декодера.

Контрольні питання

- 1. У чому полягає відмінність між блоковими й безперервними кодами?
- 2. Як визначається відстань між кодовими комбінаціями?
- 3. Який зв'язок коригувальної здатності з кодовою відстанню?
- 4. Як будується породжувальна матриця групового коду?
- 5. Які умови побудови перевірочної підматриці?
- 6. Який алгоритм визначення перевірочних символів за допомогою перевірочної матриці?
- 7. Як визначається склад перевірочних рівностей за допомогою перевірочної матриці?

Контрольний приклад (N=35)

Завдання: розробити кодер і декодер для групового коду, що виправляє одиночну помилку.

Вихідні дані

Кількість інформаційних символів коду: k = [(N+5)/2] = 20.
Варіант В - мінімальні апаратні витрати кодера й декодера.

Побудова коду

- Визначення мінімальної кількості контрольних розрядів.

  k = 20,
  $p \geq [\log_2\{(k+1)+[\log_2(k+1)]\}] = 5$.

  Для коду з мінімальними апаратними витратами кодера й декодера кожний рядок перевірочної підматриці повинен містити 2 одиниці (для коду, що виправляє одиночну помилку), причому всі рядки повинні бути різними.

  Якщо вибрати p=5, одержимо 10 рядків перевірочної підматриці з вагою 2 і 10 рядків з вагою 3.

  Якщо вибрати p=7, одержимо 20 рядків з вагою 2, тобто 20*2=40 одиниць у перевірочній підматриці.

- Побудова породжувальної матриці $P_{(27, 20)}$

$$P_{(27,20)}=\begin{vmatrix}
100\ 000\ 000\ 0\ 000\ 000\ 000\ 0\ 000\ 001\ 1\\
010\ 000\ 000\ 0\ 000\ 000\ 000\ 0\ 000\ 010\ 1\\
001\ 000\ 000\ 0\ 000\ 000\ 000\ 0\ 000\ 100\ 1\\
000\ 100\ 000\ 0\ 000\ 000\ 000\ 0\ 001\ 000\ 1\\
000\ 010\ 000\ 0\ 000\ 000\ 000\ 0\ 010\ 000\ 1\\
000\ 001\ 000\ 0\ 000\ 000\ 000\ 0\ 100\ 000\ 1\\
000\ 000\ 100\ 0\ 000\ 000\ 000\ 0\ 000\ 011\ 0\\
000\ 000\ 010\ 0\ 000\ 000\ 000\ 0\ 000\ 101\ 0\\
000\ 000\ 001\ 0\ 000\ 000\ 000\ 0\ 001\ 001\ 0\\
000\ 000\ 000\ 1\ 000\ 000\ 000\ 0\ 010\ 001\ 0\\
000\ 000\ 000\ 0\ 100\ 000\ 000\ 0\ 100\ 001\ 0\\
000\ 000\ 000\ 0\ 010\ 000\ 000\ 0\ 000\ 110\ 0\\
000\ 000\ 000\ 0\ 001\ 000\ 000\ 0\ 001\ 010\ 0\\
000\ 000\ 000\ 0\ 000\ 100\ 000\ 0\ 010\ 010\ 0\\
000\ 000\ 000\ 0\ 000\ 010\ 000\ 0\ 100\ 010\ 0\\
000\ 000\ 000\ 0\ 000\ 001\ 000\ 0\ 001\ 100\ 0\\
000\ 000\ 000\ 0\ 000\ 000\ 100\ 0\ 010\ 100\ 0\\
000\ 000\ 000\ 0\ 000\ 000\ 010\ 0\ 100\ 100\ 0\\
000\ 000\ 000\ 0\ 000\ 000\ 001\ 0\ 011\ 000\ 0\\
000\ 000\ 000\ 0\ 000\ 000\ 000\ 1\ 101\ 000\ 0
\end{vmatrix}$$

- Побудова перевірочної матриці

$$H=\left|\begin{array}{ccccccccccccccccccccccc}
a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} & a_{20} & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{array}\right|$$

- Визначення рівностей для перевірочних розрядів:

$b_1 = a_6 + a_{11} + a_{15} + a_{18} + a_{20}$
$b_2 = a_5 + a_{10} + a_{14} + a_{17} + a_{19}$
$b_3 = a_4 + a_9 + a_{13} + a_{16} + a_{19} + a_{20}$
$b_4 = a_3 + a_8 + a_{12} + a_{16} + a_{17} + a_{18}$
$b_5 = a_2 + a_7 + a_{12} + a_{13} + a_{14} + a_{15}$
$b_6 = a_1 + a_7 + a_8 + a_9 + a_{10} + a_{11}$
$b_7 = a_1 + a_3 + a_2 + a_4 + a_5 + a_6$

- Визначення рівностей для розрядів синдрому:

$s_1 = b_1 + a_6 + a_{11} + a_{15} + a_{18} + a_{20}$
$s_2 = b_2 + a_5 + a_{10} + a_{14} + a_{17} + a_{19}$
$s_3 = b_3 + a_4 + a_9 + a_{13} + a_{16} + a_{19} + a_{20}$
$s_4 = b_4 + a_3 + a_8 + a_{12} + a_{16} + a_{17} + a_{18}$
$s_5 = b_5 + a_2 + a_7 + a_{12} + a_{13} + a_{14} + a_{15}$
$s_6 = b_6 + a_1 + a_7 + a_8 + a_9 + a_{10} + a_{11}$
$s_7 = b_7 + a_1 + a_3 + a_2 + a_4 + a_5 + a_6$

Принципова схема кодера й декодера

У форматі PDF

Позначення сигналів
in1, in2, in3, in4, і 5-i20 - входи кодера (інформаційні символи);
out1, out2, out3, out4, о 5-о20 - виходи декодера;
ier1, ier2 - входи імітації помилок (імітуються 4 ситуації: без помилок, два варіанта одиночних помилок, що виправляються, подвійна помилка).

2. КОДИ ХЕММІНГА

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Код Хеммінга являє собою один з найважливіших класів лінійних кодів, що знайшли широке застосування на практиці і що має простий і зручний для технічної реалізації алгоритм виявлення і виправлення одиночної помилки.

До цих кодів звичайно відносять коди з виправленням одиночних помилок і коди з виправленням одиночних і виявленням дворазових помилок.

Код Хеммінга, що забезпечує виправлення всіх одиночних помилок, повинний мати мінімальну кодову відстань $d_{min} = 3$.

$$n = k + p,$$

де k - кількість інформаційних символів;

p - кількість перевірочних символів;

n - загальна кількість символів у коді.

При передачі коду може бути перекручений будь-який символ (n комбінацій). Однак може бути і такий випадок, коли жоден із символів не перекручений. Тоді за допомогою p контрольних символів повинна бути створена така кількість комбінацій, щоб розрізнити $n + 1$ варіант.

Тому

$$2^p => n + 1.$$

Разом з тим $2^n = 2^{(k+p)} = 2^k * 2^p$, чи

$$2^n = 2^k * 2^p => (n + 1)2^k.$$

Звідси число інформаційних символів коду, що коректує одиночну помилку

$$2^k <= 2^n / (n + 1).$$

Критерій оптимальності таких кодів - близькість до нижньої границі Хеммінга

$$2^p = 2^{(n - k)} => \sum_{i=1}^{r} C_n^i ,$$

де r - кратність помилки.

Для $r = 1 \quad 2^{(n - k)} - 1 = n$.

Це вираження є нижньою границею тому, що вона встановлює те мінімальне співвідношення коригувальних і інформаційних розрядів, нижче якого код не може зберігати задані коригувальні здібності.

Коди, оптимальні по цій умові називаються щільноупакованими.

Код Хеммінга є щільноупакованим.

Код будується таким чином, щоб у результаті p = n - k перевірок одержати p - розрядне двійкове число, що вказує номер перекрученої позиції кодової комбінації. Для цього перевірочні символи повинні знаходитися на номерах позицій, що виражаються степенем двійки (2^0, 2^1, 2^2, ... , $2^{(p-1)}$), тому що кожний з них входить тільки в одне з перевірочних рівнянь. Таким чином, якщо нумерувати позиції зліва-праворуч, то контрольні символи повинні знаходитися на 1-й, 2-й, 4-й, 16-й і т.д. позиціях.

Результат першої перевірки дає цифру молодшого розряду синдрому в двійковому записі. Якщо результат перевірки дасть 1 , то один із символів перевіреної групи перекручений. Таким чином, першою перевіркою повинні бути охоплені символи з номерами, що містять у двійковому записі одиниці в першому розряді: 1, 3, 5, 7, 9 і т.д. Результат другої перевірки дає цифру другого розряду синдрому. Отже, другою перевіркою повинні бути охоплені символи з номерами, що містять у двійковому записі одиниці в другому розряді: 2, 3, 6, 7, 10 і т.д.

Аналогічно при третій перевірці повинні перевірятися символи, номери яких у двійковому записі містять одиниці в третьому розряді: 4, 5, 6, 7, 12 і т.д.

Таким чином, перевірочні рівності повинні мати вид:

a1 = a3 + a5 + a7 + a9 + ...
a2 = a3 + a6 + a7 + a10 + ...
a4 = a5 + a6 + a7 + a12 + ...
a8 = a9 + a10 + a11 + a12 + ...
. . .

Перевірочна матриця коду повинна мати n стовпців і p рядків. Кожен стовпець повинний складати двійкову комбінацію, що вказує номер відповідної позиції коду.

Наприклад, для коду довжиною n = 9, що забезпечує виправлення однократних помилок, кількість надлишкових символів p = 4. При цьому перевірочною матрицею може бути обрана наступна матриця:

$$H = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{vmatrix}$$

Представимо як приклад просту двійкову комбінацію 10011 кодом Хеммінга. Тому що перевірочні символи повинні займати 1-у, 2-у, 4-у і 8-у позиції, то інформаційні - 3-ю, 5-у, 6-у, 7-у і 9-у позиції. Тоді a3 = 1, a5 = 0, a6 = 0, a7 = 1, a9 = 1. З умови забезпечення парності сум b1 , ... , b4 одержимо наступні значення перевірочних

символів: a1 = 1; a2 = 0; a4 = 1; a8 = 1. Отже, простому 5-розрядному коду 10011 відповідає 9-розрядний код Хеммінга 101100111.

Нехай тепер при передачі відбулося перекручування п'ятого символу, тобто код прийняв вид 101110111. Тоді b1 = 1, b2 = 0, b3 = 1, b4 = 0. Таким чином, у результаті перевірки одержуємо синдром 0101, що вказує на перекручування п'ятого символу. Виправлення помилки зводиться до інвертування символу на п'ятій позиції. Тому що номер перекрученого символу в двійковому записі збігається зі значенням синдрому, у декодері для виправлення помилки зручно використовувати стандартний дешифратор.

КОД ХЕММІНГА З ВИПРАВЛЕННЯМ ОДИНОЧНОЇ І ВИЯВЛЕННЯМ ДВОРАЗОВОЇ ПОМИЛОК.

Код Хеммінга з кодовою відстанню $d_{min} = 4$ виходить шляхом додавання до коду Хеммінга з $d_{min} = 3$ перевірочного символу, що представляє собою результат підсумовування по модулю два всіх символів кодової комбінації. Перевірочна матриця для коду з n = 9 може мати вид:

$$H = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{vmatrix}.$$

Додаткове перевірочне співвідношення, що вводиться для збільшення мінімальної відстані коду Хеммінга, має вид:

b5 = a1 + a2 + a3 + a4 + a5 + a6 + a7 + a8 + a9 + a10.

Операція декодування складається з двох етапів. На першому визначається синдром, що відповідає коду з $d_{min} = 3$, на другому – перевіряється останнє перевірочне співвідношення.

Тоді у випадку однієї помилки синдром укаже номер помилкової позиції, а перевірка на парність - на наявність помилки (непарної). Якщо синдром указує на наявність помилки, а перевірка на парність не фіксує її, значить у кодовій комбінації дві помилки (виправлення - неможливе).

Розрядність запам’ятовуючих пристроїв, як правило, кратна байту. Тому при використанні в запам’ятовуючих пристроях коригувальних кодів приходиться вирішувати задачу одержання укорочених кодів із заданою мінімальною кодовою відстанню d = 4, H - матриці для неукороченого коду з найменшим значенням p, з якої видаляються зайві стовпці. При цьому з вихідної H - матриці доцільно виключати стовпці з максимальною кількістю одиниць для того, щоб зменшити число доданків для одержання контрольних

розрядів і розрядів синдрому і спростити пристрої кодування і декодування. Стовпці матриці, що містять по одній одиниці, відповідають контрольним розрядам і виключенню не підлягають.

Надмірність коду Хеммінга p/n залежить від кількості інформаційних символів і при зміні k від 4 до 1013 змінюється від 0.429 до 0.098 при $d_{min} = 3$ і від 0.5 до 0.0107 при $d_{min} = 4$.

Завдання № 2

ПРОЕКТУВАННЯ КОДЕРА І ДЕКОДЕРА КОДУ ХЕММІНГА

Завдання: розробити кодер і декодер для кодів Хеммінга, що виправляють одиночну помилку й виправляють одиночну і виявляють двократні помилки.

Варіанти завдання:

- 1. Кількість k інформаційних розрядів: k = [(N+5) / 2],
  де квадратні дужки означають округлення до найближчого більшого цілого, N – номер варіанта.
- 2. Варіант А - код Хеммінга з виправленням одиночної помилки й виявленням двократної (N - непарне). Варіант В - код Хеммінга з виправленням одиночної помилки (N - парне).

Порядок виконання роботи

- 1. Визначити мінімальну кількість контрольних розрядів.
- 2. Побудувати перевірочну матрицю коду Хеммінга за варіантом А або В. Визначити рівності для перевірочних розрядів і рівності для визначення розрядів синдрому.
- 3. Синтезувати кодер і декодер. Для виправлення одиночної помилки в декодері використати стандартний дешифратор.
- 4. Розробити функціональні схеми кодера і декодера.

Зміст звіту

- 1. Титульний аркуш.
- 2. Завдання.
- 3. Вихідні дані.
- 4. Перевірочна матриця коду Хеммінга.
- 5. Функціональна схема кодера і декодера.

Контрольні питання

- 1. Що обумовило широке розповсюдження двійкових кодів?
- 2. Який принцип побудови кодів Хеммінга?
- 3. Як складаються перевірочні рівності коду Хеммінга?
- 4. Як будується перевірочна матриця для коду Хеммінга з виправленням одиночної помилки?
- 5. Як будується перевірочна матриця для коду Хеммінга з виправленням одиночної й виявленням двократної помилок?
- 6. Як визначається коефіцієнт надлишковості коду?
- 7. Як визначаються номери позицій контрольних розрядів у коді Хеммінга?
- 8. Які існують різновиди кодів Хеммінга? У чому їхня відмінність?

Контрольний приклад (N=-5)

Завдання: розробити кодер і декодер для кодів Хеммінга, що виправляють одиночну помилку й виправляють одиночну і виявляють двократні помилки.

Вихідні дані

Довжина коду: n = [(35-N)/2] = 20.
Варіант В - код Хеммінга, що виправляє одиночну помилку (d_{min}=3).

Побудова коду

- Визначення мінімальної кількості контрольних символів.

  Довжина коду n = 20.

  Мінімальна кількість перевірочних символів p = [$\log_2$ (n+1)] = 5.
  Кількість інформаційних символів k = n-p = 20-5 = 15.

- Побудова перевірочної матриці

  Контрольні розряди : b_1, b_2, b_4, b_8, b_{16} . Позиції перевірочних розрядів: 1-а, 2-а, 4-а, 8-а, 16-а.
  Перевірочна матриця:

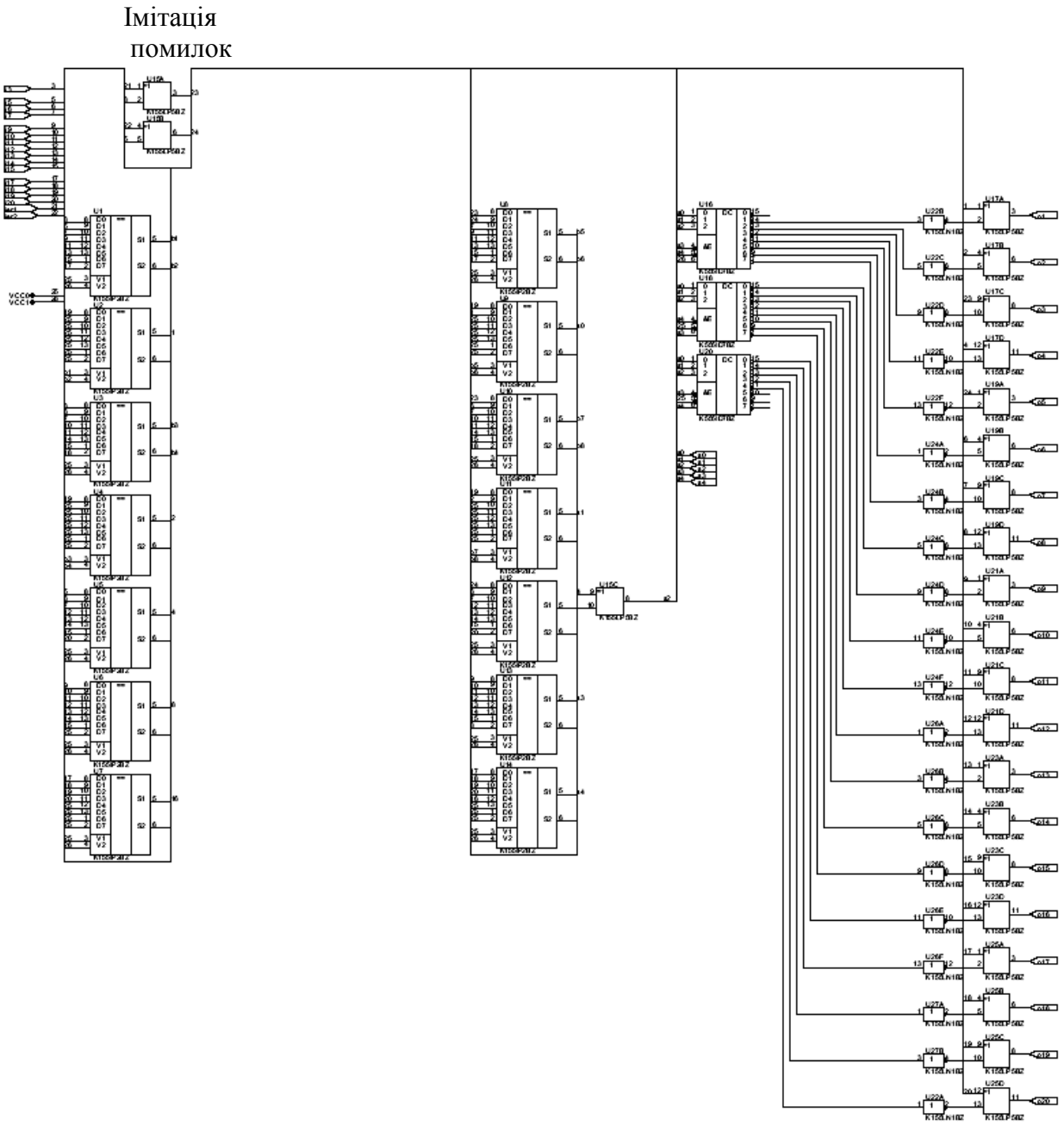
$$H = \begin{array}{c|cccccccccccccccccc} & b_1 & b_2 & a_3 & b_4 & a_5 & a_6 & a_7 & b_8 & a_9 & a_{10} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & b_{16} & a_{17} & a_{18} & a_{19} & a_{20} \\ \hline & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{array}$$

- Визначення рівностей для перевірочних розрядів:
  $b_1 = a_3 + a_5 + a_7 + a_9 + a_{11} + a_{13} + a_{15} + a_{17} + a_{19}$;
  $b_2 = a_3 + a_6 + a_7 + a_{10} + a_{11} + a_{14} + a_{15} + a_{18} + a_{19}$;
  $b_4 = a_5 + a_6 + a_7 + a_{12} + a_{13} + a_{14} + a_{15} + a_{20}$;
  $b_8 = a_9 + a_{10} + a_{11} + a_{12} + a_{13} + a_{14} + a_{15}$;
  $b_{16} = a_{17} + a_{18} + a_{19} + a_{20}$.
- Визначення рівностей для розрядів синдрому:

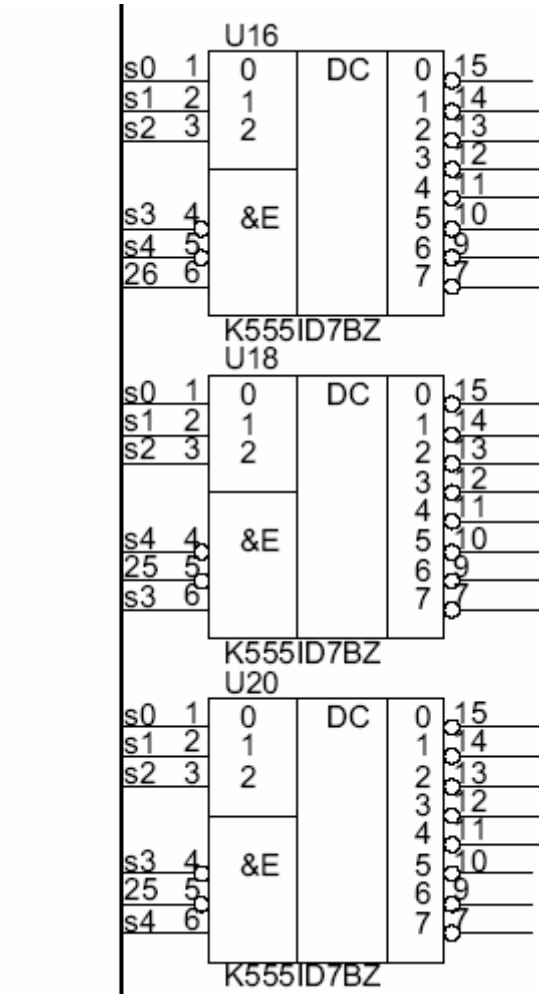
  $s_0 = b_1 + a_3 + a_5 + a_7 + a_9 + a_{11} + a_{13} + a_{15} + a_{17} + a_{19}$;
  $s_1 = b_2 + a_3 + a_6 + a_7 + a_{10} + a_{11} + a_{14} + a_{15} + a_{18} + a_{19}$;
  $s_2 = b_4 + a_5 + a_6 + a_7 + a_{12} + a_{13} + a_{14} + a_{15} + a_{20}$;
  $s_3 = b_8 + a_9 + a_{10} + a_{11} + a_{12} + a_{13} + a_{14} + a_{15}$;
  $s_4 = b_{16} + a_{17} + a_{18} + a_{19} + a_{20}$.

Принципова схема кодера й декодера
У форматі PDF

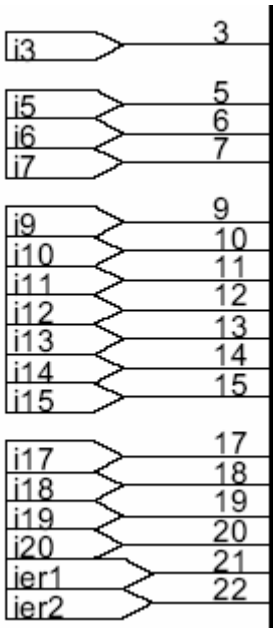
Схема кодера, імітації помилок і декодера (загальний вид).



Дешифратор декодера коду Хеммінга (25 - логічний 0, 26 - логічна 1).



Входи кодера коду Хеммінга.



Позначення сигналів

in3, in5, in6, і 3-i20 - входи кодера (інформаційні символи);
out3, out5, out6, о 3-о20 - виходи декодера;
ier1, ier2 - входи імітації помилок (імітуються 4 ситуації: без помилок, два варіанта одиночних помилок, що виправляються, двократна помилка):
s4-s0 - розряди синдрому.

3. ЦИКЛІЧНІ КОДИ

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Циклічні коди одержали досить широке застосування завдяки їх ефективності при виявленні і виправленні помилок. Схеми пристроїв, що кодують і декодують, для цих кодів надзвичайно прості і будуються на основі звичайних регістрів зрушення. Ці коди лінійні блокові.

Назва кодів відбулася від їхньої властивості, що полягає в тім, що кожна кодова комбінація може бути отримана шляхом циклічної перестановки символів комбінації, що належить до цього ж коду. Це значить, що якщо, наприклад, комбінація $a_0\ a_1\ a_2\ ...\ a_{n-1}$ є дозволеною комбінацією циклічного коду, то комбінація $a_{n-1}\ a_0\ a_1\ a_2\ ...\ a_{n-2}$ також належить цьому коду.

ПРЕДСТАВЛЕННЯ ДВІЙКОВОГО КОДУ У ВИГЛЯДІ ПОЛІНОМА

Циклічні коди зручно розглядати, представляючи комбінацію двійкового коду не у вигляді послідовностей нулів і одиниць, а у вигляді полінома від фіктивної перемінної X, а саме:

$$b(X) = a_{n-1} X^{n-1} + a_{n-2} X^{n-2} + ... + a_1 X + a_0,$$

де a_i - цифри даної системи числення (у двійковій системі 0 і 1), знак "+" - сума по модулю два.

Так, наприклад, двійкове 7-розрядне число 1110101 може бути записане у вигляді полінома:

$$b(X) = 1*X^6 + 1*X^5 + 1*X^4 + 0*X^3 + 1*X^2 + 0*X + 1 =$$
$$X^6 + X^5 + X^4 + X^2 + 1.$$

Найбільший степінь X в доданку з ненульовим коефіцієнтом називається степенем полінома.

У розглянутому прикладі степінь b(X) дорівнює deg b(X) = 6.

ОПЕРАЦІЇ НАД ПОЛІНОМАМИ

Представлення кодових комбінацій у вигляді полінома дозволяє звести дії над комбінаціями до дій над поліномами.

При цьому додавання двійкових поліномів зводиться до додавання по модулю два коефіцієнтів при рівних степенях перемінної X.

Приклад. $A(X) = X^5 + X^4 + X^3 + 1;$

$B(X) = X^7 + X^4 + X^3 + X^2;$

$A(X) + B(X) = X^5 + X^4 + X^3 + 1 + X^7 + X^4 + X^3 + X^2 =$

$= X^7 + X^5 + X^2 + 1.$

Множення виконується за звичайним правилом перемножування степеневих функцій, однак, отримані при цьому коефіцієнти при рівних степенях перемінної X складаються по модулю два.

Приклад. $A(X) = X^5 + X^2 + 1;$

$B(X) = X^3 + X + 1;$

$A(X) * B(X) = (X^5 + X^2 + 1)*(X^3 + X +1) =$

$X^5 + X^2 + 1$

$*$

$X^3 + X + 1$

$X^5 + X^2 + 1$

$X^6 + X^3 + X$

$X^8 + X^5 + X^3$

$X^8 + X^6 + X^2 + X + 1$

Розподіл здійснюється за правилами розподілу степеневих функцій, при цьому операції віднімання заміняються операціями підсумовування по модулю два.

Приклад. $A(X) = X^5 + X^2 + 1;$

$B(X) = X^3 + X + 1;$

$X^5 + X^2 + 1$ | $X^3 + X + 1$

$+$ | ---

$X^5 + X^3 + X^2$ | $X^2 + 1$

$X^3 + 1$

$+$

$X^3 + X + 1$

X – залишок

$A(X)/B(X) = X^2 + 1 + X/(X^3 + X + 1).$

ПОБУДОВА ЦИКЛІЧНИХ КОДІВ

Ідея побудови циклічних кодів базується на використанні незвідних поліномів.

Незвідними називаються поліноми, що не можуть бути представлені у вигляді добутку поліномів нижчих степенів з коефіцієнтами з цього ж поля. Вони так само, як і прості числа, не можуть бути представлені добутком інших чисел. Іншими словами, незвідні поліноми поділяються без залишку тільки на себе чи на одиницю. Наприклад: $X^3 + X^2 + 1.$

Ідея корекції помилок у циклічних кодах базується на тім, що дозволені комбінації коду поділяються без залишку на деякий утворюючий поліном, що вибирається з числа незвідних поліномів.

Для виявлення помилки при розподілі на обраний (чи побудований за спеціальними правилами) поліном треба, щоб усі комбінації коду не поділялися ні на який інший поліном, а для цього необхідно, щоб обраний поліном не розкладався на інші поліноми (як, наприклад, прості числа натурального ряду не розкладаються на співмножники), тобто був незвідним поліномом. Такі поліноми варто шукати серед непарних поліномів, тобто серед поліномів, що містять непарне число одиниць, тому що з усіх парних поліномів легко виділити двочлен (X + 1), тобто парні поліноми складаються мінімум із двох співмножників, тобто не є незвідними.

Незвідні поліноми у теорії циклічних кодів відіграють роль утворюючих (генераторних, виробляючих), тому що коректуючі здібності циклічного коду визначаються цим незвідним поліномом.

СИСТЕМАТИЧНІ КОДИ

Нехай потрібно закодувати одну з комбінацій чотиризначного двійкового коду: $A(X) = X^3 + X^2 + 1$, тобто 1101. Поки, не обґрунтовуючи свій вибір, беремо з таблиці незвідних поліномів як утворюючий поліном $K(X) = X^3 + X + 1$, тобто 1011. Потім множимо A(X) на одночлен того ж степеня, що й утворюючий поліном. Від множення полінома на поліном X^p степінь кожного члена полінома підвищиться на p, що еквівалентно приписуванню p нулів з боку молодших розрядів полінома.

Тому що deg K(X) = 3, то інформаційна комбінація A(X) збільшується на X^3:

$A(X)*X^p = (X^3 + X^2 + 1)*X^3 = X^6 + X^5 + X^3 = 1101000$.

Ця процедура здійснюється для того, щоб у наслідку замість цих нулів можна було записати коригувальні розряди.

Значення коригувальних розрядів знаходять у результаті розподілу $A(X)*X^p$ на K(X):

$$
\begin{array}{l|l}
X^6 + X^5 + X^3 & X^3 + X + 1 \\
+ & \hline
X^6 + X^4 + X^3 & X^3 + X^2 + X + 1 \\
\hline
X^5 + X^4 & \\
+ & \\
X^5 + X^3 + X^2 & \\
\hline
\end{array}
$$

$$
\begin{array}{l}
X^4 + X^3 + X^2 \\
+ \\
X^4 + X^2 + X \\
\hline
X^3 + X \\
+ \\
X^3 + X + 1 \\
\hline
1
\end{array}
$$

У результаті розподілу $(A(X)*X^3)/K(X) = X^3 + X^2 + X + 1 + 1/(X^3 + X + 1)$, або в загальному виді $(A(X)*X^p)/K(X) = Q(X) + R(X)/K(X)$, де Q(X) - частка, а R(X) - залишок від розподілу A(X) на K(X).

Останнє вираження можна переписати в наступному виді:

$A(X)*X^p = Q(X)*K(X) + R(X)$, або

$F(X) = Q(X)*K(X) = A(X)*X^p + R(X)$.

Для розглянутого приклада $F(X) = (X^3 + X^2 + X + 1)(X^3 + X + 1) = (X^3 + X^2 + 1)*X^3 + 1$, або в двійковому представленні

$F(X) = 1111*1011 = 1101000 + 001 = 1101001$.

Поліном 1101001 і є шукана кодова комбінація, де 1101 - інформаційна частина, а 001 - контрольні символи.

Помітимо, що цей поліном поділяється на утворюючий поліном K(X) без залишку. Перевіримо це:

$$
\begin{array}{lll}
1101001 \rightarrow X^6 + X^5 + X^3 + 1 & \Big| & X^3 + X + 1 \\
+ & & \overline{\qquad\qquad} \\
\quad X^6 + X^4 + X^3 & & X^3 + X^2 + X + 1 \\
\quad \overline{\qquad\qquad\qquad} & & \\
\quad X^5 + X^4 + 1 & & \\
+ & & \\
\quad X^5 + X^3 + X^2 & & \\
\quad \overline{\qquad\qquad} & & \\
\qquad X^4 + X^3 + X^2 + 1 & & \\
+ & & \\
\qquad X^4 + X^2 + X & & \\
\qquad \overline{\qquad\qquad\qquad} & & \\
\qquad\quad X^3 + X + 1 & & \\
+ & & \\
\qquad\quad X^3 + X + 1 & & \\
\qquad\quad \overline{\qquad\qquad} & & \\
\qquad\qquad\quad 0 & &
\end{array}
$$

Залишки від розподілу поліномів є визначниками помилок циклічних кодів.

Таким чином, по залишку від розподілу кодової комбінації на утворюючий поліном судять про наявність у ній помилок: якщо залишок дорівнює нулю - помилок немає, якщо залишок не дорівнює нулю - помилки є.

Ми розглянули перший спосіб побудови циклічного коду: кодова комбінація циклічного (n, k) - коду виходить шляхом множення простої кодової комбінації степеня (k-1) на одночлен X^n-k і додавання до цього добутку залишку, отриманого від розподілу отриманого добутку на утворюючий поліном K(X) степеня (n-k).

При цьому способі кодування перші k символів отриманої кодової комбінації збігаються з відповідними символами вихідної простої кодової комбінації.

НЕСИСТЕМАТИЧНІ КОДИ

Нехай потрібно закодувати одну з комбінацій чотиризначного двійкового коду:

$A(X) = X^3 + X^2 + 1$, тобто 1101.

Як утворюючий поліном виберемо той же, що й у 1 способі побудови циклічного коду: $K(X) = X^3 + X + 1$.

Шукана кодова комбінація в цьому випадку визначається як добуток поліномів A(X) і K(X).

$$
\begin{array}{l}
\quad X^3 + X^2 + 1 \\
\quad * \\
\quad X^3 + X + 1 \\
\quad \overline{\qquad\qquad} \\
\quad X^3 + X^2 + 1 \\
+ \\
\quad X^4 + X^3 + X \\
+ \\
X^6 + X^5 + X^3 \\
\overline{\qquad\qquad\qquad\qquad\qquad\qquad} \\
X^6 + X^5 + X^4 + X^3 + X^2 + X + 1
\end{array}
$$

Отже, шукана кодова комбінація в даному прикладі має вигляд:

$F(X) = A(X)*K(X) = X^6 + X^5 + X^4 + X^3 + X^2 + X + 1$, або в двійковому представленні 1111111.

Таким чином, другий спосіб побудови циклічного коду припускає множення простої кодової комбінації степеня (k-1) на утворюючий поліном K(X) степеня (n-k).

Як і при першому способі, по залишку від розподілу кодової комбінації на утворюючий поліном судять про наявність у ній помилок: якщо залишок дорівнює нулю, помилок немає; у іншому випадку помилки мають місце.

Однак при другому способі в отриманій кодовій комбінації інформаційні символи не завжди збігаються з відповідними символами вихідної простої кодової комбінації, зокрема, у розглянутому прикладі перші k символів мають вид 1111, а вихідна комбінація - 1101.

Тому в декодері повинне бути передбачене одержання вихідних символів. Воно виконується шляхом розподілу кодової комбінації на утворюючий поліном.

Наприклад, у розглянутому нами прикладі вихідні символи A(X) визначаються шляхом розподілу F(X) на K(X):

$$
\begin{array}{l}
F(X) = X^6 + X^5 + X^4 + X^3 + X^2 + X + 1 \\
\quad + \\
\quad X^6 + X^4 + X^3 \\
\quad \underline{\qquad\qquad\qquad\qquad} \\
\qquad\quad X^5 + X^2 + X + 1 \\
\qquad + \\
\qquad\quad X^5 + X^3 + X^2 \\
\qquad\quad \underline{\qquad\qquad\qquad\qquad} \\
\qquad\qquad\quad X^3 + X + 1 \\
\qquad\qquad + \\
\qquad\qquad\quad X^3 + X + 1 \\
\qquad\qquad\quad \underline{\qquad\qquad\qquad} \\
\qquad\qquad\qquad\quad 0 = R(X)
\end{array}
\qquad
\begin{array}{|l}
X^3 + X + 1 = K(X) \\
\hline
X^3 + X^2 + 1 = A(X)
\end{array}
$$

Висновок: перший і другий способи побудови циклічних кодів мають різну апаратну реалізацію кодуючих і декодуючих пристроїв.

МАТРИЧНЕ ПРЕДСТАВЛЕННЯ ЦИКЛІЧНИХ КОДІВ

Для формування рядків утворюючої матриці при 1-му способу утворення циклічного коду беруть комбінації простого k-розрядного коду A(X) утримуючі одиницю в одному розряді. Ці комбінації збільшуються на X^{n-k}, визначається залишок R(X) від розподілу отриманого добутку $X^{n-k}*A(X)$ на утворюючий поліном і записується відповідний рядок матриці у виді суми добутку $X^{n-k}*A(X)$ і залишку R(X). При цьому утворююча матриця $P_{(n,\ k)}$ представляється двома підматрицями - інформаційної U_k і додаткової H_p:

$$P_{(n,\ k)} = |\ U_k, H_p\ |.$$

Інформаційна підматриця U_k являє собою квадратну одиничну матрицю з кількістю рядків і стовпців, рівним k. Додаткова підматриця H_p містить p = n - k стовпців і k рядків і утворена залишками R(X).

Утворююча матриця дозволяє одержати k комбінацій коду. Інші комбінації виходять підсумовуванням по модулю два рядків утворюючої матриці у всіх можливих сполученнях.

Нехай, наприклад, необхідно побудувати утворюючу матрицю (7, 4) циклічного коду. Утворюючий поліном $K(X) = X^3 + X^2 + 1$.

Інформаційна підматриця має вигляд:

$$Uk = \begin{vmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{vmatrix}.$$

Для одержання першого рядка додаткової підматриці перший рядок інформаційної підматриці збільшується на X^3 і поділяється на утворюючий поліном. Це відповідає виконанню операцій

0001*1000/1101. Залишок цих операцій 101 і складе перший рядок додаткової підматриці. Аналогічно визначаються інші рядки додаткової підматриці.

Остаточно утворююча матриця має вигляд

$$P_{7,4} = \begin{vmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \end{vmatrix}.$$

При 2-му способу утворення циклічного коду утворююча матриця $P_{(n, k)}$ формується шляхом множення утворюючого полінома K(X) степеня p = n - k на поліном X^{k-1} і наступних k-1 зрушень отриманої комбінації.

Інший варіант побудови утворюючої матриці $P_{(n, k)}$ - безпосереднє множення елементів одиничної підматриці на утворюючий поліном.

Незалежно від способу утворення циклічного коду і побудови утворюючої матриці, коректуючі властивості коду визначаються тільки обраним утворюючим поліномом K(X).

ВИБІР УТВОРЮЮЧОГО ПОЛІНОМА

При побудові циклічного коду спочатку визначається необхідне число інформаційних розрядів k. Потім знаходиться найменша довжина кодових комбінацій n, що забезпечує виявлення чи виправлення помилок заданої кратності. Ця задача зводиться до пошуку потрібного утворюючого полінома K(X).

Степінь утворюючого полінома повинна дорівнювати числу перевірочних розрядів p.

Оскільки в циклічному коді визначниками помилок є залишки від розподілу полінома прийнятої комбінації на утворюючий поліном, коректуюча здатність коду буде тим вище, чим більше залишків може бути утворене в результаті цього розподілу.

Найбільше число залишків, рівне $2^p - 1$ (крім нульового), може забезпечити тільки незвідний поліном степеня p (тобто не поділяється ні на який інший поліном). Це необхідна умова.

Достатня умова забезпечення найбільшого числа залишків: утворюючий поліном повинний бути примітивним.

Перший спосіб вибору утворюючого полінома:

- по заданому числу інформаційних розрядів k визначається число перевірочних розрядів p, необхідне для виправлення однократних помилок;

- по таблиці знаходиться примітивний поліном степеня p.

Другий спосіб вибору утворюючого полінома заснований на деяких властивостях поліномів.

По-перше, утворюючий поліном повинний бути дільником двочлена $(X^n + 1)$.

По-друге, відомо, що двочлен типу $(X^n + 1)$ дорівнює X в степені $2^{(z-1)} + 1$, у розкладанні якого, як співмножник, повинний входити утворюючий поліном, володіє тією властивістю, що він є загальним кратним для усіх без винятку незвідних поліномів степеня z і розкладається на множники з усіх незвідних поліномів степеня z, які поділяють без залишку число z.

Однак не всякий поліном степеня P, що входить у розкладання двочлена $X^n + 1$, може бути використаний як утворюючий поліном. Необхідно, щоб для кожної з однократних помилок забезпечувався свій, відмінний від інших, залишок від розподілу прийнятої кодової комбінації на утворюючий поліном. Це буде мати місце за умови, якщо обраний незвідний поліном степеня P, будучи дільником двочлена $X^n + 1$, не входить у розкладання ніякого іншого двочлена $X^i + 1$, степінь якого $i < n$.

У цьому випадку користаються таблицями незвідних поліномів.

Приклад. Вибрати утворюючий поліном для побудови циклічного коду, що містить $k = 4$ інформаційних символів й має можливість виправлення однократних і виявлення дворазових помилок.

$$p = n - k => \log_2 (n + 1).$$

$$p=[\log_2 \{(k + 1) + [\log_2 (k + 1)]\}] = [\log_2 (5 + [\log_2 5])]=3.$$

$$n = k + p = 7.$$

Утворюючий поліном K(X) повинний бути степеня p=3 і входити як співмножник у розкладання двочлена $X^n + 1 = X^{2\wedge(z-1)} + 1$.

Тому що $n = 7$, то складові співмножники двочлена повинні бути незвідними поліномами, степені яких є дільниками числа $z = 3$. До чисел, на які $z = 3$ поділяється без залишку, відносяться 1 і 3. Отже, співмножниками двочлена $X^7 + 1$ повинні бути незвідні поліноми першої і третьої степенів.

Користаючись таблицями незвідних поліномів, одержимо

$$X^7 + 1 = (X + 1)(X^3 + X + 1)(X^3 + X^2 + 1).$$

Жоден із співмножників степеня 3 не входить у розкладання іншого двочлена $X^n + 1$ степеня $n < 7$. Тому кожний з цих співмножників може бути обраний як утворюючий поліном.

ВИЗНАЧЕННЯ ДВОЇСТОГО ПОЛІНОМА

Поліном $P'(X) = X^{\deg P(X)} * P(X^{-1})$ називається двоїстим (зворотним) поліному $P(X)$.

Приклад. $P(X) = X^4 + X^3 + 1$.

$$P'(X) = X^4(X^{-4} + X^{-3} + 1) = X^4 + X + 1.$$

Двоїстий поліном незвідного полінома також незвідний, а двоїстий поліном примітивного полінома примітивний. Тому в таблицях приводиться або сам поліном, або двоїстий поліном.

Для того, щоб перевірити, чи є поліном $f(X)$ степеня m примітивним, застосовується наступна послідовність операцій:

1. Знаходяться відрахування $1, X, X^2, X^3,...,X^{2\^m}$ - 1 по модулю $f(X)$.

2. Ці відрахування збільшуються і приводяться по модулю $f(X)$ для того, щоб побудувати відрахування $X^{2\^m}$ - 1. Якщо результат відрізняється від 1, то поліном відкидається. Якщо результат дорівнює 1, то перевірка продовжується.

3. Для кожного співмножника r у розкладанні числа $2, m$ - 1 відрахування для X^r утвориться перемножуванням придатної комбінації відрахувань, знайдених на першому етапі. Якщо всі ці відрахування не рівні 1, то поліном є примітивним.

У таблицях поліноми дані у восьмеричному представленні. Наприклад, 3525 позначає поліном 10-го степеня: 011101010101 чи $X^{10} + X^9 + X^8 + X^6 + X^4 + X^2 + 1$.

Утворюючі поліноми кодів, здатних виправляти помилки будь-якої кратності, можна визначати, користаючись наступним правилом Хеммінга:

1. По заданому числу інформаційних розрядів k визначається число перевірочних розрядів p, необхідне для виправлення однократних помилок, і знаходиться утворюючий поліном.

2. Розглядаючи отриманий (n, k) - код як некоригувальний n - розрядний код, визначають додаткові розряди для забезпечення виправлення однієї помилки в цьому коді, і знаходять відповідний утворюючий поліном.

3. Повторюється дана процедура стільки разів, поки не буде отриманий код, що виправляє незалежні помилки до даної кратності включно.

Однак код, побудований таким чином, є неоптимальним з погляду числа надлишкових розрядів. У цьому відношенні більш досконалий код Боуза-Чоудхурі-Хоквінгема (БЧХ), що забезпечує мінімальне число перевірочних символів при заданому k.

Завдання № 3

ПРОЕКТУВАННЯ КОДЕРА Й ДЕКОДЕРА КОДУ ХЕММІНГА

Завдання : розробити кодер і декодер для циклічних кодів Хеммінга, що виправляють одиночну помилку.

Варіанти завдання:

1. Кількість k інформаційних розрядів коду Хеммінга

$$k = [\ (33-N)\ /\ 4\ ],$$

де N - номер варіанта.

2.

$$N \bmod 4 = \begin{cases} 0 - \text{варіант A}; \\ 1 - \text{варіант B}; \\ 2 - \text{варіант C}; \\ 3 - \text{варіант D}. \end{cases}$$

Для варіантів C і D вибирається породжувальний поліном K(X) з таблиці; для варіантів A і B - поліном K'(X), двоїстий поліному K(X) з таблиці.

Для варіантів A і C - несистематичний код; для варіантів B і D - систематичний код.

Таблиця 1 - Примітивні поліноми

Ступінь полінома	Поліном в 8-ричній системі числення
2	7
3	13
4	23
5	45
6	103

Приклад визначення полінома для 6-го степеня:
поліном в восьмеричній системі числення: 103 ;
поліном в двійковій системі числення: 001 000 011 ;
поліноміальна форма представлення: $X^6 + X + 1$.

Порядок виконання роботи

- 1. Визначити мінімальну кількість перевірочних розрядів. Вибрати породжувальний поліном з таблиці.
- 2. Відповідно до заданого варіанта побудувати породжувальну матрицю циклічного коду.
- 3. Синтезувати кодер і декодер на основі лінійних перемикальних схем.
- 4. Розробити функціональні й принципові схеми кодера й декодера.
- 5. Скласти й налагодити програмну модель.

- 6. Виконати моделювання на ЕОМ схеми, що імітує кодер, двійковий канал, декодер. У двійковому каналі передбачити можливість імітації помилок. Дослідити коригувальні здатності декодера.

Зміст звіту

- 1. Титульний аркуш.
- 2. Завдання.
- 3. Вихідні дані.
- 4. Визначення мінімальної кількості контрольних розрядів.
- 5. Вибір породжувального полінома.
- 6. Породжувальна матриця циклічного коду.
- 7. Функціональна схема кодера і декодера.

Контрольні питання

- 1. Чим обумовлена назва циклічних кодів?
- 2. Які відомі способи побудови циклічних кодів?
- 3. Як вибирається породжувальний поліном циклічного коду?
- 4. Як будується перевірочна матриця для циклічного коду з виправленням одиночної помилки?
- 5. Яка процедура виявлення й виправлення помилки в циклічних кодах з $d_{min}=3$?
- 6. Що таке "декодер Меггітта"?
- 7. Що таке "укорочений циклічний код"?
- 8. Як реалізується операція ділення на поліном за допомогою лінійної перемикальної схеми?
- 9. Як виконується множення поліномів за допомогою лінійної перемикальної схеми?
- 10. Як визначити поліном, двоїстий заданому?
- 11. Що таке "незвідний поліном"?

Контрольний приклад (N=35)

Завдання: розробити кодер і декодер для циклічних кодів Хеммінга, що виправляють одиночну помилку.

Вихідні дані

Кількість інформаційних символів k = [(N+1)/4] = 9.
Варіант С - систематичний циклічний код Хеммінга;
породжувальний поліном - K'(X), двоїстий табличному K(X).

Побудова коду

- 1. Визначення мінімальної кількості контрольних розрядів:
  $p = [\log_2\{(k+1)+[\log_2(k+1)]\}] = 4$
  (у цьому випадку квадратні дужки означають округлення до найближчого більшого цілого).
  $n = k + p = 9 + 4 = 13$.

- 2. Вибір породжувального полінома. У таблиці незвідних примітивних поліномів, поліном ступеня чотири (p=4, отже, deg K(X)=4) представлений у вигляді восьмеричного запису ненульових коефіцієнтів, дорівнює 23, тобто 10011, або в поліноміальній формі запису $K(X) = X^4 + X + 1$. ($23_8 = 010\ 011_2 = 0*X^5 + 1*X^4 + 0*X^3 + 0*X^2 + 1*X^1 + 1*X^0 = X^4+X+1$).
- Для варіанта С породжувальний поліном K'(X), двоїстий поліному ДО(X).

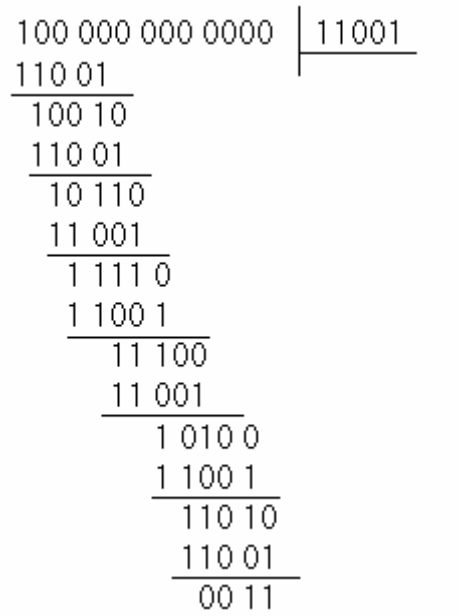
  $$K'(X) = X^{deg(X)} * K(X\text{-}1) = X^4 *(X\text{-}4 +X^{-1} +1) = X^4 *(X\text{-}4 +X^{-1} +1)= = X^4 +X^3 +1.$$

- 3. Побудова породжувальної матриці $P_{(n,k)} = I\ H_p$, де I – одинична матриця (інформаційна підматриця), H_p – перевірочна підматриця.
  Інформаційна підматриця (розмір kxk)

$$U_k = \begin{vmatrix} 100\ 000\ 000 \\ 010\ 000\ 000 \\ 001\ 000\ 000 \\ 000\ 100\ 000 \\ 000\ 010\ 000 \\ 000\ 001\ 000 \\ 000\ 000\ 100 \\ 000\ 000\ 010 \\ 000\ 000\ 001 \end{vmatrix}$$

- Перевірочна підматриця H_p складається із залишків ділення інформаційного рядка, доповненого p нулями, на породжувальний поліном.

Залишок від першого рядка, доповненого p нулями



Залишок від другого рядка, доповненого р нулями

```
100 000 000 000 | 11001
110 01
 100 10
 110 01
  10 110
  11 001
   1 111 0
   1 100 1
     11 100
     11 001
        1 010 0
        1 100 1
          110 1
```

Після аналогічних обчислень отримуємо породжувальну матрицю:

$$P_{(13,9)} = \begin{array}{c} a_1\ a_2\ a_3\ a_4\ a_5\ a_6\ a_7\ a_8\ a_9\ b_1\ b_2\ b_3\ b_4 \\ \left|\begin{array}{cccccccccccc} 1&0&0&0&0&0&0&0&0&0&0&1&1 \\ 0&1&0&0&0&0&0&0&0&1&1&0&1 \\ 0&0&1&0&0&0&0&0&0&1&0&1&0 \\ 0&0&0&1&0&0&0&0&0&0&1&0&1 \\ 0&0&0&0&1&0&0&0&0&1&1&1&0 \\ 0&0&0&0&0&1&0&0&0&0&1&1&1 \\ 0&0&0&0&0&0&1&0&0&1&1&1&1 \\ 0&0&0&0&0&0&0&1&0&1&0&1&1 \\ 0&0&0&0&0&0&0&0&1&1&0&0&1 \end{array}\right| \end{array}$$

Декодер

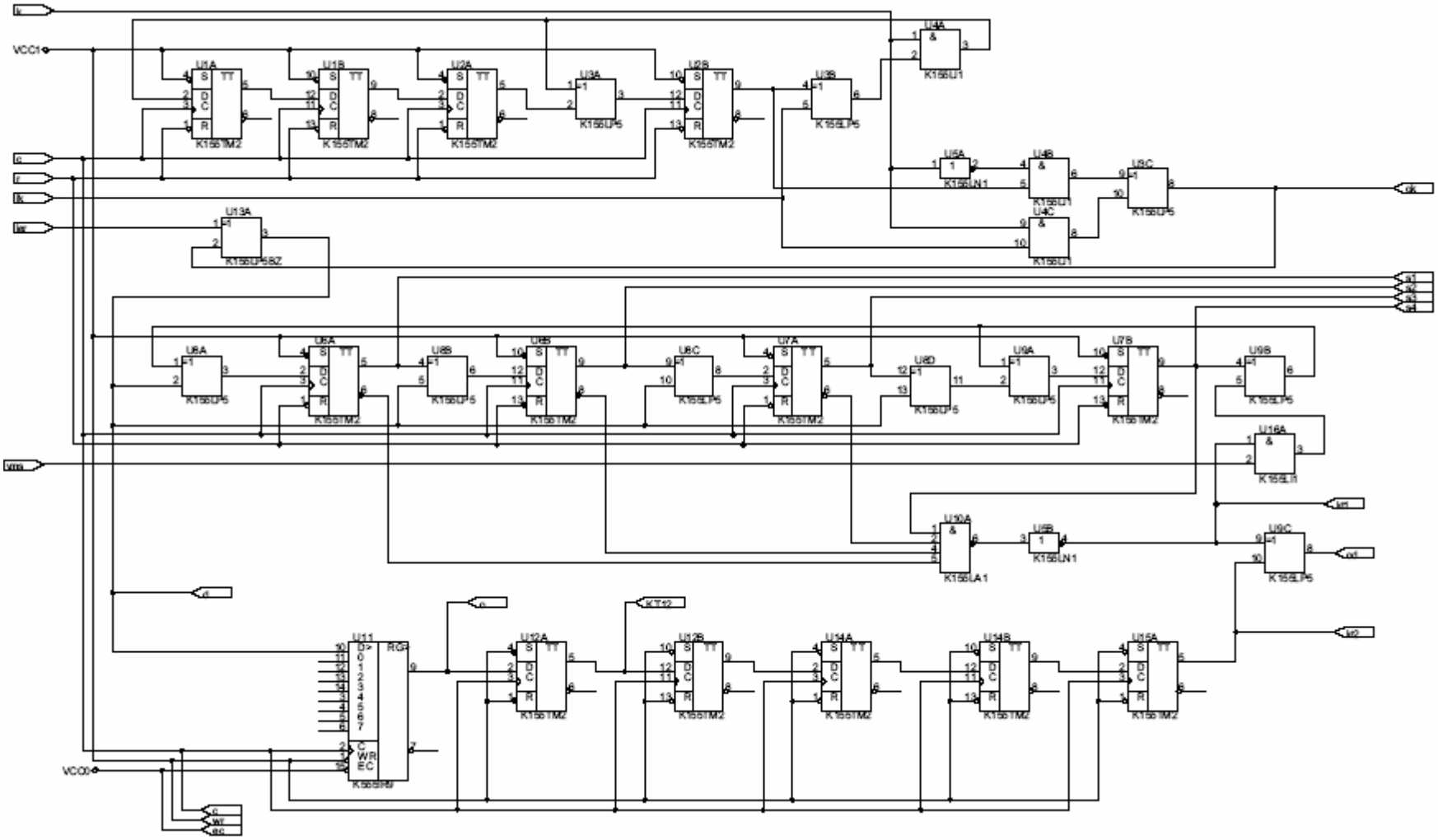
- Оскільки код (13, 9) є вкороченим (n-i, k-i) циклічним кодом, необхідно знайти залишок від ділення полінома $X^{(n-k+i)}$ на породжувальний поліном. Параметр укорочування i=2, тому що код (13, 9) утворюється шляхом укорочування коду (15, 11) (параметри повного коду Хеммінга (2^p - 1, 2^p - 1 - p)); n-k=p=4, n-k+i=4+2=6.
- Визначення залишку від ділення $X^{n-k+i} = X^6$ на породжувальний поліном:

$R(X^6) = X^3 + X^2 + X + 1$

```
X^6                    | X^4 + X^3 + 1
X^6 + X^5 + X^2        | X^2 + X + 1
    X^5 + X^2
    X^5 + X^4 + X
        X^4 + X^2 + X
        X^4 + X^3 + 1
            X^3 + X^2 + X + 1
```

Принципова схема кодера й декодера

1) **У форматі PDF**
2) **Кодер, імітація помилок і декодер**



Позначення сигналів

r - (reset) сигнал скидання;
c - (clock) синхросигнали;
ik - вхід кодера;
k - керування ключем;
ok - вихід кодера;
ier - (input error) вхід імітації помилок;
od - вихід декодера;
s1 - s4 - розряди синдрому;
vms - сигнал дозволу модифікації синдрому.

ЛІТЕРАТУРА

Основна:

1. Richard E Blahut. Theory and Practice of Error Control codes / Addison-Wesley Publishing Company, 1986.– 576 p.
2. Peterson W.W., Weldon E.J.,Jr. Error-correcting codes.- 2nd ed.- Cambridge (Mass.): MIT Press., 1971.– 595 p.
3. Блейхут Р. Теория и практика кодов, контролирующих ошибки: Пер. с англ. - М.: Мир, 1986. - 576 с.: ил.
4. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. - М.: Мир, 1976. - 595с.: ил.
5. Кузьмин В. П., Кедрус В. А. Основы теории информации и кодирования. - К.: Вища шк. Головное изд-во, 1986. - 238с.
6. Цымбал В. П. Теория информации и кодирования: Учебник. - К.: Вища шк., 1992.-263 с.: ил.

Додаткова:

7. Сидельников В.М. Теория кодирования. – М.: 2006. - 289с.
8. Shu Lin, Daniel J. Costello. Error Control Coding.Fundamentals and Applications/ Prentice-Hall, 1983. - 617 p.
9. Michael Purser. Introduction to Error-Correcting Codes/ Artech House, 1995. - 133 p.
10. W. Carry Huffman, Vera Pless. Fundamentals of Error-Correcting Codes/ Cambridge University Press., 2003. - 662 p.
11. Robert H. Morelos-Zaragoza. The art of error correcting coding/ SONY Computer Science Laboratories, Inc. JAP, John Wiley & Sons, Ltd, 2002. – 219 p.
12. Todd K. Moon. Error Correction Coding. Mathematical Methods and Algorithms/ Wiley-Interscience, John Wiley & Sons, Inc., 2005.- 755p.
13. Огнев И. В., Сарычев К. Ф. Надежность запоминающих устройств. - М.: Радио и связь, 1988. - 224 с.
14. Дяченко О.Н. Графический способ представления сверточных кодов// Наукові праці Донецького національного технічного університету. Серія “Інформатика, кібернетика і обчислювальна техніка” (ІКОТ-2007). Випуск 8 (120) - Донецьк: ДонНТУ, 2007. – С.89-98.
15. Дяченко О.Н. Аппаратная реализация и корректирующие возможности кодов Рида-Соломона// Наукові праці Донецького національного технічного університету. Серія “Проблеми моделювання та автоматизації проектування динамічних систем” (МАП-2007). Випуск: 6 (127) - Донецьк: ДонНТУ. - 2007. – С.113-121.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

ЗВІТ
з контрольної роботи № __
з курсу
“Найменування дисципліни”

Виконав:
ст. гр. СП-ХХу
Іваненко І.І.
Перевірив:
Петренко П.П.

Донецьк-20ZZ

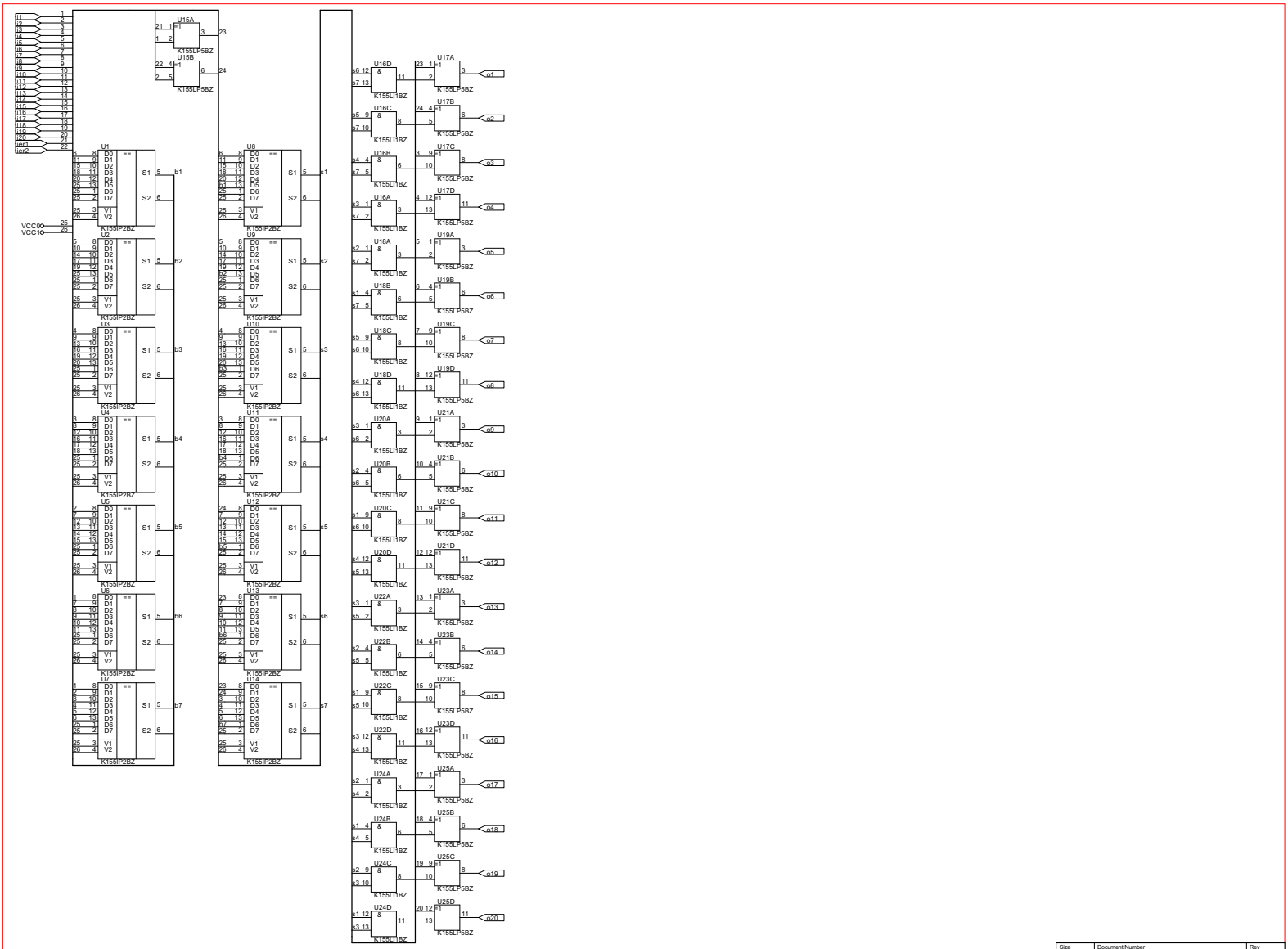
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ, МОЛОДЕЖИ И СПОРТА УКРАИНЫ

ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

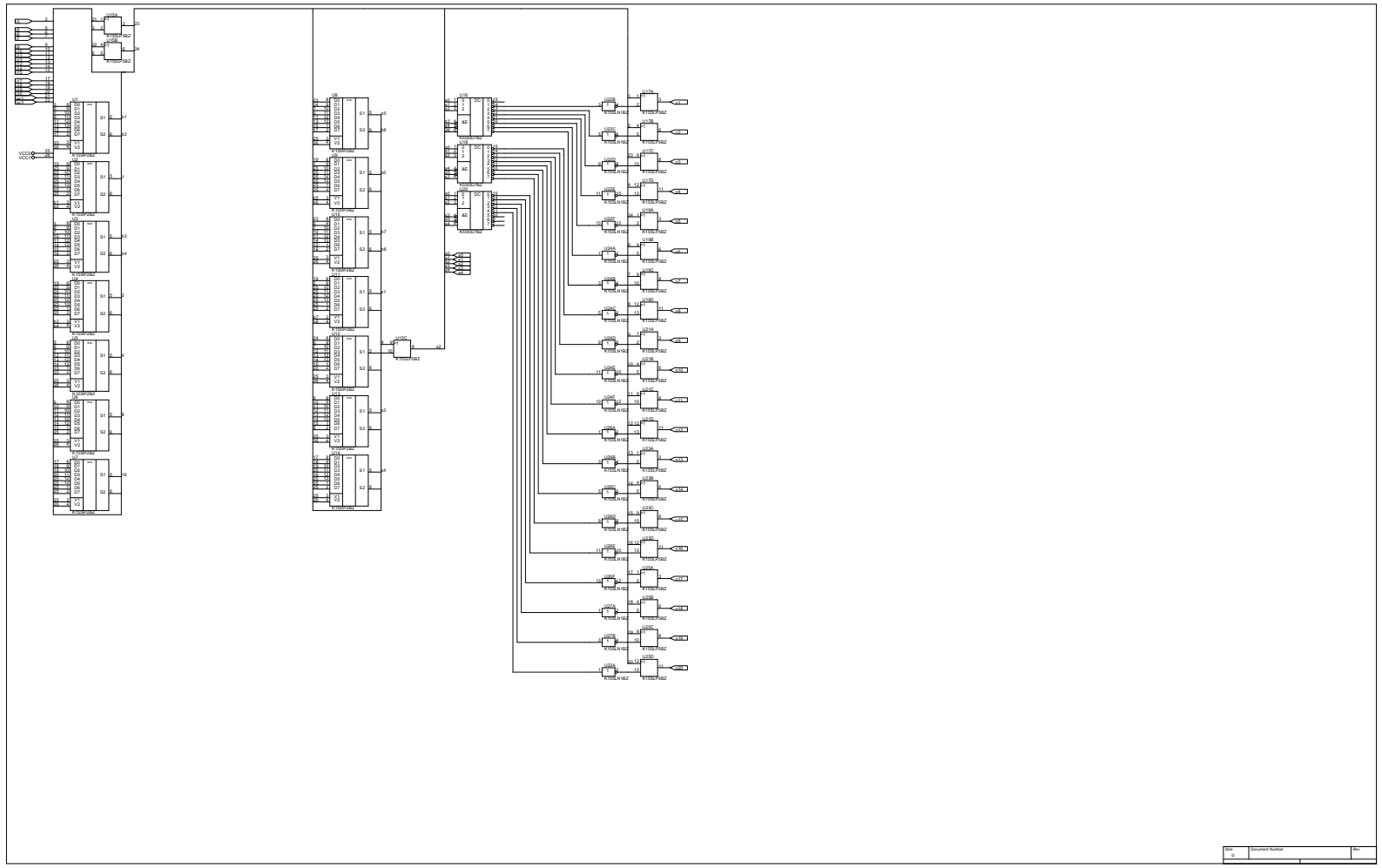
ОТЧЕТ
по контрольной работе №__
по курсу
“Наименование дисциплины”

Выполнил:
ст. гр. СП-ХХу
Иваненко И.И.
Проверил:
Петренко П.П.

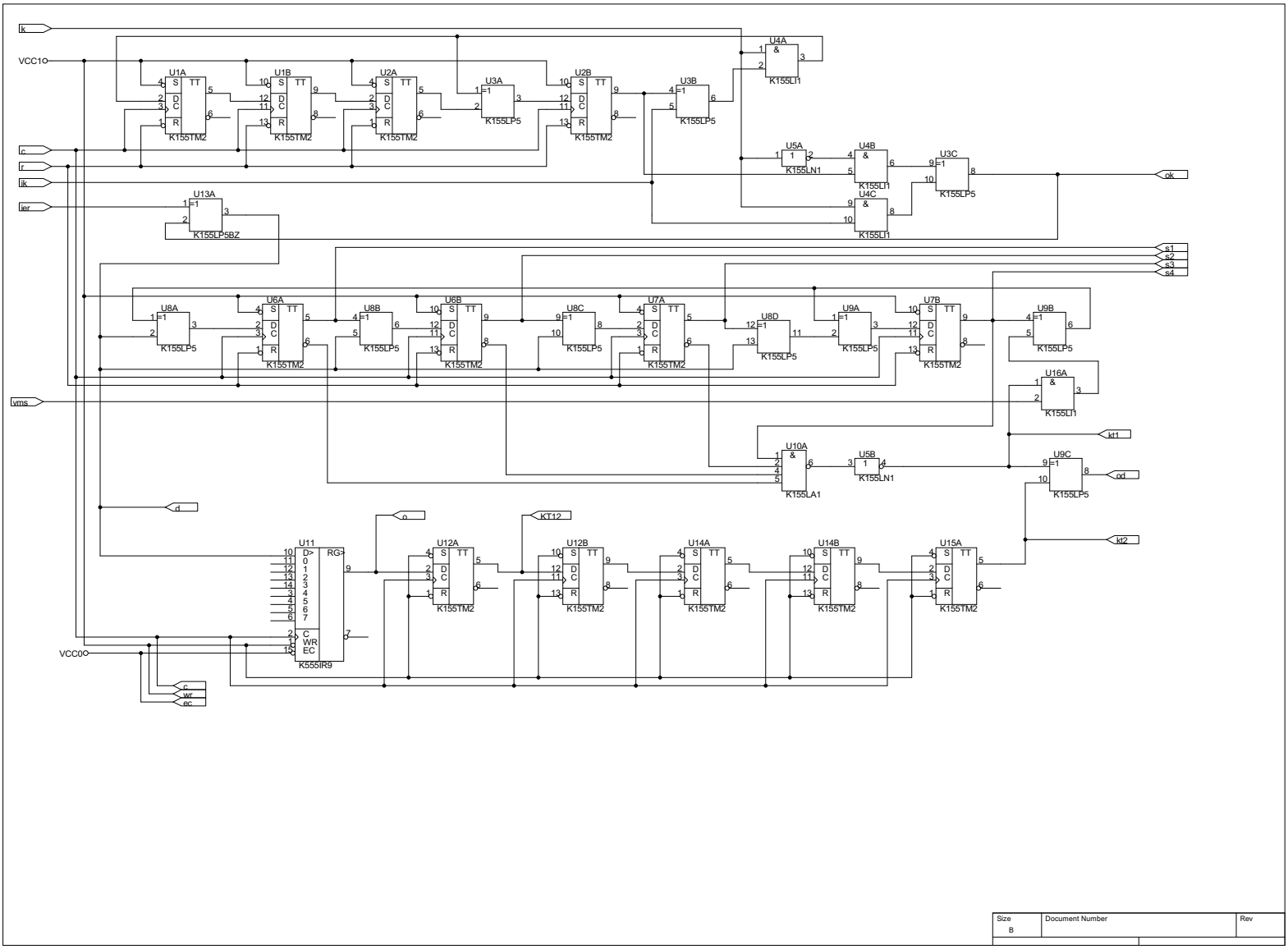
Донецк-20ZZ



Size	Document Number	Rev
C		



Size	Document Number	Rev
D		



Size	Document Number	Rev
B		