***Помехоустойчивый код Хэмминга***

1. Цель работы

Изучить принципы помехоустойчивого кодирования, получить навыки моделирования помехоустойчивых кодов с помощью Electronics Workbench (EWB) или Multisim.

2. Общие сведения

При работе устройств вычислительной техники и телекоммуникационной аппаратуры возможно появление ошибок в обрабатываемых цифровых данных. Причинами сбоев могут быть мощные электромагнитные помехи, резкое изменение напряжения питания, старение радиоэлементов, ненадёжный контакт разъёмов, радиоактивное излучение естественных и искусственных источников и т.п.

Сбои проявляются в виде случайного изменения одного или нескольких битов машинного слова (вместо единицы в отдельных разрядах передаётся ноль или наоборот).

Автоматическое обнаружение и исправление ошибок сопровождается введением избыточности в передаваемые или хранимые данные. Для этих целей разработаны специальные коды, в которые помимо информационных битов b1b2...bn дополнительно вводят контрольные (проверочные) биты k1k2...km .

Контрольные биты позволяют проверять целостность (не искаженность) информационных битов машинного слова, а наиболее сложные коды могут не только обнаружить, но и исправить неверно принятые биты. Разработанные помехоустойчивые коды позволяют решать разные задачи: обнаружить одиночную ошибку, обнаружить и исправить единственную ошибку, обнаружить и исправить несколько ошибок. Первые коды называются обнаруживающими, а вторые – корректирующими кодами.

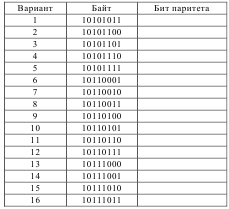
На передающей стороне формирование помехоустойчивого кода осуществляют с помощью специального устройства – кодера. Обнаружение и исправление ошибок на приёмной стороне производят с помощью устройств, которые называются декодерами. В данной лабораторной работе кодер и декодер строят с помощью комбинационных цифровых устройств (логические элементы, которые выполняют функцию неравнозначности). 3. Задания на выполнение лабораторной работы

3.1. Задание 1.

Формирование бита чётности

Сформировать бит чётности (бит паритета) для заданного байта передаваемых данных. При выполнении задания нужно в правую крайнюю колонку таблицы 3.1.1 записать единицу или ноль.

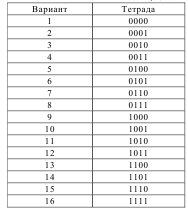
Таблица 3.1.1



3.2. Задание 2.

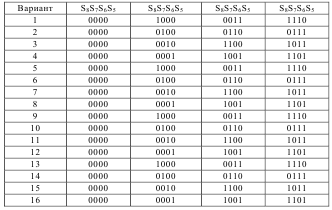
Исследование помехоустойчивого кода с формированием бита чётности

Выполнить моделирование процесса передачи информации (одной тетрады - половины байта). Исходные данные приведены в таблицах 3.2.1 и 3.2.2. Для моделирования использовать программы Electronics Workbench (EWB) или Multisim. Таблица 3.2.1



Моделирование следует выполнить четырежды при заданных значениях помех (табл.3.2.2). Результаты моделирования в виде таблицы поместите в отчёт. Прокомментируйте полученные результаты.

Таблица 3.2.2

3.3. Задание 3. Исправление ошибки с помощью кода Хэмминга

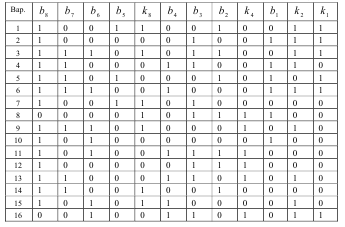
Расчётным путём (вручную) определить, в каком разряде принятого

кода Хэмминга произошло искажение. Исходные данные для разных вари-

антов приведены в таблице 3.3.1. Процесс вычисления искажённого бита

следует подробно описать в отчёте.

Таблица 3.3.1



3.4. Задание 4. Моделирование работы кода Хэмминга

С помощью программ Electronics Workbench или Multisim собрать схему для моделирования процесса передачи информации с использованием помехоустойчивого кодирования. Провести моделирование процесса передачи данных, приведённых в табл. 3.3.1 (для своего варианта). В отчёт

следует поместить схему, соответствующую заданному варианту (в том числе с указанием положения ключей и содержимого генератора слов). 4. Порядок выполнения лабораторной работы

4.1. Методические указания к заданию 3.1

Простейший код, предназначенный для обнаружения одной ошибки (точнее – для обнаружения нечётного числа ошибок), основан на добавлении к информационным битам одного контрольного бита. При этом контрольный бит должен быть таким, чтобы суммарное число единиц в образованном машинном слове было чётным. Добавляемый бит называется битом паритета.

Проверочный бит k для n-битного двоичного слова b1b2...bn вычисляется по формуле:



В результате такого преобразования формируется (n+1) – битное

слово b1b2...bnk , число единиц в котором будет чётное.

В задании 3.1 требуется для своего варианта определить необходимый бит паритета, который нужно добавить к информационным битам.

Пример.

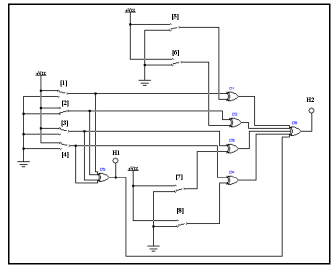
Пусть дан байт 10111100. Число информационных единиц в этом байте нечётное, поэтому бит паритета нужно установить равным единице.

В результате этого получается машинное слово 101111001.

4.2. Методические указания к заданию 3.2

Схема исследований кода с формированием бита паритета показана

на рисунке.



Переключатели S1…S4 имитируют передаваемую тетраду (четыре информационных бита). Переключатели S5…S8 симулируют помехи, возникающие при передаче или хранении цифровых данных. Нижнее положение переключателей S5…S8 соответствует отсутствию искажений в соответствующем разряде. Схема D5 формирует бит чётности (если на вход подаётся чётное число единиц, то выходной сигнал равен нулю). При этом светодиод H1 не горит.

Логическая схема Исключающее ИЛИ D6 на приёме осуществляет

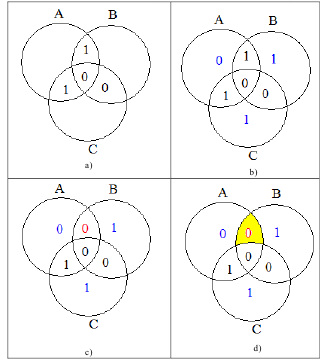
контроль искажений. Если светодиод H2 не горит, то считается, что искажений нет. Фактически это не так. Появление чётного числа искажений в

машинном слове не фиксируется (не обнаруживается) данным устройством. Это является существенным недостатком данного метода контроля искажений. Такой код способен лишь сигнализировать о наличии редко появляющихся одиночных сбоев (точнее, нечётного числа сбоев).

4.3. Методические указания к заданию 3.3

Проиллюстрируем идею обнаружения и определения неверно принятого бита с помощью кода Хэмминга. Для этого используем диаграммы Вена [1]. Предположим, что передаётся тетрада 1010. Запишем эти четыре бита в сегменты трёх пересекающихся окружностей.

Таблица 4.3.1



Окружности A, B и C дают семь сегментов. В четыре внутренних сегмента поместим информационные биты числа 1010 (табл. 4.3.1 a). В оставшиеся три сегмента поместим контрольные биты (табл. 4.3.1 b).

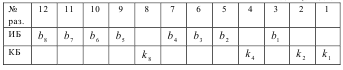
Правило формирование контрольных битов такое: в каждой окружности должно быть чётное число единиц. В данном случае в каждой окружности получилось по две единицы. Пусть в процессе передачи информации один информационный бит будет искажён (табл. 4.3.1 с). На приёмной стороне осуществляется анализ принятой информации. Легко заметить, что в окружности С число единиц осталось чётным, а окружностях А и В число единиц стало нечётным. Это говорит о том, что искажённый бит находится в сегменте, который принадлежит окружностям А и В, но не принадлежит окружности С (табл. 4.3.1 d).

Рассмотрим пример нахождения искажённого бита с помощью кода Хэмминга.

Места расположения информационных битов (ИБ) и контрольных

битов (КБ) в передаваемых данных указаны в следующей таблице. В верхней строке таблицы указан порядковый номер каждого бита в машинном слове.

Таблица 4.3.2



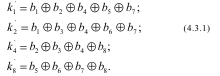
Форма записи машинного слова, приведённая в предыдущей таблице, выбрана такой с целью повышения наглядности (из методических соображений). Фактически данные представляют машинным словом, состоящим из 12-ти бит: b8b7b6b5k8b4b3b2k4b1k2k1.

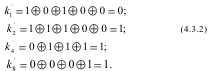
Пример.

Предположим, что в процессе передачи некоторых данных произошло искажение одного информационного бита и на приёме получены указанные в таблице 4.3.3 данные. Требуется найти и исправить искажённый информационный бит.

Таблица 4.3.3. 

Решение.

Вычислим значения контрольных битов на приёме. Будем обозначать проверочные биты на приёме со штрихом (чтобы отличить их от контрольных битов, сформированных на передающей стороне). Расчёт производится по формулам [1]: 

Используя формулу (4.3.1) и верхнюю строчку таблицы 4.3.2, получим конкретные значения контрольных битов для рассматриваемого примера на приёме: 

Результаты расчётов (4.3.2) показывают, что контрольные биты, сформированные на передающей и приёмной сторонах, различаются: 

Различие контрольных битов, сформированных на передающей и приёмной сторонах, говорит о том, что в процессе передачи произошло искажение машинного слова. Теперь необходимо определить, какой именно бит был принят неверно.

Для определения неверно принятого бита требуется вычислить так

называемый синдром S s8s4s2s1 , где 

Используя результаты (4.3.2) и нижнюю строчку таблицы 4.3.3, вычислим для рассматриваемого примера четыре бита синдрома:



Переведём синдром S 1010 2 из двоичной системы счисления (СС)

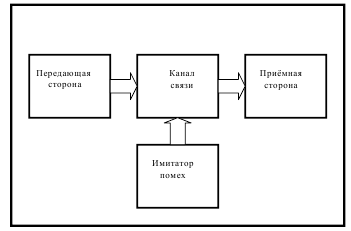
в десятичную СС S 10 10. Десятичное число 10 говорит о том, что десятый разряд принятых данных ( b6 ) искажён, и этот бит нужно исправить (проинвертировать). Таким образом, после корректировки принятые данные будут иметь вид, показанный в таблице 4.3.4. Напомним, что счёт разрядов ведётся справа налево.

Таблица 4.3.4.



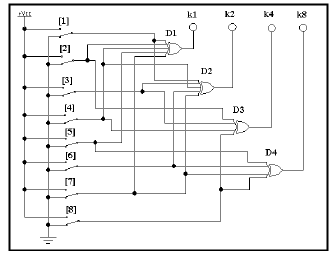
4.4. Методические указания к заданию 3.4

Структурная схема устройства для моделирования передачи данных с использованием кода Хэмминга показана на рисунке



На передающей стороне формируются контрольные биты в соответствии с выражением (4.3.1). На приёмной стороне вычисляется синдром в соответствии с выражениями (4.3.1) и (4.3.3). Имитатор помех позволяет исказить любой бит данных, передаваемых по каналу связи.

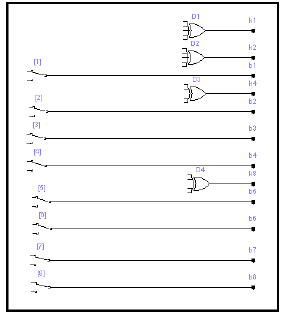
Рассмотрим поочерёдно конструкцию каждого блока, указанного на структурной схеме.

На следующем рисунке показана схема Передатчика (кодера). 

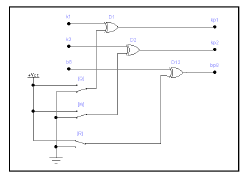
Переключатели 1…8 имитируют информационные биты передаваемых данных. С помощью четырёх логических элементов Исключающее

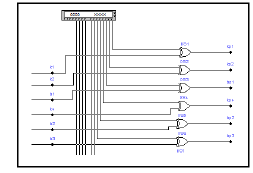
ИЛИ D1…D4 формируются контрольные биты k1, k2, k4, k8 . В канал связи передаётся 12 бит (восемь информационных и четыре контрольных бита). Схематично передаваемые по каналу связи данные показаны на следующем рисунке.

Заметим, что в устройствах телекоммуникаций информация передаётся не с помощью двенадцати проводной линии (параллельный код), а

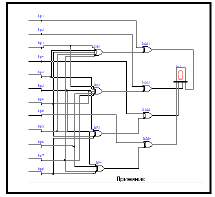
преобразуется в последовательный код, например, с помощью регистра сдвига. 

Имитатор помех позволяет исследователю моделировать возникновение сбоев в любом разряде передаваемых данных. Конструкция имитатора помех сходна с конструкцией блока формирования помех. В имитаторе помех содержится 12 логических схем Исключающее ИЛИ и 12 переключателей, с помощью которых можно изменить любой бит. Фрагмент этого блока показан на следующем рисунке. Переключатели Q, W, R могут изменять значения битов (соответственно k1, k2 , b8 ). Нижнее положение переключателей не изменяет значения передаваемых битов, а верхнее положение приводит к инверсии соответствующего бита. Указанное положение переключателей Q, W и R приведёт к тому, что биты k1, k2 будут переданы без искажений (k p1 =k1, k p2 =k2 ) , а бит k8 будет проинвертирован 



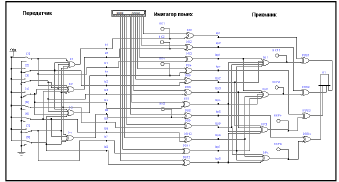
Рассмотренная конструкция имитатора помех не единственная. Этот блок можно построить иначе, например, с помощью генератора слов (см. следующий рисунок). Если генератор слов (Word Generator) формирует во всех разрядах логические нули, то искажений не происходит. Естественно, что логическая единица в каком-либо разряде управляющего слова приводит к искажению соответствующего бита (инверсии). 

На приёмной стороне поступившие данные должны быть обработаны так, чтобы можно было автоматически исправить искажённый бит. На следующем рисунке показана принципиальная схема Приёмника (декодер).



Четыре логических элемента DP1…DP4 производят расчёт контрольных битов на приёме. Следующие четыре логические схемы Исключающее ИЛИ DSS1…DSS4 вычисляют синдром. Значение синдрома в шестнадцатеричной СС отображается на индикаторе H1. Если индикатор показывает нуль, то это означает, что передача данных произошла без искажений.

При наличии сбоев в канале связи показания индикатора изменяются от 1 до CH. Данная конструкция приёмника только указывает номер искажённого разряда, но коррекции принятых данных не производит. Естественно, что путём некоторого усложнения схемы эту опцию можно реализовать.

Полная принципиальная схема устройства моделирования показана на последнем рисунке. Восемь светодиодов позволяют анализировать работу устройства в местах формирования контрольных битов. 

5. Требования к отчёту

Отчёт подготавливается в электронном виде. Он должен содержать постановки задач, результаты расчётов контрольных битов, схемы моделирования.

6. Контрольные вопросы

6.1. Можно ли исправить возникшую ошибку в передаваемых данных, если для помехоустойчивого кодирования используется лишь единственный бит паритета?

6.2. Перечислите программы, предназначенные для моделирования радиоэлектронных устройств.

6.3. Запишите формулы для формирования контрольных битов с помощью кода Хэмминга.

6.4. Нужно ли исправлять искажённые контрольные биты?

6.5. Как рассчитывается слово синдрома в коде Хэмминга?

6.6. К чему приведёт искажение контрольного бита при использовании кода Хэмминга?

6.7. Как составить схему цифрового устройства по заданному логическому выражению?

6.8. Какое действие выполняет декодер кода Хэмминга?

***Исследование кода Хемминга***

Цель работы:

1. Изучить принцип помехоустойчивого кодирования.

2. Ознакомиться с основными блоками, осуществляющими помехоустойчивое кодирование.

3. Изучить принципы работы кодера/декодера Хемминга.

Коды Хэмминга — вероятно, наиболее известный из первых самоконтролирующихся и самокорректирующихся кодов. Построены они применительно к двоичной системе счисления.

История

В середине 1940-х годов Ричард Хэмминг работал в знаменитых лабораториях фирмы Белл (Bell Labs) на счётной машине Bell Model V. Это была электромеханическая машина, использующая релейные блоки, скорость которых была очень низка: один оборот за несколько секунд. Данные вводились в машину с помощью перфокарт, поэтому в процессе чтения часто происходили ошибки. В рабочие дни использовались специальные коды, чтобы обнаруживать и исправлять найденные ошибки, при этом оператор узнавал об ошибке по свечению лампочек, исправлял и снова запускал машину. В выходные дни, когда не было операторов, при возникновении ошибки машина автоматически выходила из программы и запускала другую.

Хэмминг часто работал в выходные дни, и все больше и больше раздражался, потому что часто был должен перезагружать свою программу из-за ненадежности перфокарт. На протяжении нескольких лет он проводил много времени над построением эффективных алгоритмов исправления ошибок. В 1950 году он опубликовал способ, который известен как код Хэмминга.

Систематические коды

Систематические коды образуют большую группу из блочных, разделимых кодов (в которых все символы можно разделить на проверочные и информационные). Особенностью систематических кодов является то, что проверочные символы образуются в результате линейных операций над информационными символами. Кроме того, любая разрешенная кодовая комбинация может быть получена в результате линейных операций над набором линейно независимых кодовых комбинаций.Самоконтролирующиеся коды

Коды Хэмминга являются самоконтролирующимися кодами, то есть кодами, позволяющими автоматически обнаруживать ошибки при передаче данных. Для их построения достаточно приписать к каждому слову один добавочный (контрольный) двоичный разряд и выбрать цифру этого разряда так, чтобы общее количество единиц в изображении любого числа было, например, четным. Одиночная ошибка в каком-либо разряде передаваемого слова (в том числе, может быть, и в контрольном разряде) изменит четность общего количества единиц. Счетчики по модулю 2, подсчитывающие количество единиц, которые содержатся среди двоичных цифр числа, могут давать сигнал о наличии ошибок. При этом невозможно узнать, в каком именно разряде произошла ошибка, и, следовательно, нет возможности исправить её. Остаются незамеченными также ошибки, возникающие одновременно в двух, четырёх, и т.д. — в четном количестве разрядов. Впрочем, двойные, а тем более четырёхкратные ошибки полагаются маловероятными.

Построение кодов Хемминга основано на принципе проверки на четность числа единичных символов: к последовательности добавляется такой элемент, чтобы число единичных символов в получившейся последовательности было четным.

Операция контроля четности двоичных чисел позволяет повысить надежность передачи и обработки информации. Ее сущность заключается в суммировании по модулю 2 всех разрядов с целью выяснения четности числа, что позволяет выявить наиболее вероятную ошибку в одном из разрядов двоичной последовательности. Например, если при передаче кода 1001 произойдет сбой во втором разряде, то на приемном пункте получим код 1101 - такую ошибку определить в общем случае затруднительно. Если же код относится к двоично-десятичному (8-4-2-1), то ошибку легко обнаружить, поскольку полученный код (десятичный эквивалент - число 13) не может в принципе принадлежать к двоично-десятичному.

Обнаружение ошибок путем введения дополнительного бита четности происходит следующим образом. На передающей стороне передаваемый код анализируется и дополняется контрольным битом до четного или нечетного числа единиц в суммарном коде. Соответственно суммарный

код называется четным или нечетным. В случае нечетного кода дополнительный бит формируется таким образом, чтобы сумма всех единиц в передаваемом коде, включая контрольный бит, была нечетной. При контроле четности все, естественно, наоборот. Например, в числе 0111 число единиц нечетно. Поэтому при контроле нечетности дополнительный бит должен быть нулем, а при контроле четности - единицей. На практике чаще всего используется контроль нечетности, поскольку он позволяет фиксировать полное пропадание информации (случай нулевого кода во всех информационных разрядах). На приемной стороне производится проверка кода четности. Если он правильный, то прием разрешается, в противном случае включается сигнализация ошибки или посылается передатчику запрос на повторную передачу. Схема формирования бита четности для четырехразрядного кода показана на рис. 1. Она содержит четыре элемента Исключающее ИЛИ, выполняющих функции сумматоров по модулю 2 (без переноса) и состоит из трех ступеней. На первой ступени попарно суммируются все биты исходного кода на входах А, В, С, D. На второй ступени анализируются сигналы первой ступени и устанавливается четность или нечетность суммы входного кода. На третьей ступени полученный результат сравнивается с контрольным сигналом на входе Е, задающим вид используемого контроля, в результате чего на выходе F формируется пятый дополнительный бит четности, сопровождающий информационный сигнал в канале передачи.

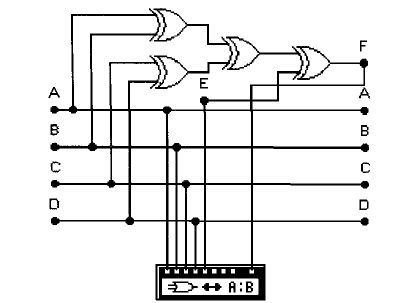


Рис. 1. Схема формирователя бита четности четырехразрядного кода

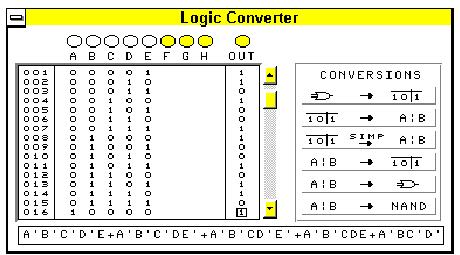


Рис. 2. Результаты моделирования схемы

Результаты моделирования формирователя показаны на рис. 2 в виде таблицы истинности из 32 возможных двоичных комбинаций и булева выражения (на рис. 2 видны только первые 16 из 32 комбинаций, остальные просматриваются с помощью линейки прокрутки). Для просмотра составляющих булева выражения необходимо мышью поместить курсор в дополнительный дисплей и передвигать его клавишами управления курсором

В библиотеке программы EWB схема проверки на четность и нечетность представлена ИМС 74280 (аналог - К555ИП5), схема ее включения показана на рис. 3. ИМС К555ИП5 имеет 9 входов (А, В...I) и два выхода (EVEN, ODD), один из которых - инверсный. Вход I используется для управления видом контроля (0 - контроль четности, 1 - контроль нечетности) и управляется переключателем Z (управляется с клавиатуры одноименной клавишей). Вывод NC - not connection - пустой, т.е. внутри ИМС к нему ничего не подключено.

Правильность функционирования схемы на рис. 3 проверяется с помощью генератора слова, при этом тип контроля (четности или нечетности) выбирается переключателем Z. На входы рассматриваемого устройства подаются различные двоичные комбинации, а состояние выходов ИМС контролируется подключенными к ним светоиндикаторами (логическими пробниками).

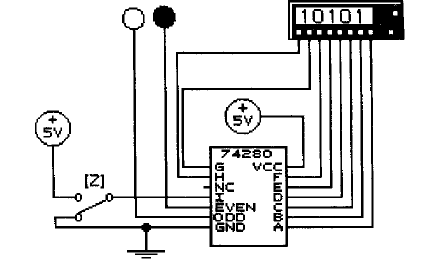


Рис. 3. Схема включения ИМС К555ИП5 (74280)

***Порядок выполнения работы***

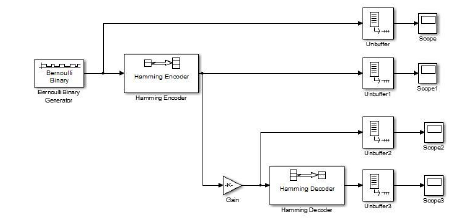
1. Создать в пакете MatLAB Simulink модель системы, осуществляющей избыточное кодирование. 

Рис. 4. Структурная схема модели.

2. Промоделировать и зарисовать осциллограммы сигналов с осциллографов, располагая их друг под другом.

3. Пояснить иллюстрирующие графики.

Контрольные вопросы:

1. Пояснить назначение узлов схемы?

2. В чем заключается принцип помехоустойчивого кодирования?

3. В чем особенность кодера Хемминга?