СОДЕРЖАНИЕ

Введение

1 Анализ способов кодирования информации

1.1 Проверка чётности

1.2 Код CRC

1.3 Код Хэмминга

1.4 Код Рида – Соломона

2 Разработка стенда контроля передаваемой информации

2.1 Разработка устройства кодирования (кодера) информации методом Хемминга

2.2 Разработка устройства декодирования (декодера) информации методом Хемминга

2.3 Реализация кодера – декодера на базе ИМС К555ВЖ1

2.3.1 Цоколёвка ИМС К555ВЖ1 (SN74LS630)

2.3.2 Разработка принципиальной схемы устройства

2.3.3 Принцип работы устройства

3. Экономическая часть

4 Охрана труда и техники безопасности

4.1 Потенциально опасные и вредные производственные факторы

4.2 Обеспечение электробезопасности

4.3 Обеспечение санитарно-гигиенических требований к помещениям учебных лабораторий

4.4 Противопожарная защита

Заключение

Список ссылок

Введение

Характерной чертой научно-технического прогресса, определяющей мощный дальнейший подъем общественного производства, является широкое внедрение электроники во все отрасли народного хозяйства. Стремительное развитие промышленности и технологий, определило дальнейшее развитие науки на несколько поколений вперёд. Одним из самых приоритетных направлений науки является микроэлектроника, позволившая достичь высочайших технологий, которые, в свою очередь, нашли широчайшее применение как в промышленности, так и в научной сфере. Стык этих двух сфер сформировал величайшее изобретение современности – цифровой электронный компьютер.

Парк компьютеров разнообразного назначения растёт стремительными темпами. В настоящее время персональный компьютер является неотъемлемой частью любого предприятия, учебных заведений, вычислительных центров, и других учреждений. С каждым днем растут объемы передаваемой и принимаемой информации. В связи с этим становится особенно актуальной проблема сохранения целостности передаваемой и обрабатываемой информации. Память компьютера время от времени может делать ошибки из-за всплесков напряжения на линии электропередачи и по другим причинам. Передача информации также сопряжена с различного рода ошибками. Чтобы бороться с такими ошибками, были разработаны специальные способы кодирования информации, позволяющие обнаружить и исправить возможные ошибки. Существует большое количество видов помехоустойчивого кодирования. Некоторые виды настолько сложны, что требуют создания специального математического аппарата, другие же, напротив, достаточно просты и понятны. Эффективность разных способов кодирования существенно различается. Изучение методов кодирования часто становится проблемой из-за излишней математизированости материала и недостаточной наглядности. Между тем, возможно самостоятельное изготовление простого электрифицированного стенда, которое не приведет к значительным материальным затратам. Это позволяет сформулировать рабочую гипотезу исследования: изготовление и использование электрифицированного учебного стенда позволит повысить наглядность работы устройств кодирования/декодирования и, как следствие, качество усвоения материала студентами, что говорит об актуальности выбранной темы.

Целью дипломного проекта является разработка относительно недорого электрифицированного стенда «Устройство кодирования – декодирования 32х разрядных слов методом Хэмминга». В ходе написания дипломного проекта использовались следующие методы: исследование проблемы, анализ возможных путей решения, проектирование и модернизация стенда.

1 Анализ способов кодирования информации

История кодирования, контролирующего ошибки, началась в 1948 г. публикацией знаменитой статьи Клода Шеннона. Шеннон показал, что с каждым каналом связано измеряемое в битах в секунду и называемое пропускной способностью канала число С, имеющее следующее значение. Если требуемая от системы связи скорость передачи информации R (измеряемая в битах в секунду) меньше С, то, используя коды, контролирующие ошибки, для данного канала можно построить такую систему связи, что вероятность ошибки на выходе будет сколь угодно мала. Основной сдвиг произошел, когда Боуз и Рой-Чоудхури и Хоквингем нашли большой класс кодов, исправляющих кратные ошибки (коды БЧХ), а Рид и Соломон нашли связанный с кодами БЧХ класс кодов для недвоичных каналов. Хотя эти коды остаются среди наиболее важных классов кодов, общая теория блоковых кодов, контролирующих ошибки, с тех пор успешно развивалась.

Код есть форма представления сообщения, не зависящая от его физической сути. Это отличает код от сигнала, который определяет физическое представление сообщения (и кода) в системе связи. На практике часто связывают абстрактную (символьную) форму кода с физическими сигналами, называя код частотным, временным, фазовым, амплитудным. Код представляют совокупностью (кодовых) символов; помехоустойчивый код позволяет обнаруживать или исправлять ошибки в совокупности кодовых символов. Если сообщения обладают внутренними корреляционными связями, т. е. если одно сообщение некоторым образом зависит от другого, как это обычно бывает при передаче текстов на естественных языках, то помехоустойчивость любого кода может быть повышена за счет статистических связей между сообщениями. Если эти связи слабые, или неизвестны, или их нельзя использовать для повышения помехоустойчивости, то в этом случае форма представления сообщения должна быть избыточной; в частности, число символов в коде сообщения увеличивают, а между кодовыми символами вводят искусственные корреляционные связи. Поэтому в некоторых случаях помехоустойчивые коды называют избыточными. Введение избыточности в код позволяет помимо обнаружения и исправления ошибок повысить энергетическую эффективность линии связи, сузить частотный спектр передаваемого сигнала, сократить время вхождения в связь путем повышения помехозащищенности тракта синхронизации, улучшить корреляционные свойства ансамбля сигналов, простыми средствами реализовать разнесенный прием. Вид помехоустойчивого кода зависит от структуры системы связи, обобщенная схема которой приведена на рис. 1.1. Рассматриваем системы связи, передающие только дискретные сообщения. В современных системах передачи дискретных сообщений последние поступают на вход системы, как правило, от нескольких источников. Даже если внешний источник один, сама система связи содержит источник сигналов служебной связи, телеуправления и телесигнализации (ТУ-ТС). Скорость поступления сообщений от разных источников может быть как одинаковой, так и различной синхронной с собственной тактовой частотой аппаратуры связи или асинхронной с ней. Блок уплотнения (БУ) объединяет сообщения, поступающие от разных источников, в единую последовательность, как правило, двоичных символов с тактовой частотой, соответствующей скорости передачи системы связи.

Рис. 1.1 — Схема системы связи

* ИИ - источник информации;
* БУ - блок уплотнения сообщений;
* КДШ, КДВ - кодеры внешний, внутренний;
* ПРШ, ПРВ - перемежители внешний, внутренний;
* М - модулятор;
* ПД - передатчик;
* ЛС - линия связи;
* ПР - приемник;
* Д - демодулятор;
* АЦП - аналого-цифровой преобразователь;
* БДС, БПС, БЛС - блоки додетекторного, последетекторного, логического сложения;
* ДПШ, ДПВ - деперемежители внешний, внутренний;
* ДКШ, ДКВ - декодер внешний, внутренний;
* БР-блок разуплотнения сообщений;
* ПИ-получатель информации;
* КОС - канал обратной связи

Если скорости поступления сообщений от источников асинхронны по отношению к собственной тактовой частоте системы связи, БУ осуществляет асинхронный ввод сообщений. Для того чтобы при временном уплотнении различить сообщения на стороне приема, БУ формирует маркер, обозначающий место первого источника в общем цифровом потоке. Маркер повторяется периодически, образуя сигнал цикловой синхронизации. Кодер вводит избыточность в передаваемый поток двоичных символов, причем кодирование сообщений в зависимости от требуемой степени повышения помехоустойчивости может выполняться поэтапно и соответственно этапам различными кодерами. Первый после БУ кодер называют внешним (КДШ), последний - внутренним (КДВ). Сформированный кодером поток символов поступает в перемежитель. Во многих случаях ошибка в одном символе кода влечет за собой ошибки и в других смежных с ним символах той же последовательности, вызывая появление пакета ошибок на входе декодера, исправляющего ошибки. Если код рассчитан на исправление m ошибок на интервале из n смежных символов, а пакет ошибок вызывает больше чем m ложных символов, ошибка декодером не будет исправлена. Перемежитель разносит во времени смежные символы исходной кодовой последовательности более чем на n символов. При деперемежении на стороне приема разнесенные символы вновь собирают вместе; одновременно ошибки в пакете будут разнесены деперемежителем во времени более чем на n символов, и соответствующий деперемежителю декодер такие разнесенные ошибки сможет исправить. Перемеженная последовательность кодированных символов поступает в общем случае в несколько ветвей разнесения, каждая из которых содержит модулятор, передатчик, линию связи и приемник. В системах с линиями радиосвязи для борьбы с замираниями и узкополосными помехами, действующими в части частотного диапазона, применяют программную (или, как ее иногда называют, псевдослучайную) перестройку рабочих частот (ППРЧ), соответствующие устройства входят в состав передатчика и приемника.

Сложение сигналов в разнесенных ветвях на стороне приема может производиться как на входе демодулятора (додетекторное сложение), так и на его выходе (последетекторное сложение). В частности, если сигналы в ветвях некогерентны, последетекторное сложение называют квадратичным. Сравнительно недавно в системах связи с кодированными сигналами стали применять логическое объединение ветвей разнесения, реализующее последетекторный автовыбор ветви с наименьшим числом ошибок. Демодулятор (Д) производит оптимальную обработку элемента сигнала, заканчивающуюся обычно интегрированием со сбросом интегратора в определенный тактовый момент времени. Тем самым демодулятор дискретизирует во времени смесь огибающей сигнала с шумом. Формирование тактовых импульсов осуществляют устройства тактовой синхронизации, входящие в состав демодулятора. Аналого-цифровой преобразователь (АЦП) на выходе демодулятора дискретизирует (квантует) смесь огибающей сигнала с шумом по уровню. При квантовании на два уровня декодируется двоичный сигнал. Максимальное число уровней квантования, как правило, не превышает 16. Обычно число уровней равно 2, 4, 8 или 16. Декодер, работающий с двоичным сигналом, называют жестким, с недвоичным - мягким. Для работы декодера необходимы специфические (групповые) тактовые импульсы, формируемые в тракте групповой синхронизации, входящем в состав декодера. Назначение декодера состоит в уменьшении числа ошибок в сообщениях, выдаваемых системой связи, путем использования избыточности, заложенной в символьный поток кодером. Часть системы связи, включающая линию (радио- или проводную), называется каналом. Часть системы от выхода модулятора до входа АЦП образует канал передачи-приема сигнала, непрерывного по уровню (но дискретного по времени). Часть системы от выхода модулятора до выхода АЦП образует канал с входным сигналом, непрерывным по уровню и времени, и с выходным дискретным сигналом. От входа модулятора до выхода АЦП имеем дискретный (по времени и уровню) канал. В двунаправленной системе связи обычно создают канал обратной связи, по которому осуществляют управление работой системы.

Схема на рис. 1.1 может видоизменяться в зависимости от конкретной реализации системы связи. В каналах действуют искажения сигналов, шумы, помехи, которые в дискретном канале проявляются в виде перехода одного значения символа в другое - ложное (событие, состоящее в появлении ошибки) или неиспользуемое (событие, которое называют стиранием). В зависимости от характера ошибок различают дискретные каналы: симметричный (все ложные значения символов равновероятны), асимметричный (некоторые ложные значения символов обладают большей вероятностью), без памяти (искажение символа не зависит статистически от искажения другого выходного символа), с памятью (искажение символа выходной последовательности зависит статистически от искажения другого символа той же последовательности), со стираниями (наряду с ошибками имеют место стирания символов).

Любой канал связи с ограниченными полосой частот, временем передачи и динамическим диапазоном (значений амплитуд) обладает конечной пропускной способностью. Теоретически пропускная способность - это максимальное число переданных двоичных единиц (бит) в единицу времени при сколь угодно малой вероятности ошибок. Реально получаемое число передаваемых бит в единицу времени называют скоростью передачи. При неограниченно малой вероятности ошибок скорость передачи всегда меньше пропускной способности. В канале с ошибками максимальное значение скорости получают путем использования помехоустойчивого кодирования. Последнее требует введения избыточности в передаваемый сигнал: по времени, частоте или амплитуде. Если код согласован с каналом, т. е. код позволяет исправлять наиболее вероятные ошибки, введенная избыточность становится оправданной. Если код не согласован с каналом, ошибки могут быть не только не исправлены, но и размножены кодом. В этом случае применение помехоустойчивого кодирования принесет не пользу, а вред. Для согласования кода с каналом связи необходимо иметь максимальный объем сведений о возможных мешающих влияниях в каналах.

Рис. 1.2 — Классификация помехоустойчивых кодов

К настоящему времени разработано иного различных помехоустойчивых кодов, отличающихся друг от друга основанием, расстоянием, избыточностью, структурой, функциональным назначением, энергетической эффективностью, корреляционными свойствами, алгоритмами кодирования и декодирования, формой частотного спектра. На рис 1.2 приведены типы кодов, различающиеся по особенностям структуры, функциональному назначению, физическим свойствам кода как сигнала. Наиболее важный подкласс непрерывных кодов образуют сверточные коды, отличающиеся от других непрерывных кодов методом построения и более широкой областью применения. В общем случае чем длиннее код при фиксированной избыточности, тем больше расстояние и тем выше помехоустойчивость кода. Однако длинные коды сложно реализуются. Составные коды дают компромиссное решение задачи, из них основное значение имеют каскадные коды и коды произведения. Как правило, каскадный код состоит из двух ступеней (каскадов): внутренней и внешней. По линии связи сигналы передают внутренним кодом nвт, символьные слова которого являются символами внешнего кода длины nвш. Основание внешнего кода равно qвтk. Коды произведения строят в виде матрицы, в которой строки суть слова одного кода, а столбцы - того же или другого кода. При формировании каскадного кода входную информационную последовательность символов разбивают на блоки по kвт символов в каждом, каждый блок сопоставляют с информационным символом внешнего кода из алфавита, содержащего qвтk значений символов. Затем kвш информационных символов внешнего кода преобразуют в блоки из nвш символов внешнего кода и, наконец, блоки из kвт информационных символов внутреннего кода преобразуют в блоки из nвт символов внутреннего кода. Возможны различные варианты: внешний и внутренний коды - блочные, внешний блочный - внутренний сверточный, внешний сверточный - внутренний блочный, внешний и внутренний сверточные.

Один из наиболее распространенных методов формирования кода произведения заключается в последовательной записи по k1 символов входной информационной последовательности в k2 строк матрицы (например, в ячейки памяти ОЗУ), добавлении избыточных символов по n1-k1 в каждую строку и по n2-k2 в каждый столбец, после чего в последовательность символов кода считывают по строкам или столбцам из матрицы. Физическим аналогом кода произведения является, в частности, частотно-временной код, у которого строки располагаются вдоль оси времени, а столбцы - по оси частот.

Параметры составных кодов: каскадных - n=nвшnвт, k=kвшkвт, d=dвшdвт; произведения - n=n1n2, k=k1k2, d=d1d2. Производные коды строят на основе некоторого исходного кода, к которому либо добавляют символы, увеличивая расстояние (расширенный код), либо сокращают часть информационных символов без изменения расстояния (укороченный код), либо выбрасывают (выкалывают) некоторые символы (выколотый, или перфорированный код). Код Хэмминга дает пример процедуры расширения, увеличивающей расстояние кода с 3 до 4. Необходимость в выкалывании возникает в результате построения на основе исходного кода другого, менее мощного, более короткого кода с тем же расстоянием. При более широкой трактовке термина "производный код" к этому классу можно отнести все коды, полученные из исходного добавлением или исключением как символов, так и слов.

Формально деление кодов на двоичные и недвоичные носит искусственный характер; по аналогии следует выделять троичные, четверичные и другие коды большего основания. Оправдывается такое деление усложнением алгоритмов построения, кодирования и декодирования недвоичных кодов. При прочих равных условиях желательно, чтобы информационные и избыточные символы располагались отдельно. В систематических кодах это условие выполняется. В циклических кодах каждое слово содержит все свои циклические перестановки. Все n циклических перестановок (слова длины n) образуют цикл. В квазициклических кодах цикл образуется на числе символов n-1 или, реже, n 2. Циклические коды важны как с точки зрения математического описания, так и для построения и реализации кода.

Ошибки в каналах связи имеют самое различное распределение, однако для выбора помехоустойчивого кода целесообразно разделить все возможные конфигурации ошибок на независимые (некоррелированные) и пакеты (коррелированные ошибки). На практике приходится учитывать качество интервалов между пакетами: они могут быть свободными от ошибок или же содержать случайные независимые ошибки. Под корреляционными подразумевают коды, обладающие хорошими корреляционными свойствами, важными при передаче сигналов вхождения в связь, для повышения защищенности от некоторых видов помех, извлечения сигналов из интенсивных шумов, обеспечения многостанционного доступа, построения асинхронно-адресных систем связи. Корреляционные коды включают в себя пары противоположных сигналов с хорошей функцией автокорреляции (метод внутриимпульсной модуляции), импульсно-интервальные коды, имеющие на фиксированном интервале времени постоянное для всех слов кода число импульсов с неперекрывающимися (при любом взаимном сдвиге слов во времени) значениями интервалов между импульсами, ансамбли сигналов с хорошими взаимокорреляционными свойствами.

Особый класс образуют частотно-компактные коды, предназначенные для сосредоточения энергии сигнала в возможно более узкой полосе частот. Столь общая постановка задачи понимается в различных системах связи по-разному: в проводных линиях и линейных трактах, содержащих полосно-ограничивающие фильтры с крутыми фронтами, необходимо основную энергию сигналa "отодвинуть" от крайних частот к центру полосы пропускания целью уменьшения межсимвольных искажений; в сетях радиосвязи с жесткими ограничениями по электромагнитной совместимости радиосредств от кода требуется значительно (на десятки децибел) уменьшить уровень внеполосных излучений. Построение кодирование и декодирование частотно-компактных кодов существенно зависят от метода модуляции.

Арифметические коды служат для борьбы с ошибками при выполнении арифметических операций в процессоре ЭВМ.

Также выделены еще два типа кодов: блоковые и древовидные. Определяющее различие между кодерами для кодов этих двух типов состоит в наличии или отсутствии памяти. Кодер для блокового кода является устройством без памяти, отображающим последовательности из k входных символов в последовательности из n выходных символов. Термин "без памяти" указывает, что каждый блок из n символов зависит только от соответствующего блока из k символов и не зависит от других блоков. Это не означает, что кодер не содержит элементов памяти. Важными параметрами блокового кода являются n, k, R=k/n и dmin. На практике значения k лежат между 3 и несколькими сотнями, a R= =1/4 ...7/8. Значения, лежащие вне этих пределов, являются возможными, но часто приводят к некоторым практическим трудностям. Входные и выходные последовательности обычно состоят из двоичных символов, но иногда могут состоять из элементов некоторого алфавита большего объема. Кодер для древовидного кода является устройством с памятью, в которое поступают наборы из m двоичных входных символов, а на выходе появляются наборы из n двоичных выходных символов. Каждый набор n выходных символов зависит от текущего входного набора и от v предыдущих входных символов. Таким образом, память кодера должна содержать v+m входных символов. Параметр v+m часто называют длиной кодового ограничения данного кода и обозначают k=v+m (не следует путать с параметром k для блокового кода). Длиной кодового ограничения будем называть величину v. Типичные значения параметров древовидных кодов таковы: m, n=1...8, R= 1/4... 7/8, v=2...60.

При другом подходе коды можно разделить на линейные и нелинейные. Линейные коды образуют векторное пространство и обладают следующим важным свойством: два кодовых слова можно сложить, используя подходящее определение суммы, и получить третье кодовое слово. В случае обычных двоичных кодов эта операция является посимвольным сложением двух кодовых слов по модулю 2 (т. е. 1+1=0, 1+0=1, 0+0=0). Это свойство приводит к двум важным следствиям. Первое из них состоит в том, что линейность существенно упрощает процедуры кодирования и декодирования, позволяя выразить каждое кодовое слово в виде "линейной" комбинации небольшого числа выделенных кодовых слов, так называемых базисных векторов. Второе свойство состоит в том, что линейность существенно упрощает задачу вычисления параметров кода, поскольку расстояние между двумя кодовыми словами при этом эквивалентно расстоянию между кодовым словом, состоящим целиком из нулей, и некоторым другим кодовым словом. Таким образом, при вычислении параметров линейного кода достаточно рассмотреть, что происходит при передаче кодового слова, состоящего целиком из нулей. Вычисление параметров упрощается еще и потому, что расстояние Хемминга между данным кодовым словом и нулевым кодовым словом равно числу ненулевых элементов данного кoдового слова. Это число часто называют весом Хемминга данного слова, и список, содержащий число кодовых слов каждого веса, можно использовать для вычисления характеристик кода с помощью аддитивной границы. Такой список называют спектром кода. Линейные коды отличаются от нелинейных замкнутостью кодового множества относительно некоторого линейного оператора, например сложения или умножения слов кода, рассматриваемых как векторы пространства, состоящего из кодовых слов - векторов. Линейность кода упрощает его построение и реализацию. При большой длине практически могут быть использованы только линейные коды. Вместе с тем часто нелинейные коды обладают лучшими параметрами по сравнению с линейными. Для относительно коротких кодов сложность построения и реализации линейных и нелинейных кодов примерно одинакова. Как линейные, так и нелинейные коды образуют обширные классы, содержащие много различных конкретных видов помехоустойчивых кодов. Среди линейных блочных наибольшее значение имеют коды с одной проверкой на четность, M-коды (симплексные), ортогональные, биортогональные, Хэмминга, Боуза-Чоудхури-Хоквингема, Голея, квадратично-вычетные (KB), Рида-Соломона. К нелинейным относят коды с контрольной суммой, инверсные, Нордстрома-Робинсона (HP), с постоянным весом, перестановочные с повторением и без повторения символов (полные коды ортогональных таблиц, проективных групп, групп Матье и других групп перестановок). Почти все схемы кодирования, применяемые на практике, основаны на линейных кодах. Двойные линейные блоковые коды часто называют групповыми кодами, поскольку кодовые слова образуют математическую структуру, называемую группой. Линейные древовидные коды обычно называют сверточными кодами, поскольку операцию кодирования можно рассматривать как дискретную свертку входной последовательности с импульсным откликом кодера.

Наконец, коды можно разбить на коды, исправляющие случайные ошибки, и коды, исправляющие пакеты ошибок. В основном будем иметь дело с кодами, предназначенными для исправления случайных, или независимых, ошибок. Для исправления пакетов ошибок было создано много кодов, имеющих хорошие параметры. Однако при наличии пачек ошибок часто оказывается более выгодным использовать коды, исправляющие случайные ошибки, вместе с устройством перемежения восстановления. Такой подход включает в себя процедуру перемешивания порядка символов в закодированной последовательности перед передачей и восстановлением исходного порядка символов после приема с тем, чтобы рандомизировать ошибки, объединенные в пакеты.

1.1 Проверка чётности

Контроль четности или коррекция ошибок (ECC (код исправления ошибок)) используется в основном только в жизненно важных компьютерных системах, где недопустима даже одна ошибка в несколько десятилетий. Проверка четности – довольно простой метод обнаружения ошибок памяти, без возможности восстановления. Каждый байт данных связан с одним битом четности или так называемым паритетным битом. Этот бит устанавливается во время записи, и затем рассчитывается и сравнивается во время чтения. Изменение состояния этого бита говорит о возникшей ошибке. Этот метод ограничен определением изменения состояния одиночного бита в байте. В случае изменения состояния двух битов, возможна ситуация, когда вычисление паритетного бита совпадет с записанным. В этом случае система не определит ошибку, и произойдет экстренная остановка системы. Так как приблизительно 90% всех нерегулярных ошибок происходит именно с одиночным разрядом, проверки четности бывает достаточно для большинства ситуаций. К сожалению необходимость в дополнительных вычислениях паритетного бита требует некоторых затрат процессорного времени, что несколько снижает производительность всей системы. Более интересным методом проверки ошибок работы памяти является т.н. ECC или коррекция ошибок или код исправления ошибок. Этот метод включает определение ошибки не только в одиночном разряде, но и двух, трех и четырех разрядах. Кроме того ECC может также исправлять ошибку в одиночном разряде. ECC может быть реализован или на модуле памяти (ECC – on – Simm, или EOS) или в чипсете. Однако модули EOS встречаются крайне редко. ECC основывается на алгоритме «хешинга», который работает одновременно с 8 байтами (64 бита), и размещает результат в 8-ми разрядное ECC слово. Во время считывания результат ECC слова сравнивается с рассчитанным, подобно тому, как происходит в методе проверки четности. Основное различие состоит в том, что в проверке четности каждый бит связан с одним байтом, в то время как ECC слово связана со всеми 8 байтами. Это означает, что разрядные значения для ECC не будут теми же, что и индивидуальные биты для проверки четности для тех же восьми байтов, поэтому модули ECC не могут использоваться в режиме четности (однако, паритетные модули, могут использоваться в режиме ECC, как описано ниже). ECC модули могут использоваться на не паритетный и на ECC/ non-ECC платах. Модуль ECC не может использоваться в режиме проверки четности. Причина этого является схемотехническая реализация модуля ECC. Он не может устанавливать отдельные биты, поэтому чипсет не будет записывать правильные данные в ECC слово.

1.2 Код CRC

Методы обнаружения ошибок предназначены для выявления повреждений со общений при их передаче по зашумленным каналам (вносящих эти ошибки). Для этого передающее устройство создает некоторое число, называемое контрольной суммой и являющееся функцией сообщения, и добавляет его к этому сообщению. Приемное устройство, используя тот же самый алгоритм, рассчитывает контрольную сумму принятого сообщения и сравнивает ее с переданным значением. Например, если для расчета контрольной суммы используем простое сложение байтов сообщения по модулю 256, то может возникнуть примерно следующая ситуация. (Все числа примера десятичные.)

Сообщение: 6 23 4

Сообщение с контрольной суммой: 6 23 4 33

Сообщение после передачи: 6 27 4 33

Как видно, второй байт сообщения при передаче оказался измененным с 23 на 27. Приемник может обнаружить ошибку, сравнивая переданную контрольную сумму (33) с рассчитанной им самим: 6 + 27 + 4 = 37. Если при правильной передаче сообщения окажется поврежденной сама контрольная сумма, то такое сообщение будет неверно интерпретировано, как искаженное. Однако, это не самая плохая ситуация. Более опасно, одновременное повреждение сообщения и контрольной суммы таким образом, что все сообщение можно посчитать достоверным. К сожалению, исключить такую ситуацию невозможно, и лучшее, чего можно добиться, это снизить вероятность ее появления, увеличивая количество информации в контрольной сумме (например, расширив ее с одного до 2 байт).

Ошибки иного рода возникают при сложных преобразованиях сообщения для удаления из него избыточной информации. Однако, данная статья посвящена только расчетам CRC, которые относятся к классу алгоритмов, не затрагивающих само го сообщения и лишь добавляющих в его конце контрольную сумму:

<исходное неизмененное сообщение> <контрольная сумма>

Требования сложности

Выше показано, что повреждение сообщения может быть обнаружено, используя в качестве алгоритма контроля простое суммирование байтов сообщения по модулю 256:

Сообщение: 6 23 4

Сообщение с контрольной суммой: 6 23 4 33

Сообщение после передачи: 6 27 4 33

Недостаток этого алгоритма в том, что он слишком прост. Если произойдет несколько искажений, то в 1 случае из 256 не сможем их обнаружить. Например:

Сообщение: 6 23 4

Сообщение с контрольной суммой: 6 23 4 33

Сообщение после передачи: 8 20 5 33

Для повышения надежности можно было бы изменить размер регистра с 8 битного на 16 битный (то есть суммировать по модулю 65536 вместо модуля 256), что скорее всего снизит вероятность ошибки с 1/256 до 1/65536. Хотя это и неплохая идея, однако, она имеет тот недостаток, что применяемая формула расчета не "случайна" в должной степени — каждый суммируемый байт оказывает влияние лишь на один байт суммирующего регистра, при этом ширина самого регистра не имеет никакого значения. Например, во втором случае суммирующий регистр мог бы иметь ширину хоть мегабайт, однако ошибка все равно не была бы обнаружена. Проблема может быть решена лишь заменой простого суммирования более сложной функцией, чтобы каждый новый байт оказывал влияние на весь регистр контрольной суммы.

Таким образом, сформулировано 2 требования для формирования надежной контрольной суммы:

Ширина: Размер регистра для вычислений должен обеспечивать изначальную низкую вероятность ошибки (например, 32 байтный регистр обеспечивает вероятность ошибки 1/232).

Случайность: Необходим такой алгоритм расчета, когда каждый новый байт может оказать влияние на любые биты регистра.

Обратим внимание, что термин "контрольная сумма" изначально описывал достаточно простые суммирующие алгоритмы, однако, в настоящее время он используется в более широком смысле для обозначения сложных алгоритмов рас чета, таких как CRC. Алгоритмы CRC, которые и будем рассматривать в дальнейшем, очень хорошо удовлетворяют второму условию и, кроме того, могут быть адаптированы для работы с различной шириной контрольной суммы.

Контрольная сумма (Checksum) – Число, которое является функцией некоторого сообщения. Буквальная интерпретация данного слова указывает на то, что выполняется простое суммирование байтов сообщения, что, по видимому, и делалось в ран них реализациях расчетов. Однако, на сегодняшний момент, несмотря на использование более сложных схем, данный термин все имеет широкое применение.

Если сложение, очевидно, не пригодно для формирования эффективной контрольной суммы, то таким действием вполне может оказаться деление при условии, что делитель имеет ширину регистра контрольной суммы.

Основная идея алгоритма CRC состоит в представлении сообщения виде огромного двоичного числа, делении его на другое фиксированное двоичное число и использовании остатка этого деления в качестве контрольной суммы. Получив сообщение, приемник может выполнить аналогичное действие и сравнить полученный остаток с "контрольной суммой" (переданным остатком).

Пример:

Предположим, что сообщение состоит из 2 байт (6, 23), как и в предыдущем примере. Их можно рассматривать, как шестнадцатеричное число 0167h, или как двоичное число 0000 0110 0001 0111. Предположим, что ширина регистра контрольной суммы составляет 1 байт, а в качестве делителя используется 1001, тогда сама контрольная сумма будет равна остатку от деления 0000 0110 0001 0111 на 1001. Хотя в данной ситуации деление может быть выполнено с использованием стандартных 32 битных регистров, в общем случае это не верно. Поэтому воспользуемся делением "в столбик". Только на этот раз, оно будет выполняться в двоичной системе счисления:

...0000010101101 = 00AD = 173 =

----\_---\_---\_---\_

9= 1001 ) 0000011000010111 = 0617 = 1559 =

0000.,,....,.,,,

----.,,....,.,,,

0000,,....,.,,,

0000,,....,.,,,

----,,....,.,,,

0001,....,.,,,

0000,....,.,,,

----,....,.,,,

0011....,.,,,

0000....,.,,,

----....,.,,,

0110...,.,,,

0000...,.,,,

----...,.,,,

1100..,.,,,

1001..,.,,,

====..,.,,,

0110.,.,,,

0000.,.,,,

----.,.,,,

1100,.,,,

1001,.,,,

====,.,,,

0111.,,,

0000.,,,

----.,,,

1110,,,

1001,,,

====,,,

1011,,

1001,,

====,,

0101,

0000,

----

1011

1001

====

0010 = 02 = 2

В десятичном виде это будет звучать так: "частное от деления 1559 на 9 равно 173 и 2 в остатке".

Хотя влияние каждого бита исходного сообщения на частное не столь существенно, однако 4 битный остаток во время вычислений может радикально измениться, и чем больше байтов имеется в исходной сообщении (в делимом), тем сильнее меняется каждый раз величина остатка. Вот почему деление оказывается применимым там, где обычное сложение работать отказывается.

В нашем случае передача сообщения вместе с 4 битной контрольной суммой выглядела бы (в шестнадцатеричном виде) следующим образом: 06172, где 0617 – это само сообщение, а 2 – контрольная сумма. Приемник, получив сообщение, мог бы выполнить аналогичное деление и проверить, равен ли остаток переданному значению (2).

Хотя арифметическое деление, описанное выше, очень похоже на схему расчета контрольной суммы, называемой CRC, сам же алгоритм CRC несколько более сложен, и, чтобы понять его, необходимо окунуться в теорию целых чисел.

Полином (Polynomial) – Полином является делителем CRC алгоритма.

Все CRC алгоритмы основаны на полиномиальных вычислениях, и для любого алгоритма CRC можно указать, какой полином он использует.

Вместо представления делителя, делимого (сообщения), частного и остатка в виде положительных целых чисел можно представить их в виде полиномов с двоичными коэффициентами или в виде строки бит, каждый из которых является коэффициентом полинома. Например, десятичное число 23 в шестнадцатеричной системе счисления имеет вид 17, а в двоичном – 10111, что совпадает с полиномом:

1\*x^4 + 0\*x^3 + 1\*x^2 + 1\*x^1 + 1\*x^0

или, упрощенно:

x^4 + x^2 + x^1 + x^0

И сообщение, и делитель могут быть представлены в виде полиномов, с которыми как и раньше можно выполнять любые арифметические действия. Предположим, что хотим перемножить, например, 1101 и 1011. Это можно выполнить, как умножение полиномов:

(x^3 + x^2 + x^0)(x^3 + x^1 + x^0)

= (x^6 + x^4 + x^3

+ x^5 + x^3 + x^2

+ x^3 + x^1 + x^0) = x^6 + x^5 + x^4 + 3\*x^3 + x^2 + x^1 + x^0

Теперь для получения правильного ответа необходимо указать, что Х равен 2, и выполнить перенос бита от члена 3\*x^3.

В результате получим:

x^7 + x^3 + x^2 + x^1 + x^0

Все это очень похоже на обычную арифметику, с той лишь разницей, что основание у нас лишь предполагается, а не строго задано. Если "X" неизвестен, то не можем выполнить перенос. Неизвестно, что 3\*x^3 – это то же самое, что и x^4 + x^3, так как не знаем, что X=2. В полиномиальной арифметике связи между коэффициентами не установлены, и, поэтому, коэффициенты при каждом члене полинома становятся строго типизированными — коэффициент при x^2 имеет иной тип, чем при x^3.

Если коэффициенты каждого члена полинома совершенно изолированы друг от друга, то можно работать с любыми видами полиномиальной арифметики, просто меняя правила, по которым коэффициенты работают. Одна из таких схем для нас чрезвычайно интересна, а именно, когда коэффициенты складываются по модулю 2 без переноса – то есть коэффициенты могут иметь значения лишь 0 или 1, перенос не учитывается. Это называется "полиномиальная арифметика по модулю 2".

Возвращаясь к предыдущему примеру:

(x^3 + x^2 + x^0)(x^3 + x^1 + x^0)

= (x^6 + x^4 + x^3

+ x^5 + x^3 + x^2

+ x^3 + x^1 + x^0)

= x^6 + x^5 + x^4 + 3\*x^3 + x^2 + x^1 + x^0

По правилам обычной арифметики, коэффициент члена 3\*x^3 распределяется по другим членам полинома, используя механизм переноса и предполагая, что X = 2. В "полиномиальной арифметике по модулю 2" не известно, чему равен "X", переносов здесь не существует, а все коэффициенты рассчитываются по модулю 2. В результате получаем:

= x^6 + x^5 + x^4 + x^3 + x^2 + x^1 + x^0

Таким образом, полиномиальная арифметика по модулю 2 – это фактически двоичная арифметика по модулю 2 без учета переносов. Хотя полиномы имеют удобные математические средства для анализа CRC алгоритмов и алгоритмов коррекции ошибок, для упрощения дальнейшего обсуждения они будут заменены, на непосредственные арифметические действия в системе, с которой они изоморфны, а именно в системе двоичной арифметики без переносов.

Двоичная арифметика без учета переносов

Оставим полиномы вне поля нашего внимания, и сфокусируем его на обычной арифметике, так как действия, выполняемые во время вычисления CRC, являются арифметическими операциями без учета переносов. Эта арифметическая система является ключевой частью расчетов CRC.

Сложение двух чисел в CRC арифметике полностью аналогично обычному арифметическому действию за исключением отсутствия переносов из разряда в разряд. Это означает, что каждая пара битов определяет результат своего разряда вне зависимости от результатов других пар. Например:

10011011

+11001010

--------

01010001

Для каждой пары битов возможны 4 варианта:

0+0=0

0+1=1

1+0=1

1+1=0 ( )

То же самое справедливо и для вычитания:

10011011

-11001010

--------

01010001

когда имеются также 4 возможные комбинации:

0-0=0

0-1=1 ( )

1-0=1

1-1=0

Фактически, как операция сложения, так и операция вычитания в CRC арифметике идентичны операции "Исключающее ИЛИ" (eXclusive OR – XOR), что позволяет заменить 2 операции первого уровня (сложение и вычитание) одним действием, которое, одновременно, оказывается инверсным самому себе. Очень удобное свойство такой арифметики.

Сгруппировав сложение и вычитание в одно единое действие, CRC арифметика исключает из поля своего внимания все величины, лежащие за пределами самого старшего своего бита. Хотя совершенно ясно, что значение 1010 больше, чем 10, это оказывается не так, когда говорят, что 1010 должно быть больше, чем 1001.

Чтобы понять почему, попытайтесь получить из 1010 значение 1001, отняв или прибавив к нему одну и ту же величину:

1001 = 1010 + 0011

1001 = 1010 – 0011

Это сводит на нет всякое представление о порядке. Определив действие сложения, перейдем к умножению и делению. Умножение, как и в обычной арифметике, считается суммой значений первого сомножителя, сдвинутых в соответствии со значением второго сомножителя.

1101x

1011

----

1101

1101.

0000..

1101...

-------

1111111

Обратим внимание: при суммировании используется CRC сложение.

Деление несколько сложнее, так как требуется знать, когда "одно число превращается в другое ". Чтобы сделать это, необходимо ввести слабое понятие размерности: число X больше или равно Y, если позиция самого старшего единичного бита числа X более значима (или соответствует) позиции самого старшего единичного бита числа Y. Ниже приведен пример деления:

1100001010

————————————————

10011 ) 11010110110000

10011,,.,,....

-----,,.,,....

10011,.,,....

10011,.,,....

-----,.,,....

00001.,,....

00000.,,....

-----.,,....

00010,,....

00000,,....

-----,,....

00101,....

00000,....

-----,....

01011....

00000....

-----....

10110...

10011...

-----...

01010..

00000..

-----..

10100.

10011.

-----.

01110

00000

-----

1110 =

Это так и есть. Однако, прежде, чем идти дальше, стоит еще немного задержаться на этих действиях.

Зная, что действия сложения и вычитания в нашей арифметике – это одно и то же. В таком случае, A+0=A и A-0=A.

Если число A получено умножением числа B, то в CRC арифметике это означает, что существует возможность сконструировать число A из нуля, применяя операцию XOR к числу B, сдвинутому, на различное количество позиций. Например, если A равно 0111010110, а B – 11, то может сконструировать A из B следующим способом:

0111010110

= .......11.

+ ....11....

+ ...11.....

11.......

Однако, если бы A было бы равно 0111010111, то не удалось составить его с помощью различных сдвигов числа поэтому говорят, что в CRC арифметике оно не делится на B.

Таким образом, видно, что CRC арифметика сводится главным образом к операции "Исключающее ИЛИ" некоторого значения при различных величинах сдвига.

Определив все правила CRC арифметики, можно теперь охарактеризовать вычисление CRC как простое деление, чем оно фактически и является. Чтобы выполнить вычисление CRC, необходимо выбрать делитель. Говоря математическим языком, делитель называется генераторным полиномом (generator polinomial), или просто полиномом, и это ключевое слово любого CRC алгоритма. Для простоты будет называть CRC полином просто полиномом. Можно выбрать и использовать в CRC любой полином. Степень полинома W (Width – ширина) (позиция самого старшего единичного бита) чрезвычайно важна, так как от нее зависят все остальные расчеты. Обычно выбирается степень 16 ил 32, так как это облегчает реализацию алгоритма на со временных компьютерах. Степень полинома – это действительная позиция старшего бита, например, степень полинома 10011 равна 4, а не 5.

Степень (Width) – Степень (или ширина) CRC алгоритма соответствует степе ни используемого полинома (или длине полинома минус единица). Например, если используется полином 11010, то степень алгоритма равна 4. Обычно используется степень, кратная 8

Выбрав полином приступим к расчетам. Это будет простое деление (в терминах CRC арифметики) сообщения на наш полином. Единственное, что надо будет сделать до начала работы, так это дополнить сообщение W нулевыми битами. Итак, начнем.

Исходное сообщение: 1101011011

Полином: 10011

Сообщение, дополненное W битами: 11010110110000

Теперь просто поделим сообщение на полином, используя правила CRC арифметики. Ранее уже рассматривалось это действие.

1100001010 = ( )

---------------

10011 ) 11010110110000 = выровненное сообщение (1101011011 + 0000)

=# ,,.,.,,.,,....

10011,,.,,....

-----,,.,,....

10011,.,,....

10011,.,,....

-----,.,,....

00001.,,....

00000.,,....

-----.,,....

00010,,....

00000,,....

-----,,....

00101,....

00000,....

-----,....

01011....

00000....

-----....

10110...

10011...

-----...

01010..

00000..

-----..

10100.

10011.

-----.

01110

00000

-----

1110

В результате получаем частное, которое нас не интересует и, поэтому, отбрасывается за ненадобностью, и остаток, который и используется в качестве контрольной суммы. Расчет закончен. Как правило, контрольная сумма добавляется к сообщению и вместе с ним передается по каналам связи. В нашем случае будет передано следующее сообщение:

11010110111110

На другом конце канала приемник может сделать одно из равноценных действий:

1. Выделить текст собственно сообщения, вычислить для него контрольную сумму (не забыв при этом дополнить сообщение W битами), и сравнить ее с переданной.

2. Вычислить контрольную сумму для всего переданного сообщения (без добавления нулей), и посмотреть, получится ли в результате нулевой остаток.

Оба эти варианта совершенно равноправны.

Таким образом, при вычислении CRC необходимо выполнить следующие действия:

1. Выбрать степень полинома W и полином G (степени W).

2. Добавить к сообщению W нулевых битов. Назовем полученную строку M'.

3. Поделим M' на G с использованием правил CRC арифметики. Полученный остаток и будет контрольной суммой.

Надо отметить, что переданное сообщение T является произведением полинома. Чтобы понять это, обратите внимание, что 1) последние W бит сообщения – это остаток от деления дополненного нулями исходного сообщения на выбранный полином, и 2) сложение равносильно вычитанию, поэтому прибавление остатка дополняет значение сообщения до следующего полного произведения. Теперь смотрите, если сообщение при передаче было повреждено, то получим со общение T E, где E – это вектор ошибки, а ' ' – это CRC сложение (или операция XOR). Получив сообщение, приемник делит T E на G. Так как T mod G = 0, (T E) mod G = E mod G. Следовательно, качество полинома, который выбираем для перехвата некоторых определенных видов ошибок, будет определяться набором произведений G, так как в случае, когда E также является произведением G, такая ошибка выявлена не будет. Следовательно, наша задача состоит в том, чтобы найти такие классы G, произведения которых будут как можно меньше похожи на шумы в канале передачи (которые и вызывают повреждение сообщения). Рассмотрим, какие типы шумов в канале передачи можем ожидать.

Однобитовые ошибки. Ошибка такого рода означает, что E=1000...0000. Можем гарантировать, что ошибки этого класса всегда будет распознаны при условии, что в G по крайней мере 2 бита установлены в "1". Любое произведение G может быть сконструировано операциями сдвига и сложения, и, в тоже время, невозможно получить значение с 1 единичным битом сдвигая и складывая величину, имеющую более 1 единичного бита, так как в результате всегда будет присутствовать по крайней мере 2 бита.

Двух битовые ошибки. Для обнаружения любых ошибок вида 100...000100...000 (то есть когда E содержит по крайней мере 2 единичных бита) необходимо выбрать такое G, которые бы не имело множителей 11, 101, 1001, 10001, и так далее. Такие полиномы должны существовать - полином с единичными битами в позициях 15, 14 и 1, который не может быть делителем ни одно числа меньше 1...1, где «...» 32767 нулей.

Ошибки с нечетным количеством бит. Может перехватить любые повреждения, когда E имеет нечетное число бит, выбрав полином G таким, чтобы он имел четное количество бит. 1) CRC умножение является простой операцией XOR постоянного регистрового значения с различными смещениями; 2) XOR – это всего-навсего операция переключения битов; и 3) если применяется в регистре операция XOR к величине с четным числом битов, четность количества единичные битов в регистре останется неизменной. На пример, начнем с E=111 и попытаемся сбросить все 3 бита в "0" последовательным выполнением операции XOR с величиной 11 и одним из 2 вариантов сдвигов (то есть, "E=E XOR 011" и "E=E XOR 110"). Это аналогично задаче о перевертывании стаканов, когда за одно действие можно перевернуть одновременно любые два стакана. Большинство популярных CRC полиномов содержат четное количество единичных битов.

Пакетные ошибки. Пакетная ошибка выглядит как E=000...000111...11110000...00, то есть E состоит из нулей за исключением группы единиц где то в середине. Эту величину можно преобразовать в E=(10000...00)(1111111...111), где имеется z нулей в левой части и n единиц в правой. Для выявления этих ошибок необходимо установить младший бит G в 1. При этом необходимо, чтобы левая часть не была множителем G. При этом всегда, пока G шире правой части, ошибка всегда будет распознана.

Вот несколько популярных полиномов:

16 битные: (16,12,5,0) [стандарт «X25»]

(16,15,2,0) ["CRC 16"]

32 битные: (32,26,23,22,16,12,11,10,8,7,5,4,2,1,0) [Ethernet]

1.3 Код Хэмминга

Предположим, что слово состоит из m битов данных, к которым прибавляем г дополнительных битов (контрольных разрядов). Пусть общая длина слова будет n (то есть n-m+г). п-битную единицу, содержащую m битов данных и г контрольных разрядов, часто называют кодированным словом. Для любых двух кодированных слов, например 10001001 и 10110001, можно определить, сколько соответствующих битов в них различается. В данном примере таких бита три. Чтобы определить количество различающихся битов, нужно над двумя кодированными словами произвести логическую операцию ИСКЛЮЧАЮЩЕЕ ИЛИ и сосчитать число битов со значением 1 в полученном результате. Число битовых позиций, по которым различаются два слова, называется интервалом Хэмминга. Если интервал Хэмминга для двух слов равен d, это значит, что достаточно d битовых ошибок, чтобы превратить одно слово в другое. Например, интервал Хэмминга кодированных слов 11110001 и 00110000 равен 3, поскольку для превращения первого слова во второе достаточно 3 ошибок в битах. Память состоит из m-битных слов, и следовательно, существует 2m вариантов сочетания битов. Кодированные слова состоят из n битов, но из-за способа подсчета контрольных разрядов допустимы только 2т из 2" кодированных слов. Если в памяти обнаруживается недопустимое кодированное слово, компьютер знает, что произошла ошибка. При наличии алгоритма для подсчета контрольных разрядов можно составить полный список допустимых кодированных слов и из этого списка найти два слова, для которых интервал Хэмминга будет минимальным. Это интервал Хэмминга полного кода. Свойства проверки и исправления ошибок определенного кода зависят от его интервала Хэмминга. .Чтобы обнаружить d ошибок в битах, необходим код с интервалом d+1, поскольку d ошибок не могут изменить одно допустимое кодированное слово на другое допустимое кодированное слово. Соответственно, чтобы исправить d ошибок в битах, необходим код с интервалом 2d+l, поскольку в этом случае допустимые кодированные слова так сильно отличаются друг от друга, что даже если произойдет d изменений, изначальное кодированное слово будет ближе к ошибочному, чем любое другое кодированное слово, поэтому его без труда можно будет определить.

В качестве простого примера кода с обнаружением ошибок рассмотрим код, в котором к данным присоединяется один бит четности. Бит четности выбирается таким образом, что число битов со значением 1 в кодированном слове четное (или нечетное). Интервал этого кода равен 2, поскольку любая ошибка в битах приводит к кодированному слову с неправильной четностью. Другими словами, достаточно двух ошибок в битах для перехода от одного допустимого кодированного слова к другому допустимому слову. Такой код может использоваться для обнаружения одиночных ошибок. Если из памяти считывается слово, содержащее неверную четность, поступает сигнал об ошибке. Программа не сможет продолжаться, но зато не будет неверных результатов.

В качестве простого примера кода с исправлением ошибок рассмотрим код с четырьмя допустимыми кодированными словами:

0000000000, 0000011111, 1111100000 и 1111111111

Интервал этого кода равен 5. Это значит, что он может исправлять двойные ошибки. Если появляется кодированное слово 0000000111, компьютер знает, что изначальное слово должно быть 0000011111 (если произошло не более двух ошибок). При наличии трех ошибок, если, например, слово 0000000000 изменилось на 0000000111, этот метод недопустим. Представим, что хотим разработать код с m битами данных и г контрольных разрядов, который позволил бы исправлять все ошибки в битах. Каждое из 2r допустимых слов имеет n недопустимых кодированных слов, которые отличаются от допустимого одним битом. Они образуются инвертированием каждого из n битов в n-битном кодированном слове. Следовательно, каждое из 2r допустимых слов требует п+1 возможных сочетаний битов, приписываемых этому слову (п возможных ошибочных вариантов и один правильный). Поскольку общее число различных сочетаний битов равно 2n, то (n+l)2m<2n.

Так как n=m+r, следовательно, (m+r+1)<2г. Эта формула дает нижний предел числа контрольных разрядов, необходимых для исправления одиночных ошибок. В табл. 1.1 показано необходимое количество контрольных разрядов для слов разного размера.

Табл.1.1 — Размерность кода

|  |  |  |  |
| --- | --- | --- | --- |
| Размер слова | Кол-во контроль разрядов | Общий размер | % увеличения длины слова |
| 8163264128256512 | 45678910 | 12213871136265522 | 50311911642 |

Этого теоретического нижнего предела можно достичь, используя метод Ричарда Хэмминга. В коде Хэмминга к слову, состоящему из m битов, добавляется r битов четности, при этом образуется слово длиной m+r битов. Биты нумеруются с единицы (а не с нуля), причем первым считается крайний левый. Все биты, номера которых — степени двойки, являются битами четности; остальные используются для данных. Например, к 16-битному слову нужно добавить 5 битов четности. Биты с номерами 1, 2, 4, 8 и 16 — биты четности, а все остальные — биты данных. Всего слово содержит 21 бит (16 битов данных и 5 битов четности). В рассматриваемом примере будем использовать положительную четность (выбор произвольный). Каждый бит четности проверяет определенные битовые позиции. Общее число битов со значением 1 в проверяемых позициях должно быть четным. Ниже указаны позиции проверки для каждого бита четности:

Бит 1 проверяет биты 1,3,5,7,9,11,13,15,17,19,21.

Бит 2 проверяет биты 2,3,6,7,10, 11, 14,15,18,19.

Бит 4 проверяет биты 4,5,6,7,12,13,14,15,20,21.

Бит 8 проверяет биты 8,9, 10, 11,12,13,14,15.

Бит 16 проверяет биты 16,17,18,19,20,21.

В общем случае бит b проверяется битами b1, b2,..., bj, такими что b1+b2+...+bj=b.

Например, бит 5 проверяется битами 1 и 4, поскольку 1+4-5. Бит 6 проверяется битами 2 и 4, поскольку 2+4=6 и т. д.

На рис. 1.3 показано построение кода Хэмминга для 16-битного слова 1111000010101110. Соответствующим 21-битным кодированным словом является 001011 100000101101110. Чтобы увидеть, как происходит исправление ошибок, рассмотрим, что произойдет, если бит 5 изменит значение из-за резкого скачка напряжения на линии электропередачи. В результате вместо кодированного слова 001011100000101101110 получится 001001100000101101110. Будут проверены 5 битов четности. Вот результаты проверки:

Бит четности 1 неправильный (биты 1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21 содержат пять единиц).

Бит четности 2 правильный (биты 2,3,6,7,10,11,14,15,18,19 содержат шесть единиц).

Бит четности 4 неправильный (биты 4,5,6,7,12,13,14,15,20,21 содержат пять единиц).

Бит четности 8 правильный (биты 8,9,10,11,12,13,14,15 содержат две единицы).

Бит четности 16 правильный (биты 16,17,18,19,20,21 содержат четыре единицы).

Общее число единиц в битах 1, 3, 5, 7, 9, 11, 13, 15, 17, 19 и 21 должно быть четным, поскольку в данном случае используется положительная четность. Неправильным должен быть один из битов, проверяемых битом четности 1 (а именно 1,3,5,7,9,11,13,15,17,19 и 21). Бит четности 4 тоже неправильный. Это значит, что изменил значение один из следующих битов: 4,5,6,7,12,13,14,15,20,21. Ошибка должна быть в бите, который содержится в обоих списках. В данном случае общими являются биты 5,7,13,15 и 21. Поскольку бит четности 2 правильный, биты 7 и 15 исключаются. Правильность бита четности 8 исключает наличие ошибки в бите 13. Наконец, бит 21 также исключается, поскольку бит четности 16 правильный. В итоге остается бит 5, в котором и содержится ошибка. Поскольку этот бит имеет значение 1, он должен принять значение 0. Именно таким образом исправляются ошибки.

Рис. 1.3 — Построение кода Хэмминга для слова 1111000010101110с помощью добавления 5 контрольных разрядов к битам данных

Чтобы найти неправильный бит, сначала нужно подсчитать все биты четности. Если они правильные, ошибки нет (или есть, но больше одной). Если обнаружились неправильные биты четности, то нужно сложить их номера. Сумма, полученная в результате, даст номер позиции неправильного бита. Например, если биты четности 1 и 4 неправильные, а 2, 8 и 16 правильные, то ошибка произошла в бите 5 (1+4)

1.4 Код Рида – Соломона

Код Рида-Соломона позволяет скорректировать одну ошибку в одном блоке данных. При его использовании к каждому блоку информации прибавляются дополнительные два элемента X и Y, значение которых находятся исходя из условий:

для трёх единиц информации (байт):байт1 + байт2 + байт3 + X + Y = 0

байт1 + 2 \* байт2 + 3 \* байт3 + 4 \* X + 5 \* Y = 0

для расчета конкретных значений X и Y для кодирования трёх байт:

Y = 3 \* байт1 + 2 \* байт2 + байт3X = -4 \* байт1 - 3 \* байт2 - 2 \* байт3

Теперь для выяснения ошибки и её коррекции применяем следующие расчеты:

Значение\_ошибки = байт1 + байт2 + байт3 + X + Y

Так как ранее (до возникновения ошибки) эта сумма была равна 0, то теперь она равна непосредственно значению ошибки, которое достаточно просто вычесть из недоброкачественного байта. В случае если блок принят безошибочно, то Значение\_ошибки = 0. Теперь найдём байт который надо исправлять:

N = байт1 + 2 \* байт2 + 3 \* байт3 + 4 \* X + 5 \* Y

Номер\_ ошибочного\_ байта = N / Значение\_ ошибки

При реализации этого в реальный алгоритм необходимо обязательно осуществлять проверку на то существует ошибка в блоке или нет, то есть Значение\_ошибки = 0 или нет, иначе получаем деление на ноль.

Если необходимо защитить кодом Рида-Соломона блок данных более 3х байт, то формулы расчета корректирующих значений лишь немного изменяются (для 16 байт):

Y = 16 \* байт1 + 15 \* байт2 + 14 \* байт3 + ... + байт16X = -17 \* байт1 - 16 \* \*байт2 - 15 \* байт3 - ... - 2 \* байт16

Значение\_ошибки = байт1 + байт2 + байт3 + ... + X + YN = байт1 + 2 \* \*байт2 + 3 \* байт3 + ... + 16 \* байт16 + 17 \* X + 18 \* Y

Данным кодом неудобно защищать блоки информации менее 4 байт, так как длинна контрольных параметров X и Y должна быть как минимум 4 байта

2 байта (DW) для X и 2 байта на Y, то есть получается. что к блоку данных из 4 байт будет добавлен корректирующий блок из 4 байт.

Но что делать, если возникло две или более ошибок в блоке ?

Как один из признаков возникновения двух ошибок можно считать получения в качестве номер ошибочного байта дробного числа, например если в блоке из нулей встретится 2 единицы (две ошибки), в третьем и четвёртом байтах, то Номер ошибочного байта = 3.5 но если 4 единицы, соответственно в 3, 4 и 5 байтах то Номер ошибочного байта = 4.

2 Разработка стенда контроля передаваемой информации

Для визуализации процесса помехоустойчивого кодирования можно разработать устройство, которое поможет понять принцип работы метода Хэмминга. Кодер – декодер будем разрабатывать на основе ИМС К555ВЖ1.

2.1 Разработка устройства кодирования информации методом Хемминга

Кодер, преобразует 32х битное слово в 38ми разрядный код Хэмминга, после чего слово хранится в памяти или передаётся по шинам и т.д. В процессе передачи или хранения в слове может произойти ошибка, поэтому декодер не просто декодирует слово, но и исправляет единичные ошибки. Процесс кодирования и декодирования можно представить в виде блок-схемы приведенной ниже.

Рис. 2.1 блок схема кодирования

Рис. 2.2 — Схема кодера

На входы а1,а2…а32 подаётся кодируемое слово. Блоки (элементы) KR1, KR2, KR4, KR8, KR16 и KR32 формируют контрольные разряды (биты чётности). Весь процесс кодирования сводится к формированию контрольных разрядов и сдвигу имеющихся разрядов. Так, 1й бит слова становится 3м битом кода, а 2й – 5м , 1,2,4,8,16,32 биты – контрольные. В результате кодирования 32х битное слово становится 38ми битным.

Рис 2.3 — Схема блоков KR1 – KR32

Блоки KR1, KR2, KR4, KR8, KR16 и KR32 собраны элементах исключающее ИЛИ. Итак:

Блок KR1 формирует 1й контрольный разряд подсчитывая чётность 3,5,7,9,11,13,15,17,19,21,23,25,27,29,31,33,35 и 37 битов, если количество единиц в этих разрядах чётное, то контрольный бит равен нулю, если нечётное – единица.

Аналогично блок KR2 формирует 2й контрольный разряд, подсчитывая чётность 3,6,7,10,11,14,15,18,19,22,23,26,27,30,31,34,35 и 38 битов, если количество единиц в этих разрядах чётное, то контрольный бит равен нулю, если нечётное – единица.

Блок KR4 формирует 4й контрольный бит проверяя 5,6,7,12,13,14,15,20,21,22,23,28,29,30,31,36,37 и 38 биты.

Блок KR8 формирует 8й контрольный бит проверяя 9,10,11,12,13,14,15,24,25,26,27,28,29,30 и 31 биты.

Блок KR16 формирует 16й контрольный бит проверяя 17,18,19,20,21,22,23,24,25,26,27,28,29,30 и 31 биты.

Блок KR32 формирует 32й контрольный бит проверяя 33,34,35,36,37 и 38 биты.

2.2 Разработка устройства декодирования информации методом Хемминга

Рис.2.4 — Схема декодера

Схема декодера состоит из 3х основных блоков: блока анализа, блока управления и блока исправления. Декодер осуществляет декодирование с исправлением произошедшей ошибки.

Рис. 2.5 — Схема блока анализа

Блок анализа с помощью блоков KR1, KR2, KR4, KR8, KR16 и KR32 заново проверяет чётность выше упомянутых разрядов и с помощью 6 и элементов исключающее ИЛИ сравнивает с пришедшими по шине контрольными разрядами. Если контрольные разряды нарушены, то формируются 6 сигналов ошибки, которые передаются в блок управления. Так же блок анализа изымает из 38ми битного слова биты чётности и передаёт 32х битное слово в блок исправления.

Рис. 2.6 — Схема блока управления

Блок управления представляет дешифратор 6/32, собранный на базе элементов И. В соответствии с комбинацией входных сигналов, блок управления активирует тот или иной управляемый инвертор блока исправления.

Рис 2.7 — Схема блока исправления ошибок

Блок исправления непосредственно производит исправление слова пришедшего из блока анализа в соответствии с сигналами блока управления. Он представляет собой 32 управляемых инвертора, на информационные входы которых подаётся слово с ошибкой, а на управляющие код из блока управления. В соответствии с поданным сигналом активируется тот или иной инвертор, который и исправляет ошибку, и слово выводится из дешифратора.

Рис.2.8 — Схема управляемого инвертора

Управляемый инвертор собран на базе 2х элементов НЕ, элемента И, и элемента исключающее ИЛИ-НЕ. Является непосредственно исполнителем исправления ошибки. Его таблица истинности приведена ниже.

Табл. 2.1 — таблица истинности управляемого инвертора

|  |  |  |
| --- | --- | --- |
| ВХОД | УПР.СИГНАЛ | ВЫХОД |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

2.3 Реализация кодера – декодера на базе ИМС К555ВЖ1

Рис 2.9 — Блок схема

32х битное слово разделяется на два слова по 16 бит. Далее каждая из половин попадает в свой 16битный кодер. Генерируются контрольные биты, и на линию передач выводятся два 22х битных кода Хэмминга. На линии передачи есть возможность внести ошибку. Далее 22х битный код считывается декодером. Проверяется на наличие ошибок. Если найдены ошибки, они исправляются, и выводятся два слова по 16 бит, которые и составляют переданное 32х битное слово.

2.3.1 Цоколёвка ИМС К555ВЖ1 (SN74LS630)

Рис. 2.10 — Цоколёвка ИМС К555ВЖ1 (SN74LS630)

* D0-D15 - входы/выходы данных;
* CB0-CB5 - входы/выходы синдрома;
* S0,S1 - входы управления;
* SEF - флаг ошибки;
* DEF - флаг некорректируемой ошибки.
* Все выходы с тремя состояниями

Схема использует модифицированный код Хэмминга, чтобы генерировать 6-битовый синдром для 16-разрядного слова. Одиночные ошибки схемой исправляются; двойные ошибки только обнаруживаются, ошибки типа всех нулей или всех единиц тоже обнаруживаются, тройные и большие ошибки могут быть не обнаружены. В случае двойной ошибки, оба бита могут быть в слове памяти, обе в слове синдрома, или по одному биту в слове памяти и слове синдрома. В течении цикла записи, схема формирует 6-разрядное проверочное слово из 16-разрядного слова данных. Коррекция ошибки выполняется идентификацией неправильного бита и инверсией его. Когда исправленное слово выдается из порта данных, из порта CBi выдается синдром, указывающий на неправильный бит.

Табл. 2.2 — Функции схемы исправления ошибок

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Цикл памяти |  Упр S1 S0 | Функция БИС | Шина данных | Шина синдрома | Флаги SEF DEF |
| ЗАПИСЬ | L L | Генерировать контрольное слово | Входные данные | Выходное контрольное слово | L L |
| ЧТЕНИЕ | L H | Читать данные и контрольное слово | Входные данные | Входное контрольное слово | L L |
| ЧТЕНИЕ | H H | Защелкнуть и разрешить флаги | Защелкнутые данные | Защелкнутое контрольное слово | Разрешено |
| ЧТЕНИЕ | H L | Корректировать данные и генерировать синдром | Выходные корректированные данные | Выходной синдром | Разрешено |

Табл. 2.3 — Таблица ошибок

|  |  |
| --- | --- |
| Контрольное | 16-разрядное слово данных |
| слово | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| CB0 | \* | \* |  | \* | \* |  |  |  | \* | \* | \* |  |  | \* |  |  |
| CB1 | \* |  | \* | \* |  | \* | \* |  | \* |  |  | \* |  |  | \* |  |
| CB2 |  | \* | \* |  | \* | \* |  | \* |  | \* |  |  | \* |  |  | \* |
| CB3 | \* | \* | \* |  |  |  | \* | \* |  |  | \* | \* | \* |  |  |  |
| CB4 |  |  |  | \* | \* | \* | \* | \* |  |  |  |  |  | \* | \* | \* |
| CB5 |  |  |  |  |  |  |  |  | \* | \* | \* | \* | \* | \* | \* | \* |

Табл. 2.4 — Синдромы ошибок

|  |  |
| --- | --- |
| Ячейка ошибки | Код синдрома ошибки |
|  | CB0 | CB1 | CB2 | CB3 | CB4 | CB5 |
| D0 | L | L | H | L | H | H |
| D1 | L | H | L | L | H | H |
| D2 | H | L | L | L | H | H |
| D3 | L | L | H | H | L | H |
| D4 | L | H | L | H | L | H |
| D5 | H | L | L | H | L | H |
| D6 | H | L | H | L | L | H |
| D7 | H | H | L | L | L | H |
| D8 | L | L | H | H | H | L |
| D9 | L | H | L | H | H | L |
| D10 | L | H | H | L | H | L |
| D11 | H | L | H | L | H | L |
| D12 | H | H | L | L | H | L |
| D13 | L | H | H | H | L | L |
| D14 | H | L | H | H | L | L |
| D15 | H | H | L | H | L | L |
| CB0 | L | H | H | H | H | H |
| CB1 | H | L | H | H | H | H |
| CB2 | H | H | L | H | H | H |
| CB3 | H | H | H | L | H | H |
| CB4 | H | H | H | H | L | H |
| CB5 | H | H | H | H | H | L |
| Нет ошибки | H | H | H | H | H | H |

Табл. 2.5 — Исправление

|  |  |  |
| --- | --- | --- |
| Полное число ошибок | Флаги ошибок | Исправление данных |
| 16-битовые данные | 6-битовое контрольное слово | SEF | DEF |  |
| 0 | 0 | L | L | Не применимо |
| 1 | 0 | H | L | Коррекция |
| 0 | 1 | H | L | Коррекция |
| 1 | 1 | H | H | Прерывание |
| 2 | 0 | H | H | Прерывание |
| 0 | 2 | H | H | Прерывание |

2.3.2 Разработка принципиальной схемы устройства

Рис. 2.11 — Принципиальная схема устройства

На рисунке представлена принципиальная схема стенда, собранного на следующих радиоэлементах:

Микросхема К555ВЖ1 – 4шт

Микросхема К555ЛН2 – 1шт

Джампера на 2 штырька – 47шт

Джампера на 3 штырька – 47шт

Диодные мосты на 0.75А – 2шт

Конденсаторы 222мФ – 2шт

Резисторы 2.2кОм – 16шт

Трансформатор – 1шт

Светодиоды красные – 84шт

Светодиоды желтые – 17шт

Светодиоды зелёные – 24шт

2.3.3 Принцип работы устройства

Верхний ряд джамперов и микросхем является кодером. Нижний ряд микросхем – декодер. Средний ряд джамперов – линии передач. С помощью джамперов 0 – 31 вводится 32х битное слово. При подаче логический 0 на входы управления S1 и S0 кодера, происходит считывание слова и формирование контрольных разрядов. 32х битное слово преобразованное в 48ми битный код выводится на верхний ряд светодиодов.

Посредствам ряда джамперов или микросхемы инвертора генерируем ошибку на линии передач.

Чтобы считать код с линии передач, подаём на входы управления декодера S1 логический 0 и на S0 логическую 1.Далее подаём на S1 и S0 логическую 1, тем самым записывая код в буфер микросхемы. Чтобы получить скорректированное (без ошибки) слово, необходимо подать на S1 логическую 1 и на S0 логический 0., и исходное слово появится на нижнем ряду светодиодов.

3. Экономическая часть

После рассмотрения вариантов технологических процессов, обеспечивающие примерно одинаковое качество изделия, соответствующие требованиям задания необходимо выбрать наиболее экономичный из возможных вариантов и только после этого произвести детальную разработку устройств. Для этого необходимо предварительно оценить прогрессивность выбираемого технологического процесса с помощью различных технико-экономических показателей, в той или иной степени влияющих на экономическую эффективность.

Сумма меняющихся при изменении технологического варианта затрат носит название технологической себестоимости. Входящие в технологическую себестоимость затраты делятся на переменные (условно пропорциональные) и условно постоянные. Переменные затраты - расходы на сырьё, материалы на ремонт и содержания оборудования, быстро изнашивающийся инструмент, силовую энергию, амортизационные отчисления. Они изменяются примерно пропорционально изменению объему выпускаемой продукции. Условно постоянные затраты - это расходы на содержание персонала, отопление, освещение – не зависят от объема производства. Сравнительный анализ ведётся по приведенным затратам и выбирается вариант, обеспечивающий минимальное значение:

STI+ENKI MIN

STI- технологическая себестоимость;

KI- удельное капитальное вложение;

EN- нормативный коэффициент экономической эффективности;

В более сложных случаях решение задач выбора оптимальных вариантов технологий широко применяются методы прикладной математики, в частности математическое программирование.

Далее производится расчет производственной себестоимости устройства.

Себестоимость устройства включает в себя:

1. Стоимость сырья и основных материалов
2. Стоимость покупных изделий и полуфабрикатов
3. Заработную плату производственных рабочих
4. Отчисления на социальное страхование
5. Накладные расходы

Расчет стоимости материалов сведен в таблицу 3.

Таблица 3.1 Расчет стоимости материалов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Материалы | Кол-во | Единица измерения | Цена за единицу (грн./ед.) | Сумма (грн.) |
| Припой ПОС-61 | 0,2 | кг | 20 | 4 |
| Провода | 10 | М | 0.50 | 5 |
| Канифоль | 0.005 | кг | 50 | 0,25 |
| Гетинакс фольгированныйРаствор хлорного железа | 250\*3500,5 | Мм2л | -------------- | 185 |
| Итого: |  |  |  | 32,25 |

Транспортные расходы составляют 10 % от стоимости материалов и равны 3,20 грн.

Таким образом, затраты на материалы составляют ~.36грн.

Расчет затрат на покупные изделия приведены в таблице 3.2.

Таблица 3.2 Расчет затрат на покупные изделия.

|  |  |  |  |
| --- | --- | --- | --- |
| Наименование | Цена/шт.(грн.) | Кол-во(шт.) | Цена (грн.) |
| Микросхемы: |  |  |  |
| К555ВЖ1 | 4 | 4 | 16 |
| К555ЛН2 | 2 | 1 | 2 |
| Джамперы: |  |  |  |
| На 2 штырька | 0,30 | 47 | 14,1 |
| На 3 штырька | 0,40 | 37 | 14,8 |
| Диодный мост | 1,5 | 2 | 3 |
| Конденсатор 222мФ | 0,7 | 2 | 1,4 |
| Резисторы | 0,15 | 16 | 2,4 |
| Светодиоды: |  |  |  |
| Красные | 0,2 | 84 | 16,8 |
| Желтые | 0,2 | 17 | 3,4 |
| Зелёные | 0,2 | 24 | 4,8 |
| Панельки под ИМС: |  |  |  |
| На 28 ножек | 4 | 2 | 8 |
| На 14 ножек | 1 | 1 | 1 |
| Трансформатор: | 15 | 1 | 15 |
| Итого: |  |  |  |
|  |  |  | 102,7 |

Стоимость разработки равна

Sраз.= Bосн.⋅N⋅(1+Wс+Wд), где (3.1)

Sраз. - стоимость разработки, грн.

Bосн. - заработная плата разработчика, грн. в месяц

N - количество месяцев разработки

Wс - отчисления на социальное страхование

Wд - дополнительная заработная плата.

Sраз. = 32\*1⋅(1+0,41+0,08) = 32\*1,49 = 47,68 грн. (3.2)

В связи с тем, что устройство было изготовлено без сторонних рабочих, своими силами, то себестоимость устройства не включает расходы на заработную плату и накладные расходы.

Sп= Sраз + Sм +Sтр+ Sпк=47,68+32,25+3,20+102,7=185,83грн. (3.3)

Sраз. - стоимость разработки, грн.

Sм- затраты на материалы, грн.

Sтр- транспортные расходы, грн.

Sпк- затраты на покупные изделия, грн.

На основании проведенных выше расчетов определим цену устройства табл.5.

Таблица 3.3 Расчёт цены устройства.

|  |  |  |
| --- | --- | --- |
| N п/п | Статьи расходов | Сумма (грн.) |
| 1 | Основные материалы | 32,25 |
| 2 | Покупные изделия и полуфабрикаты | 102,7,4 |
| 3 | Заработная плата производственных рабочих | 0 |
| 4 | Отчисления на социальное страхование | 0 |
| 5 | Накладные расходы, включая затраты на разработку | 47,68 |
| 6 | Производственная себестоимость: | 182,63 |
|  | Цена устройства | 182,63 |

Таким образом, цена устройства составляет 185 грн. Аналогичные устройства по оснащенности, которые можно приобрести только под заказ, будут стоить как минимум в два раза выше, без учета затрат на транспортировку.

4 Охрана труда и техники безопасности

4.1 Потенциально опасные и вредные производственные факторы

Имеющийся в настоящее время в нашей стране комплекс разработанных организационных мероприятий и технических средств защиты, накопленный передовой опыт работы ряда учебных заведений показывает, что имеется возможность добиться значительно больших успехов в деле устранения воздействия на работающих опасных и вредных производственных факторов. Однако состояние условий труда и его безопасности в ряде учебных заведений еще не удовлетворяют современным требованиям. Курсанты, студенты, преподаватели и другие работники колледжа еще сталкиваются с воздействием таких физически опасных и вредных производственных факторов, как повышенный уровень шума, повышенная температура внешней среды, отсутствие или недостаточная освещенность рабочей зоны, электрический ток, статическое электричество и другие.

Многие курсанты и студенты связаны с воздействием таких психофизических факторов, как умственное перенапряжение, перенапряжение зрительных и слуховых анализаторов, монотонность труда, эмоциональные перегрузки. Воздействие указанных неблагоприятных факторов приводит к снижению работоспособности, вызванное развивающимся утомлением. Появление и развитие утомления связано с изменениями, возникающими во время работы в центральной нервной системе, с тормозными процессами в коре головного мозга. Например, сильный шум вызывает трудности с распознанием цветовых сигналов, снижает быстроту восприятия цвета, остроту зрения, зрительную адаптацию, нарушает восприятие визуальной информации, уменьшает на 5-12 % производительность труда. Длительное воздействие шума с уровнем звукового давления 90 дБ снижает производительность труда на 30-60 %.

Медицинские обследования курсантов и студентов показали, что помимо снижения производительности труда, высокие уровни шума приводят к ухудшению слуха. Длительное нахождение человека в зоне комбинированного воздействия различных неблагоприятных факторов может привести к профессиональному заболеванию. Анализ травматизма среди курсантов и студентов показывает, что в основном несчастные случаи происходят от воздействия физически опасных производственных факторов таких как: выполнении несвойственных им работ, выполнение работ без спец одежды или не исправным инструментом. На втором месте случаи, связанные с воздействием электрического тока.

4.2 Обеспечение электробезопасности

Электрические установки, к которым относится практически все оборудование ЭВМ, представляют для человека большую потенциальную опасность, так как в процессе эксплуатации или проведении профилактических работ человек может коснуться частей, находящихся под напряжением. Специфическая опасность электроустановок: токоведущие проводники, корпуса ЭВМ и прочего оборудования, оказавшегося под напряжением в результате повреждения (пробоя) изоляции, не подают каких-либо сигналов, которые предупреждают человека об опасности. Реакция человека на электрический ток возникает лишь при протекании последнего через тело человека. Исключительно важное значение для предотвращения электротравматизма имеет правильная организация обслуживания действующих электроустановок учебных лабораторий, проведения ремонтных, монтажных и профилактических работ. При этом под правильной организацией понимается строгое выполнение ряда организационных и технических мероприятий и средств, установленных действующими “Правилами технической эксплуатации электроустановок потребителей и правила техники безопасности при эксплуатации электроустановок потребителей” (ПТЭ и ПТБ потребителей) и “Правила установки электроустановок” (ПУЭ). В зависимости от категории помещения необходимо принять определенные меры, обеспечивающие достаточную электробезопасность при эксплуатации и ремонте электрооборудования. Так, в помещениях с повышенной опасностью электроинструменты, переносные светильники должны быть выполнены с двойной изоляцией или их напряжение питания не должно превышать 42В. В учебных заведениях к таким помещениям могут быть отнесены помещения машинного зала, помещения для размещения сервисной и периферийной аппаратуры. В особо опасных же помещениях напряжение питания переносных светильников не должно превышать 12В, а работа с напряжением не выше 42В разрешается только с применением СИЗ (диэлектрических перчаток, ковриков и т.п.). Работы без снятия напряжения на токоведущих частях и вблизи них, работы проводимые непосредственно на этих частях или при приближении к ним на расстояние менее установленного ПЭУ. К этим работам можно отнести работы по наладке отдельных узлов, блоков. При выполнении такого рода работ в электроустановках до 1000В необходимо применение определенных технических и организационных мер, таких как:

* ограждения, расположенные вблизи рабочего места и других токоведущих частей, к которым возможно случайное прикосновение;
* работа в диэлектрических перчатках, или стоя на диэлектрическом коврике;
* применение инструмента с изолирующими рукоятками, при отсутствии такого инструмента следует пользоваться диэлектрическими перчатками.

Работы этого вида должны выполнятся не менее чем двумя курсантами или под присмотром преподавателя.

В соответствии с ПТЭ и ПТВ потребителям и обслуживающему персоналу электроустановок предъявляются следующие требования:

* лица, не достигшие 18-летнего возраста, не могут быть допущены к работам в электроустановках;
* лица не должны иметь увечий и болезней, мешающих производственной работе;
* лица должны после соответствующей теоретической и практической подготовки пройти проверку знаний и иметь удостоверение на доступ к работам в электроустановках.

В учебных лабораториях разрядные токи статического электричества чаще всего возникают при прикосновении к любому из элементов ЭВМ. Такие разряды опасности для человека не представляют, но кроме неприятных ощущений они могут привести к выходу из строя ЭВМ. Для снижения величины возникающих зарядов статического электричества в учебных лабораториях покрытие технологических полов следует выполнять из однослойного поливинилхлоридного антистатического линолеума. Другим методом защиты является нейтрализация заряда статического электричества ионизированным газом. К общим мерам защиты от статического электричества в учебных лабораториях можно отнести общие и местное увлажнение воздуха.

4.3 Обеспечение санитарно-гигиенических требований к помещениям учебных лабораторий

Помещения учебных лабораторий, их размеры (площадь, объем) должны в первую очередь соответствовать количеству работающих и размещаемому в них комплекту технических средств. В них предусматриваются соответствующие параметры температуры, освещения, чистоты воздуха, обеспечивают изоляцию, от производственных шумов и т.п. Для обеспечения нормальных условий труда санитарные нормы СН 245-71 устанавливают на одного работающего, объем помещения не менее 15 м3, площадь помещения выгороженного стенами или глухими перегородками не менее 4,5 м3.

Для эксплуатации ЭВМ следует предусматривать следующие помещения:

* машинный зал, помещение для размещения сервисной и периферийной аппаратуры, помещение для хранения запасных деталей, инструментов, приборов (ЗИП);
* помещения для размещения приточно-вытяжных вентиляторов;
* помещение для персонала;
* помещение для приема-выдачи информации.

Основные помещения учебных лабораторий располагаются в непосредственной близости друг от друга. Их оборудуют вентиляцией и искусственным освещением. К помещению машинного зала предъявляются особые требования. Площадь машинного зала должна соответствовать площади, необходимой по заводским техническим условиям данного типа ЭВМ.

Высота зала над технологическим полом до подвесного потолка должна быть 3-3,5м. Расстояние между подвесным и основным потолками при этом должно быть 0,5-0,8м. Высоту подпольного пространства принимают равной 0,2-0,6м.

В учебных лабораториях, как правило, применяется боковое естественное освещение. Рабочие комнаты и кабинеты должны иметь естественное освещение. В остальных помещениях допускается искусственное освещение.

В тех случаях, когда одного естественного освещения не хватает, устанавливается совмещенное освещение. При этом дополнительное искусственное освещение применяется не только в темное, но и в светлое время суток.

Искусственное освещение по характеру выполняемых задач делится на рабочее, аварийное, эвакуационное.

Рациональное цветовое оформление помещения направлено на улучшение санитарно-гигиенических условий труда, повышение его производительности и безопасности. Окраска помещений учебных лабораториях влияет на нервную систему человека, его настроение, и в конечном счете на производительность труда. Основные производственные помещения целесообразно окрашивать в соответствии с цветом технических средств. Освещение помещения и оборудования должно быть мягким, без блеска.

Снижение шума, создаваемого на рабочих местах учебных лабораторий внутренними источниками, а также шума, проникающего извне, является очень важной задачей. Снижение шума в источнике излучения можно обеспечить применением упругих прокладок между основанием машины, прибора и опорной поверхностью. В качестве прокладок используются резина, войлок, пробка, различной конструкции амортизаторы. Под настольные шумящие аппараты можно подкладывать мягкие коврики из синтетических материалов, а под ножки столов, на которых они установлены прокладки из мягкой резины, войлока, толщиной 6-8мм. Крепление прокладок возможно путем приклейки их к опорным частям.

Возможно также применение звукоизолирующих кожухов, которые не мешают технологическому процессу. Не менее важным для снижения шума в процессе эксплуатации является вопрос правильной и своевременной регулировки, смазывания и замены механических узлов шумящего оборудования.

Рациональная планировка помещения, размещения оборудования в учебных лабораториях является важным фактором, позволяющим снизить шум при существующем оборудовании ЭВМ. При планировке учебных лабораторий машинный зал и помещение для сервисной аппаратуры необходимо располагать вдали от шумящего и вибрирующего оборудования.

Снижение уровня шума, проникающего в помещение извне, может быть достигнуто увеличением звукоизоляции ограждающих конструкций, уплотнением по периметру притворов окон, дверей.

Таким образом, для снижения шума создаваемого на рабочих местах внутренними источниками, а также шума, проникающего извне, следует:

* ослабить шум самих источников (применение экранов, звукоизолирующих кожухов);
* снизить эффект суммарного воздействия отраженных звуковых волн (звукопоглощающие поверхности конструкций);
* применять рациональное расположение оборудования;
* использовать архитектурно-планировочные и технологические решения изоляции источников шума.

4.4 Противопожарная защита

Пожары в учебных лабораториях представляют особую опасность, так как сопряжены с большими материальными потерями. Характерная особенность учебных лабораторий - небольшие площади помещений. Как известно, пожар может возникнуть при взаимодействии горючих веществ, окисления и источников зажигания. В помещениях учебных лабораторий присутствуют все три основные фактора, необходимые для возникновения пожара.

Горючими компонентами на учебных лабораториях являются: строительные материалы для акустической и эстетической отделки помещений, перегородки, двери, полы, перфокарты и перфоленты, изоляция кабелей и др.

Противопожарная защита - это комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, на предотвращение пожара, ограничение его распространения, а также на создание условий для успешного тушения пожара.

Источниками зажигания в учебных лабораториях могут быть электронные схемы от ЭВМ, приборы, применяемые для технического обслуживания, устройства электропитания, кондиционирования воздуха, где в результате различных нарушений образуются перегретые элементы, электрические искры и дуги, способные вызвать возгорание горючих материалов.

В современных ЭВМ имеет место быть очень высокая плотность размещения элементов электронных схем. В непосредственной близости друг от друга располагаются соединительные провода, кабели. При протекании по ним электрического тока выделяется значительное количество теплоты. При этом возможно оплавление изоляции. Для отвода избыточной теплоты от ЭВМ служат системы вентиляции и кондиционирования воздуха. При постоянном действии эти системы представляют собой дополнительную пожарную опасность.

Энергоснабжение учебных лабораторий осуществляется от трансформаторной станции. На трансформаторных подстанциях особую опасность представляют трансформаторы с масляным охлаждением. В связи с этим предпочтение следует отдавать сухим трансформаторам.

Для безопасной работы необходим правильный расчет и выбор аппаратов защиты. При проведении обслуживающих, ремонтных и профилактических работ используются различные смазочные вещества, легковоспламеняющиеся жидкости, прокладываются временные электропроводки, ведется пайка и чистка отдельных узлов. Возникает дополнительная пожарная опасность, требующая дополнительных мер пожарной защиты. В частности, при работе с паяльником следует использовать несгораемую подставку с несложными приспособлениями для уменьшения потребляемой мощности в нерабочем состоянии.

Для большинства помещений учебных лабораторий установлена категория пожарной опасности В.

Одной из наиболее важных задач пожарной защиты является защита строительных помещений от разрушений и обеспечение их достаточной прочности в условиях воздействия высоких температур при пожаре. Учитывая высокую стоимость электронного оборудования учебных лабораторий, а также категорию его пожарной опасности, здания для учебных лабораторий и части здания другого назначения, в которых предусмотрено размещение ЭВМ, должны быть 1 и 2 степени огнестойкости.

Для изготовления строительных конструкций используются, как правило, кирпич, железобетон, стекло, металл и другие негорючие материалы. Применение дерева должно быть ограничено, а в случае использования, необходимо пропитывать его огнезащитными составами. В учебных лабораторий противопожарные преграды в виде перегородок из несгораемых материалов устанавливают между машинными залами.

К средствам тушения пожара, предназначенных для локализации небольших возгораний, относятся пожарные стволы, внутренние пожарные водопроводы, огнетушители, сухой песок, асбестовые одеяла и т. п.

В зданиях учебных заведений пожарные краны устанавливаются в коридорах, на площадках лестничных клеток и входов. Вода используется для тушения пожаров в помещениях программистов, библиотеках, вспомогательных и служебных помещениях. Применение воды в машинных залах ЭВМ, помещениях контрольно-измерительных приборов ввиду опасности повреждения или полного выхода из строя дорогостоящего оборудования возможно в исключительных случаях, когда пожар принимает угрожающе крупные размеры. При этом количество воды должно быть минимальным, а устройства ЭВМ необходимо защитить от попадания воды, накрывая их брезентом или полотном.

Для тушения пожаров на начальных стадиях широко применяются огнетушители. По виду используемого вещества огнетушители подразделяются на следующие основные группы:

* Пенные огнетушители, применяются для тушения горящих жидкостей, различных материалов, конструктивных элементов и оборудования, кроме электрооборудования, находящегося под напряжением.
* Газовые огнетушители, применяются для тушения жидких и твердых веществ, а также электроустановок, находящихся под напряжением.
* В производственных помещениях учебных заведений применяются главным образом углекислотные огнетушители, достоинством которых является высокая эффективность тушения пожара, сохранность электронного оборудования, диэлектрические свойства углекислого газа, что позволяет использовать эти огнетушители даже в том случае, когда не удается обесточить электроустановку сразу.

Для обнаружения начальной стадии загорания и оповещения службы пожарной охраны используют системы автоматической пожарной сигнализации (АПС). Кроме того, они могут самостоятельно приводить в действие установки пожаротушения, когда пожар еще не достиг больших размеров. Системы АПС состоят из пожарных извещателей, линий связи и приемных пультов (станций).

Эффективность применения систем АПС определяется правильным выбором типа извещателей и мест их установки. При выборе пожарных извещателей необходимо учитывать конкретные условия их эксплуатации: особенности помещения и воздушной среды, наличие пожарных материалов, характер возможного горения, специфику технологического процесса и т.п.

В соответствии с “Типовыми правилами пожарной безопасности для промышленных предприятий”, залы ЭВМ, помещения для внешних , подготовки данных, сервисной аппаратуры, архивов, копировально-множительного оборудования и т.п. необходимо оборудовать дымовыми пожарными извещателями. В этих помещениях в начале пожара при горении различных пластмассовых, изоляционных материалов и бумажных изделий выделяется значительное количество дыма и мало теплоты.

В других помещениях учебных заведений допускается применение тепловых пожарных извещателей.

Объекты, учебных заведений, крайне опасные с точки зрения противопожарной безопасности, необходимо оборудовать установками стационарного автоматического пожаротушения. Наиболее целесообразно применять в установки газового тушения пожара, действие которых основано на быстром заполнении помещения огнетушащим газовым веществом с резким снижением содержания в воздухе кислорода.

Пожарам в учебных заведений должно уделяться особое внимание, так как пожары в учебных заведений сопряжены с опасностью для человеческой жизни и большими материальными потерями.

Заключение

С увеличением количества компьютеров стало увеличиваться количество передаваемой информации. Появилась необходимость контроля и защиты от ошибок. Для повышения наглядности при изучении базовых принципов кодирования применяются электрифицированные стенды. В данной работе показывается актуальность разработки подобного электрифицированного стенда - «Устройство кодирования – декодирования 32х разрядных слов методом Хэмминга», который достаточно просто демонстрирует алгоритм кодирования.

В работе приведена классификация видов помехоустойчивого кодирования, представлены распространенные алгоритмы кодирования.

Также, в работе приведены расчеты принципиальной логической схемы каждого блока в ElectronicWorkbench, что позволяет эмулировать работу каждого блока в этой среде.

Расчет себестоимости данного устройства показал экономическую целесообразность данной работы.

Список ссылок

1 Э. Таненбаум Архитектура Компьютера «ПИТЕР» 2006 г.

2 Ю. І. Якименко Мікропроцесорна техніка «КИЇВ» 2004 г.

3 Е. Угрюмов Цифровая схемотехника «БХВ – Санкт-Петербург» 2001 г.

Проф .В.Г .Герасимова Основы промышленной электроники Москва «Высшая школа» 1978г.

4 О.Г. Верховцев , К.П. Лютов Практические советы мастеру – любителю по электротехнике и электронике «Энергоатомиздат» 1984г.

5 А.Медведев Технология производства печатных плат «Техносфера» 2005г.

6 В.С. Гутников Интегральная электроника в измерительных устройствах, Л.: Энргоатомиздат, 1988

7 А.Л. Булычев, В.И.Галкин Аналоговые интегральные схемы, Мн.: Беларусь, 1994

8 М.И. Богданович, И.Н. Грель Цифровые интегральные микросхемы: справочник, Mн.: Беларусь, 1991

9 В.Л. Шило Популярные цифровые микросхемы: справочник, М.: Радио и связь, 1987