Полтавський Військовий Інститут Зв’язку

**Кафедра схемотехніки радіоелектронних систем**

##### **ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА МІКРОПРОЦЕСОРИ**

напрям підготовки 0924 «Телекомунікації»

**Компаратори слів, перетворювачі кодів та схеми контролю.**

Полтава – 2006

**Навчальна література.**

1. Тиртишніков О.І., Корж Ю.М. Обчислювальна техніка та мікропроцесори. Частина 2. Цифрові автомати: Навчальний посібник. – Полтава: ПВІЗ, 2006, с. 20 – 33.
2. Калабеков Б.А., Мамзелев И.А. Цифровые устройства и микропроцессорные системы. М.: Радио и связь, 1987.

**1. Компаратори слів (схеми порівняння).**

*Компаратори слів*(схеми порівняння) – це комбінаційні цифрові вузли, які виконують функцію порівняння двох кодових слів визначеної розрядності. Основними операціями, що виконуються схемами порівняння, є визначення ознаки рівності або нерівності двох *n*-розрядних чисел, причому операція порівняння може супроводжуватися визначенням знака нерівності.

Розглянемо синтез схеми порівняння двох трирозрядних кодових слів *X2X1X0* і *Y2Y1Y0*, яка має три виходи (*Y=X, Y>X, Y<X*), за умови, що активний рівень сигналів – логічна 1.

Таблиця істинності схеми буде мати вигляд:

*Таблиця 1*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **№ набору** | ***X2*** | ***X1*** | ***X0*** | ***Y2*** | ***Y1*** | ***Y0*** | ***Y=X*** | ***Y<X*** | ***Y>X*** |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| ... | ... | ... | ... | ... | ... | ... | … | … | … |
| 56 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| 57 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| ... | ... | ... | ... | ... | ... | ... | … | … | … |
| 63 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |

Очевидно, що синтез схеми традиційним методом – з поданням вихідних функцій у вигляді ДДНФ та подальшою їх мінімізацією буде занадто складним, оскільки потребує мінімізації трьох функцій шести змінних. Тому виконаємо синтез схеми з використанням евристичних прийомів та без визначення обмежень на застосування елементів тих чи інших типів.

По-перше, очевидно, що схема, яка виконує функцію *Y = X,* може бути реалізована порозрядним порівнянням слів *X2X1X0* та *Y2Y1Y0*, за допомогою елементів рівності та елемента „ТА” (якщо всі розряди двох кодових слів попарно дорівнюють один одному, то і кодові слова в цілому еквівалентні). Відповідна схема зображена на рис. 1.

По-друге, будь-яка з трьох вихідних функцій може бути виражена через дві інші. Наприклад, якщо *Y* не менше *X* та *Y* не дорівнює *X*, то *Y > X*. Це твердження можна подати таблицею істинності (табл. 2). Отримана таблиця істинності може бути реалізована елементом АБО-НІ на два входи, як показано на рис. 2.

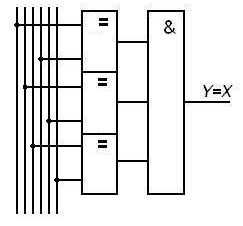
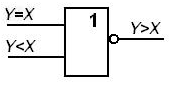
*Таблиця 2*

|  |  |  |
| --- | --- | --- |
| **Вхідні функції** | | **Вихідна функція** |
| *Y=X* | *Y<X* | *Y>X* |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Таким чином, залишилося синтезувати схему, яка буде реалізувати функцію *Y < X.* Вона може бути отримана на підставі наступного твердження: *для того, щоб одне кодове слово було більше, ніж друге, достатньо, щоб старший розряд першого слова був більшим, ніж старший розряд другого, або щоб будь-який розряд першого слова був більшим відповідного розряду другого слова за умови, що старші розряди обох слів попарно рівні.*

Функція порівняння відповідних окремих розрядів двох кодових слів за умови *Xn > Yn* може бути отримана на підставі таблиці істинності (табл. 3).

**X1Y1X2Y2X3Y3**



*Рис. 2. Схема, що виконує функцію Y > X*

*Рис. 1. Схема, що виконує функцію Y = X*

*Таблиця 3*

|  |  |  |
| --- | --- | --- |
| *X* | *Y* | *FX>Y* |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |



Таким чином, схема порівняння відповідних окремих розрядів двох кодових слів, за умови *X > Y,* являє собою елемент ТА, а результати порівняння старших розрядів кодових слів, за умови *Y = X,* можуть бути отримані з виходів відповідних елементів рівності схеми, що зображена на рис.1.

Трирозрядна схема порівняння відповідних розрядів двох кодових слів, за умовою *X >Y,* повинна реалізовувати логічну функцію

*FX>Y = F1 + F2 + F3,*

де:



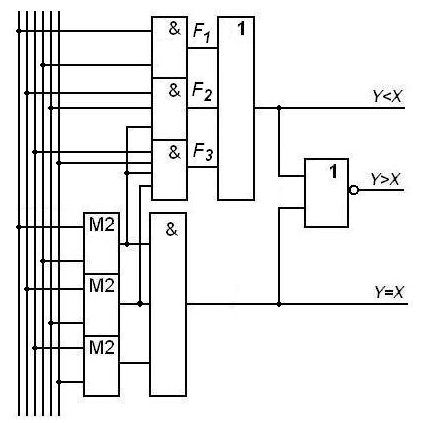
Так як у схемах порівняння окремих розрядів вхідних кодових слів використовуються інверсії розрядів слова *Y*, доцільно для зменшення загальної кількості входів схеми перетворити схему, що виконує функцію *Y = X* (рис. 1) таким чином, щоб вона також використовувала інверсії розрядів слова *Y* як вхідні аргументи. Це може бути зроблено наступним чином:

.



Тобто в схемі, що виконує функцію *Y = X*, будуть використовуватися замість елементів рівності суматори за модулем 2.

Отримана схема компаратора трирозрядних кодових слів зображена на рис. 3.



*Рис. 3. Схема компаратора трирозрядних кодових слів*

**2. Перетворювачі кодів та схеми контролю.**

**2.1. Перетворювачі кодів.**

У цифрових пристроях часто виникає необхідність перетворення інформації з одного двійкового коду в інший. Коди, що відрізняються від найбільш простого натурального 8421, наприклад, застосовуються:

* у цифрових пристроях, ЕОМ та системах передачі даних для виявлення і корекції помилок (код з контролем на парність; код Хеммінга, циклічні коди);
* у перетворювачах аналогових фізичних сигналів у цифрові сигнали для забезпечення погрішності перетворення, що не перевищує одиниці молодшого розряду (код Грея);
* при виконанні арифметичних операцій в ЕОМ (прямий, зворотний, додатковий коди); для введення в ЕОМ даних (ДДК);
* для побудови цифрових індикаторів (семисегментний код).

Для синтезу перетворювачів кодів можна користуватися двома методами.

1. Перетворення вихідного двійкового коду в десятковий і наступне перетворення десяткового коду у необхідний двійковий код.
2. Використання комбінаційного логічного пристрою, що безпосередньо реалізовує необхідне перетворення.

Перший метод структурно реалізовується з'єднанням дешифратора і шифратора і є зручним у випадках, коли можна використовувати стандартні дешифратори і шифратори в інтегральному виконанні. Використання другого методу нерідко зменшує апаратні витрати на реалізацію перетворювача.

Розглянемо синтез перетворювача другим методом на прикладі трирозрядного перетворювача натурального двійкового коду код Грея, що також називають циклічним або рефлексно-двійковим.

Особливістю коду Грея є те, що в ньому кодові комбінації двох сусідніх чисел відрізняються тільки в одному двійковому розряді. Відповідність трирозрядних двійкових чисел їх кодам Грея показано в таблиці 4.

*Таблиця 4*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Код 8421** | | | **Код Грея** | | |
| ***Х2*** | ***Х1*** | ***Х0*** | ***У2*** | ***У1*** | ***У0*** |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 0 | 0 |

Безпосередньо з таблиці запишемо функції виходів *Y0* і *Y1* у ДДНФ. Що стосується функції виходу *Y2*, із таблиці видно, що *Y2 = X2*.



Застосовуючи до отриманих виразів правило склеювання, маємо (у першому виразі склеюються перший і третій, другий і четвертий додаток, у другому - перший і другий, третій і четвертий):



Таким чином, найпростіше даний перетворювач реалізовується на двох суматорах по модулю два. Відповідна схема зображена на рис. 4.

Взагалі, схема *n*-розрядного перетворювача натурального двійкового коду в код Грея може бути побудована на основі наступного правила: старші розряди вхідного та вихідного кодів співпадають, а будь-який наступний розряд *Yk* коду Грея дорівнює сумі по модулю два відповідного *Xk* та попереднього *Xk-1* розрядів натурального коду:

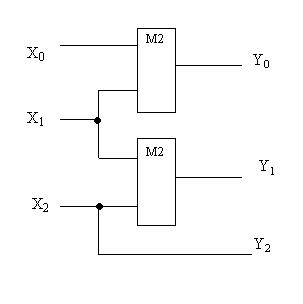


Узагальнений варіант УГП перетворювача кодів показаний на рис. 5а. Чинні стандарти допускають заміну літер *Х* і *Y* позначеннями типу подання вхідної і вихідної інформації відповідно. Зокрема, для двійкового коду (стандартного) можна використовувати літеру *В*, для десяткового коду скорочення *DЕС*, для коду Грея – *G*, для семисегментного коду *7S* і т.д. Таким чином, позначення синтезованого трирозрядного перетворювача може мати вигляд, зображений на рис. 5б.

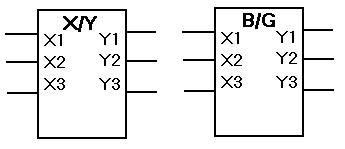
**Поняття про кодування, захищене від завад.**

Унаслідок дії завад під час передачі, обробки та збереження двійкових кодів у МПС можуть статися помилки, наприклад, прийом 1 замість 0 або навпаки. Це може привести до неправильного результату роботи МПС.

Одним з найбільш ефективних шляхів захисту інформації у МПС є кодування, стійке до завад, яке здійснюється введенням у кодові слова додаткових бітів, призначених або для виявлення і виправлення помилок, або тільки для виявлення помилок. Відповідно до цього коди, стійкі до завад, поділяють на коректувальні коди, які виявляють і виправляють помилки, та коди, які тільки виявляють помилки.



*Рис. 4. Перетворювач натурального двійкового коду у код Грея*



*а) б)*

*Рис. 5. УГП перетворювачів кодів : а – загальне позначення,  
б – позначення трирозрядного перетворювача натурального коду у код Грея*

Можливість виявлення помилок за наявності додаткових бітів обумовлена тим, що для передачі інформації використовуються не всі можливі комбінації *n*-розрядного двійкового коду, а лише деяка частина з них. Дозволені комбінації вважаються безпомилковими, інші є забороненими. Поява заборонених комбінацій розглядається як помилка. Слід уявляти, що можливі такі помилки, за яких одна дозволена комбінація переходить у іншу. У цьому випадку помилки не виявляються.

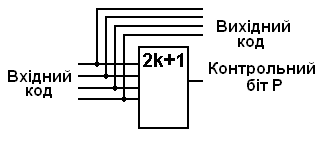
Найпростішим поширеним кодом, стійким до завад, який використовується у мікропроцесорній техніці, є код з контролем на парність. У цьому коді до інформаційних бітів праворуч додається один контрольний біт. Якщо кількість одиниць в інформаційних бітах є парною, то значення контрольного біта дорівнює 0, у протилежному випадку – 1. Отже, у будь-якому випадку кількість одиниць у повній послідовності (кодовому слові з контрольним бітом) є парною. Якщо при перевірці після передачі кількість одиниць є непарною, то це означає, що відбулася помилка. Код із контролем на парність дозволяє виявляти всі помилки непарної кратності (у одному біту кодового слова) і не дозволяє виявляти помилки парної кратності (у двох бітах кодового слова одночасно).

Неважко помітити, що алгоритм отримання додаткового контрольного біту для *n*-розрядного паралельного кодового слова співпадає з логічною функцією „виключне АБО” *n* аргументів.

Схема отримання контрольного біта Р також може бути реалізована на суматорах за модулем 2 (елементах „виключне АБО” на 2 входи). У цьому разі схема будується за багатоярусним принципом: спочатку попарно додаються значення окремих розрядів кодового слова, потім отримані результати також попарно додаються за допомогою суматорів за модулем 2 другого ярусу і т.д., наприклад:



Багатоярусні схеми на суматорах за модулем 2 часто називають схемами згортки. На рис. 6 зображена схема отримання двійкового коду з контролем на парність, де схема згортки позначена прямокутником з написом *2k+1*.



*Рис. 6. Схема отримання двійкового коду з контролем на парність*

Очевидно, що схема, яка перевіряє на парність прийняті кодові слова (схема контролю на парність), також являє собою логічний елемент „виключне АБО” *n + 1* аргументів (з урахуванням контрольного біта), або багатоярусну схему на суматорах за модулем 2, яку часто називають схемою згортки.

**Висновок:** аналізуючи схеми перетворювачів кодів та схем контролю, які були розглянути вище, неважко помітити, що в більшості з них використовуються суматори за модулем 2, або схеми згортки, побудовані на основі цих суматорів.

Додатковий матеріал для самостійного вивчення: (Тиртишніков О.І., Корж Ю.М. Обчислювальна техніка та мікропроцесори. Частина 2. Цифрові автомати: Навчальний посібник. – Полтава: ПВІЗ, 2006, с. 25 – 28.

Іншим поширеним кодом є код Хеммінга, що виявляє і виправляє одноразові помилки. Кожній з *2n-1* ненульових комбінацій *n*-розрядного кодового слова відповідає комбінація з *n + k* бітів. Значення контрольних бітів отримують в результаті додавання за модулем 2 значень бітів у деяких визначених інформаційних розрядах. Із загальної кількості *2n+k-1* можливих помилок код Хеммінга може виявити та виправити *2k-1* помилок.

Припустимо, що треба передати або обробити 15 різних двійкових повідомлень. Без кодування для цього достатньо чотирьох інформаційних бітів (*n* = *4*). Потрібну кількість додаткових контрольних бітів обчислюють за формулою *2k-1* = *n + k*. Звідки визначають кількість перевірних розрядів та кількість одноразових помилок, які можуть бути виявлені та виправлені. У цьому випадку кількість додаткових розрядів *k* = *3,* а кількість одноразових помилок – *2k-1 = 7*.

Контрольні біти *ki* розташовують у послідовності інформаційних бітів *uj*на позиціях із номерами *2i-1*, як показано у табл. 5.

*Таблиця 5*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Позиція | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 001 | 010 | 011 | 100 | 101 | 110 | 111 |
| Біт | k1 | k2 | u1 | k3 | u2 | u3 | u4 |

Значення перевірних бітів *ki* обчислюється додаванням за модулем 2 значень бітів, у двійковому виразі номерів яких наявна одиниця в *i*-му розряді. Відповідно, для обчислення значення *k1* потрібно додати за модулем 2 значення бітів із непарними номерами:

.



Для визначення *k2* треба додати за модулем 2 біти, у двійковому виразі номерів яких наявна одиниця у другому розряді, тобто:

.



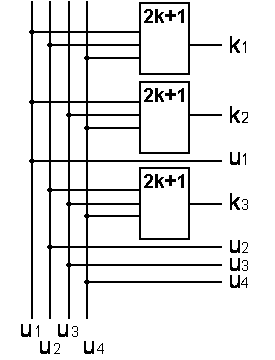
Контрольний біт *k3* визначається додаванням за модулем 2 бітів, у двійковому виразі номерів яких наявна одиниця у третьому розряді:

.



Схема перетворювача чотирирозрядних кодових слів у код Хеммінга зображена на рис. 7.

Визначення та виправлення помилок здійснюється *k* перевірками. При кожній перевірці додаються за модулем 2 біти прийнятої послідовності інформаційних та контрольних розрядів, двійкові номера яких мають одиницю в першому, другому і так далі розрядах. Якщо під час передавання не було збою, то результати всіх перевірок дорівнюють нулю. Якщо збій відбувся, то хоча б одна перевірка не дорівнює нулю. У цьому випадку треба сформувати кодову комбінацію з результатів перевірок, який вкаже на розряд, де відбувся збій (він має назву синдром). Молодший розряд коду результатів перевірок формує перша перевірка, старший – остання. Інверсія біта в розряді з одержаним номером виправить помилку.



*Рис. 7. Схема перетворювача чотирирозрядних кодових слів в код Хеммінга*

Наприклад, необхідно сформувати код Хеммінга, що виявляє та виправляє одноразові помилки у послідовності:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 1 | 0 | 0 |
| u1 | u2 | u3 | u4 |

Відповідно,



Послідовність, що закодована кодом Хеммінга, буде мати вигляд:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 1 | 1 | 1 | 0 | 0 |
| k1 | k2 | u1 | k3 | u2 | u3 | u4 |

Нехай після передачі відбувся збій в одному розряді і прийнята послідовність 0110100 (помилка в четвертому розряді – *k3*). Тоді перша та друга перевірки дадуть значення 0, а третя – 1:

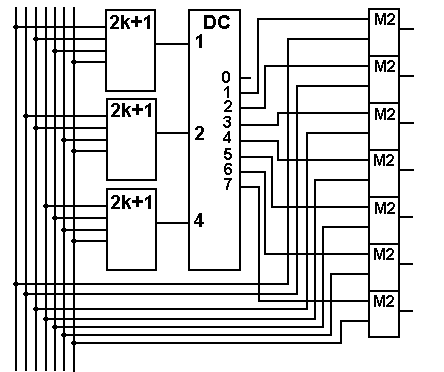


Код 100, що створюють результати перевірок, вказує, що відбувся збій у четвертому розряді. Якщо проінвертувати четвертий розряд, то одержимо виправлену послідовність 0111100.

Із розглянутого прикладу видно, що схема пристрою контролю повинна містити такі складові частини: схеми згортки (елементи „виключне АБО”) відповідно кількості перевірок, що виконуються; дешифратор, який на основі отриманого синдрому керує інверсією помилкового розряду та елементів, що виконують саму інверсію. Така схема зображена на рис. 16 (схеми згортки позначені прямокутником з написом *2k* + *1*). У якості „керованих інверторів” у схемі використовуються суматори за модулем 2.

k1k2u1k3u2u3u4

*Вихідний код*



*Рис. 16. Схема пристрою контролю*