ВВЕДЕНИЕ

Информация - это одна из самых ценных вещей в современной жизни. Появление глобальных компьютерных сетей сделало простым получение доступа к информации. Легкость и скорость доступа к данным с помощью компьютерных сетей, таких как Интернет, сделали значительными следующие угрозы безопасности данных:

− неавторизованный доступ к информации;

− неавторизованное изменение информации;

− неавторизованный доступ к сетям и другим сервисам;

Вне зависимости от того, насколько ценна информация, хранящаяся на компьютере, законодательством она признается объектом частной собственности. Владелец этой информации, имеет право определять правила ее обработки и зашиты, а также предпринимать необходимые меры для предотвращения утечки, хищения, утраты и подделки информации.

На сегодняшний день для защиты информации от несанкционированного доступа применяются программные, аппаратные и программно-аппаратные средства.

Программная реализация системы защиты информации имеет низкую стоимость, но качество защиты данных невысокое. Аппаратная реализация, обеспечивает более высокое качество защиты информации, но цена таких технических решений возрастает.

Программно-аппаратные системы для защиты информации предоставляют пользователю гибкость настройки и высокую защищенность данных. Программно-аппаратная криптосистема состоит из электронного устройства, которое подключается к персональному компьютеру и программного обеспечения для работы с устройством. В таких системах выполнение функций, некритичных к скорости работы и безопасности, перекладывается на программное обеспечение, что способствует снижению их стоимости.

Целью данной работы является создание программно-аппаратного комплекса, который обеспечивает надежную защиту информации и не создает неудобств при эксплуатации.

1. АНАЛИЗ ТЕХНИЧЕСКОГО ЗАДАНИЯ

## Выбор метода шифрования

Любая ценная информация, которая распространяется на разнообразных носителях, или по каналам связи в локальных и глобальных сетях, может быть модифицирована по некоторым правилам, с целью сохранения ее конфиденциальности и целостности [1].

Чтобы предупредить потерю целостности информации применяют помехоустойчивое кодирование, которое фиксирует нарушения целостности данных и предоставляет возможность их полного или частичного восстановления.

Для уменьшения объема информации, которая должна быть передана адресату, применяют сжатие (архивацию) данных. Часть архивной информации может быть предназначена для возможности восстановления содержимого архива после ошибок при транспортировке по электронным каналам связи, либо в результате повреждения носителей с исходными данными. Важным преимуществом использования сжатия информации является то, что с уменьшением объема данных, которые необходимо передать адресату, уменьшается вероятность возникновения ошибок и потери целостности данных.

Несмотря на преимущества кодирования и сжатия информации, существуют проблемы, которые нельзя решить с помощью этих методов.

Целью кодирования информации является предотвращение ее повреждения. Кодирование не изменяет статистических характеристик информации, так как код для восстановления отбрасывается сразу после проверки целостности данных.

Архивация информации уменьшает ее избыточность. Для того чтобы сделать это как можно эффективнее, используются специальные алгоритмы, которые не предназначены для сокрытия статистических зависимостей между частями исходного сообщения. В большинстве случаев достаточно просто реализовать обратное преобразование.

С целью обеспечения конфиденциальности информации используют особый вид преобразований, который называется «шифрование» [1]. Целью шифрования является сокрытие содержательной и статистической зависимости между частями исходного сообщения.

В качестве устройств шифрования наиболее широко используют три вида шифраторов: программные, аппаратные и программно-аппаратные. Основное различие между ними заключается не только в способе реализации шифрования и степени надежности защиты данных, но и в цене. Стоимость аппаратных шифраторов существенно выше программных, но это окупается за счет более высокого качества защиты информации.

В данной работе будет реализовано аппаратное шифрование, потому что этот метод:

* гарантирует неизменность самого алгоритма, тогда как программный алгоритм может быть намеренно модифицирован.
* исключает какое-либо вмешательство в процесс шифрования.
* использует аппаратный датчик случайных чисел, который гарантирует случайность генерации ключей шифрования и повышает качество реализации криптографических алгоритмов.
* позволяет напрямую загружать ключи шифрования в шифропроцессор, минуя оперативную память компьютера (в программной реализации ключи находятся в памяти во время работы шифратора).

## Обзор аналогов разрабатываемого устройства

Большая часть устройств для аппаратного шифрования реализована в виде PCI плат расширения или приборов типа USB-ключ. Производителями подобных продуктов являются, в основном, Россия, США и Япония.

### “Криптон”

В России лидером по производству шифровального оборудования является компания “Анкад”. Одной из ведущих разработок этой фирмы является линейка устройств под названием “Криптон”. Все устройства взаимодействуют с компьютером через интерфейс PCI. Программное обеспечение устройств “Криптон” позволяет:

* шифровать компьютерную информацию (файлы, группы файлов и разделы дисков), обеспечивая их конфиденциальность;
* осуществлять электронную цифровую подпись файлов, проверяя их целостность и авторство;
* создавать прозрачно шифруемые логические диски, максимально облегчая и упрощая работу пользователя с конфиденциальной информацией;
* формировать криптографически защищенные виртуальные сети, шифровать IP-трафик и обеспечивать защищенный доступ к ресурсам сети мобильных и удаленных пользователей;
* создавать системы защиты информации от несанкционированного доступа и разграничения доступа к компьютеру.

Основные технические данные и характеристики

Алгоритм шифрования ГОСТ 28147-89

Размерность ключа шифрования, бит 256 Количество уровней ключевой системы 3

Датчик случайных чисел аппаратный (аттестован экспертной организацией)

Шина PCI (Bus Master, Target)

Реализация алгоритма шифрования аппаратная

Скорость шифрования, Кбайт/с до 8500

Носители ключей дискеты, смарт-карты

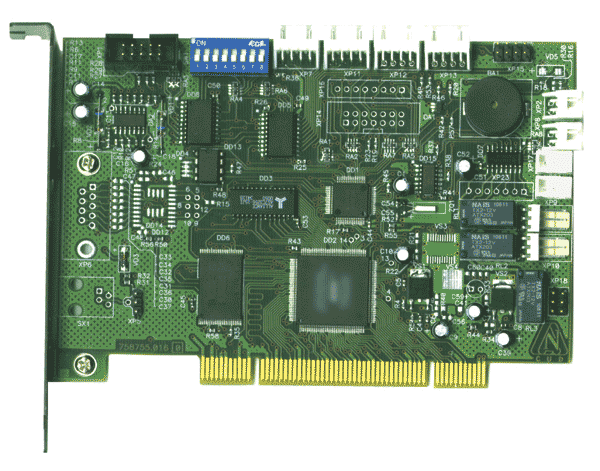


Рис. 1.1 – Внешний вид устройства “Криптон-7”

### 

### 1.2.2 “ruToken”

Другим устройством, обеспечивающим аппаратную криптографическую защиту данных, является электронный брелок “ruToken”. Брелок взаимодействует с компьютером через интерфейс USB. “ruToken” может использоваться для решения следующих задач:

* Аутентификация − замена парольной защиты при доступе к БД, Web-серверам, VPN-сетям и security-ориентированным приложениям на программно-аппаратную аутентификацию; − защищенные соединения при доступе к почтовым серверам, серверам баз данных, Web-серверам, файл-серверам, аутентификации при удаленном доступе.
* Защита данных − защита информации (шифрование по ГОСТ 28147-89); − защита электронной почты (ЭЦП, шифрование); − защита доступа к компьютеру (авторизация пользователя при входе в операционную систему).



Рис. 1.2 – Внешний вид устройства ruToken

Общие технические характеристики ruToken:

* Аппаратное шифрование по ГОСТ 28147-89;
* Файловая система по ISO 7816;
* 8, 16 или 32 Кбайт энергонезависимой памяти;
* Поддержка PC/SC, PKCS#11, MS CryptoAPI, X.509.
* Базируется на защищенном микроконтроллере;
* Интерфейс USB (USB 1.1 / USB 2.0);
* EEPROM память 8, 16 и 32 Кб;
* 2-факторная аутентификация (по факту наличия ruToken и по факту предъявления PIN-кода;
* 32-битовый уникальный серийный номер;
* Поддержка ОС Windows 98/ME/2000/XP/2003;
* Поддержка стандартов ISO/IEC 7816, PC/SC, ГОСТ 28147-89, MS CryptoAPI и MS SmartcardAPI;
* Возможность интеграции в любые smartcard-ориентированные программные продукты (e-mail-, internet-, платежные системы и т.п.).

## Аспекты создания устройства для аппаратного шифрования

### Структура аппаратного шифратора

В состав аппаратного шифратора входят блок управления, шифропроцессор, аппаратный датчик случайных чисел, контроллер, микросхемы памяти, переключатели режимов работы и интерфейсы для подключения ключевых носителей (рис. 1.3).

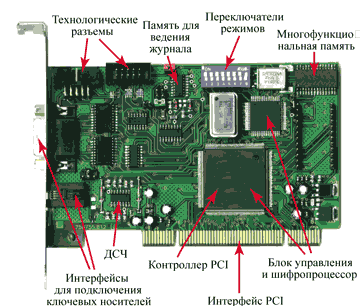


Рис. 1.3 – Структура устройства криптографической защиты данных "Криптон-9"

Блок управления, служит для управления работой всего шифратора. Обычно он реализован на базе микроконтроллера. Шифропроцессор представляет собой специализированную микросхему или микросхему программируемой логики (PLD - Programmable Logic Device), которая выполняет шифрование данных. Для генерации ключей шифрования в устройстве предусмотрен аппаратный датчик случайных чисел (ДСЧ), вырабатывающий статистически случайный и непредсказуемый сигнал, преобразуемый затем в цифровую форму. Обмен командами и данными между шифратором и компьютером обеспечивается контроллером. Для хранения программного обеспечения микроконтроллера необходима энергонезависимая память, реализованная на одной или нескольких микросхемах. Это же внутреннее ПЗУ используется для записи журнала операций и других целей.

При хранении ключевой информации на дискете ее считывание производится через системную шину компьютера и существует возможность перехвата. Поэтому аппаратные шифраторы обычно снабжают интерфейсом для непосредственного подключения устройств хранения ключей. Наиболее распространенные среди них - разъемы для подключения считывателей смарт-карт (рис. 1.4) и разъемов для работы с электронными таблетками Touch Memory.



Рис. 1.4 – Устройство для считывания смарт-карт SR-210

Помимо функций шифрования информации, каждый шифратор должен обеспечивать:

* выполнение различных операций с ключами шифрования: их загрузку в шифропроцессор и выгрузку из него, а также взаимное шифрование ключей;
* расчет имитоприставки для данных и ключей (имитоприставка представляет собой криптографическую контрольную сумму, вычисленную на определенном ключе);
* генерацию случайных чисел по запросу.

### Ключевые схемы и процесс шифрования файлов

Аппаратные шифраторы должны поддерживать несколько уровней ключей шифрования. Обычно реализуется трехуровневая иерархия ключей. Большее количество уровней, не дает заметного улучшения качества защиты, а меньшего может не хватить для ряда ключевых схем. Трехуровневая иерархия предусматривает использование сеансовых или пакетных ключей (1-й уровень), долговременных пользовательских или сетевых ключей (2-й уровень) и главных ключей (3-й уровень).

Каждому уровню ключей соответствует ключевая ячейка памяти шифропроцессора. Подразумевается, что шифрование данных выполняется только на ключах первого уровня, остальные предназначены для шифрования самих ключей при построении различных ключевых схем.

Трехуровневую схему лучше всего иллюстрирует упрощенный пример процесса шифрования файла (рис. 1.5). На этапе начальной загрузки в ключевую ячейку № 3 заносится главный ключ. Сеансовый ключ генерируется в результате запроса к ДСЧ шифратора на получение случайного числа, которое загружается в ключевую ячейку № 1, соответствующую сеансовому ключу. С его помощью шифруется содержимое файла и создается новый файл, хранящий зашифрованную информацию.

Далее у пользователя запрашивается долговременный ключ, который загружается в ключевую ячейку № 2 с расшифровкой посредством главного ключа, находящегося в ячейке № 3. Шифратор должен иметь режим расшифровки одного ключа с помощью другого внутри шифропроцессора. В этом случае ключ в открытом виде никогда не покидает устройство. Сеансовый ключ зашифровывается при помощи долговременного ключа, находящегося в ячейке № 2, выгружается из шифратора и записывается в заголовок зашифрованного файла.

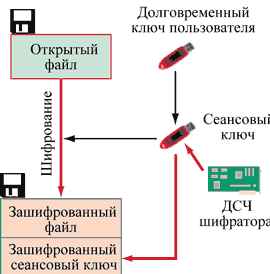


Рис. 1.5 – Шифрование файла

При расшифровке файла сначала с помощью долговременного ключа пользователя расшифровывается сеансовый ключ, а затем с его помощью восстанавливается информация.

Преимущества многоуровневой ключевой схемы:

* снижается нагрузка на долговременный ключ - он используется только для шифрования коротких сеансовых ключей; это усложняет потенциальному злоумышленнику криптоанализ зашифрованной информации с целью получения долговременного ключа.
* при смене долговременного ключа можно быстро перешифровать файл: достаточно перешифровать сеансовый ключ со старого долговременного на новый.
* разгружается ключевой носитель - на нем хранится только главный ключ, а все долговременные ключи могут храниться в зашифрованном с помощью главного ключа виде даже на жестком диске ПК.

### Хэширование паролей

При разработке любого криптоалгоритма следует учитывать, что в половине случаев конечным пользователем системы является человек, а не автоматическая система. Необходимо принимать во внимание тот факт, что пользователю придется запоминать ключ длиной до 4096 битов (512 ASCII символов), а предел запоминаемости чисел человеческим мозгом лежит на границе 12-15 символов [1].

Для решения этой проблемы были разработаны методы, преобразующие осмысленную строку произвольной длины – пароль, в указанный ключ заранее заданной длины. Для этой операции используются хэш-функции (от англ. hashing – мелкая нарезка и перемешивание).

Хэш-функцией называется такое математическое или алгоритмическое преобразование заданного блока данных, которое обладает следующими свойствами:

* бесконечная область определения;
* конечная область значений;
* необратимость;
* лавинный эффект (изменение входного потока информации на один бит меняет около половины всех бит выходного потока);

Эти свойства позволяют подавать на вход хэш-функции пароли произвольной длины и, ограничив область значений функции диапазоном 0..2N-1, где N – длина ключа в битах, получать на выходе достаточно равномерно распределенные по области значения блоки информации – ключи.

Размер пароля ограничен исключительно используемым алгоритмом хэширования. Например, при использовании хэш-функции SHA2, с длиной результата в 384 бита, на вход можно подавать число меньшее бита. Если пароль – строка ASCII символов, то длина пароля в этом случае ограничена  символами. Это значит, что паролем могут быть строки текста, цитаты или стихотворения, что значительно улучшает запоминание пароля.

### Длина ключа, используемого в криптосистеме

Любой шифровальный алгоритм с использованием ключа, может быть вскрыт методом перебора всех значений ключа. Если ключ подбирается методом грубой силы (brute force), требуемая мощность компьютера растет экспоненциально с увеличением длины ключа. Ключ длиной в 32 бита требует для вскрытия  шагов. Системы с 40-битным ключом требуют  шагов. Системы с 56-битными ключами могут быть легко вскрыты с помощью специальной аппаратуры (используя суперкомпьютер стоимостью 250 тыс. долларов, сотрудники RSA Laboratory "взломали" утвержденный правительством США алгоритм шифрования данных DES менее чем за три дня - рис.1.6). Полный перебор ключа длиной 64 бита для RC5 в настоящее время продолжается. Ключи длиной 80 бит могут в будущем стать уязвимыми. Ключи длиной 128 бит вероятно останутся недоступными для вскрытия методом грубой силы в обозримом будущем. Можно использовать и более длинные ключи.



Рис. 1.6 – Микропроцессор и плата суперкомпьютера DES Cracker

Длины ключей, используемых в криптографии с открытым ключом обычно значительно больше, чем в симметричных алгоритмах. Здесь проблема заключается не в подборе ключа, а в воссоздании секретного ключа по открытому. В алгоритме RSA проблема эквивалентна разложению на множители большого целого числа, которое является произведением пары неизвестных простых чисел. В случае других криптосистем (DSA, Эльгамаль), проблема эквивалентна вычислению дискретного логарифма по модулю большого целого числа (такая задача считается примерно аналогичной по трудности задаче разложения на множители).

В 2007 году, группа Швейцарских ученых под руководством Арьена Ленстра осуществила факторизацию 700 битного ключа. Арьен Ленстра считает, что взлом 1024 битного RSA шифра станет возможным через 5-10 лет. Ключи длиной в 2048 и 4096 бит, криптоаналитики считают надежными на десятилетия.

Для создания надежной криптосистемы, специалисты в области криптографии, в частности Брюс Шнайер [1], рекомендуют использовать 256 битный ключ для симметричных алгоритмов и 2048 бит для алгоритмов с открытым ключом.

### Проблемы, возникающие при шифровании файлов

Особенности шифрования файлов, о которых необходимо помнить вне зависимости от применяемого криптографического алгоритма:

* после шифрования файла его незашифрованная копия может остаться на магнитном диске, другом компьютере или в виде распечатки;
* размер блока в блочном алгоритме шифрования может значительно превышать размер отдельной порции данных в структурированном файле, в результате чего зашифрованный файл окажется намного длиннее исходного;
* если пользователь использует один и тот же ключ для шифрования всех файлов, то в результате у криптоаналитика будет много шифротекста, полученного на одном ключе, что существенно облегчит вскрытие этого ключа.

Лучше шифровать каждый файл на отдельном ключе, а затем зашифровать ключи при помощи мастер ключа. Пользователи будут избавлены от суеты, связанной с организацией надежного хранения множества ключей, т.к. разграничение доступа к различным файлам будет осуществляться путем деления множества всех сеансовых ключей на подмножества и шифрования этих подмножеств на различных мастер-ключах.

Ключи, применяемые для шифрования файлов, необходимо генерировать случайным образом. Это увеличит защищенность криптосистемы.

## Выбор алгоритма шифрования

Криптографический алгоритм, также называемый шифром или алгоритмом шифрования, представляет собой математическую функцию, используемую для шифрования и расшифровки.

Алгоритм шифрования называется ограниченным, если надежность криптографического алгоритма обеспечивается за счет сохранения в тайне сути самого алгоритма.

Ограниченные алгоритмы непригодны при современных требованиях, предъявляемых к шифрованию, потому что:

* каждая группа пользователей, желающих обмениваться секретными сообщениями, должна разработать оригинальный алгоритм шифрования;
* невозможно применение готового оборудования и стандартных программ;
* необходимо разрабатывать собственный криптографический алгоритм, каждый раз, когда кто-то из пользователей группы захочет ее покинуть или когда детали алгоритма случайно станут известны посторонним.

В криптографии, описанные проблемы решаются при помощи использования ключа, который обозначается буквой «К» (от английского слова key). Ключ должен выбираться среди значений, принадлежащих множеству, которое называется ключевым пространством. И функция шифрования «Е», и функция расшифровки «D» зависят от ключа. Этот факт выражается присутствием «К» в качестве подстрочного индекса у функций «Е» и «D»:

Е к (Р) = С (1.1)

D к (С) = Р (1.2)

Справедливо следующее тождество:

D k (E k (P)) = P (1.3)

Некоторые алгоритмы шифрования используют различные ключи для шифрования и расшифровки. Это означает, что ключ шифрования «К1»отличается от ключа расшифровки «К2». В этом случае справедливы следующие соотношения:

Е k1 ( P ) = С (1.4)

D k2 (С) = Р (1.5)

D k2 (E k1 (Р) ) = Р (1.6)

Надежность алгоритма шифрования с использованием ключей достигается за счет их надлежащего выбора и последующего хранения в секрете. Это означает, что такой алгоритм не требуется держать в тайне. Знание криптографического алгоритма не позволит злоумышленнику прочесть зашифрованные сообщения, поскольку он не знает секретный ключ, использованный для их шифрования.

Под криптосистемой понимается алгоритм шифрования, а также можно всевозможных ключей, открытых и шифрованных текстов.

Существуют две разновидности алгоритмов шифрования с использованием ключей — симметричные и с открытым ключом.

### Симметричные алгоритмы шифрования

Симметричным называют криптографический алгоритм, в котором ключ, используемый для шифрования сообщений, может быть получен из ключа для расшифровки и наоборот. Одноключевыми или алгоритмами с секретным ключом называются симметричные алгоритмы, в которых применяют всего один ключ и требуют, чтобы отправитель сообщений и их получатель условились о том, каким ключом они будут пользоваться. Надежность одноключевого алгоритма определяется выбором ключа, поскольку его значение дает возможность злоумышленнику без помех расшифровывать все перехваченные сообщения. Поэтому выбранный ключ следует хранить в тайне от посторонних.

Симметричные алгоритмы шифрования бывают двух видов. Одни из них обрабатывают открытый текст побитно. Они называются потоковыми алгоритмами, или потоковыми шифрами. Согласно другим, открытый текст разбивается на блоки, состоящие из нескольких бит. Такие алгоритмы блочными шифрами. В современных компьютерных алгоритмах блочного шифрования обычно длина блока составляет 64 бита или 128 бит.

### Блочные алгоритмы

### Тройной DES

Базовый алгоритм DES был разработан фирмой IBM в середине 1970-х годов. Через несколько лет, DES был принят в качестве государственного стандарта шифрования США. Тройной DES - это вариация DES, в которой базовый алгоритм выполняется трижды на одном блоке данных. Длина общего ключа – 168 бит (в DES – 56 бит). Алгоритм оперирует 64-битовыми блоками данных. Расчётная стойкость такого алгоритма к лобовой атаке составляет 112 бит.

### AES

"Улучшенный стандарт шифрования" (Advanced Encryption Standard). Алгоритм принят национальным институтом стандартов и технологий (NIST) США в 1999 году в качестве стандарта шифрования важных несекретных коммуникаций. Пришел на смену устаревшему DES. Авторское название – Rijndael ("Рэндал"). Это блочный шифр со 128-, 192- или 256-битовым размером ключа и 128-битовым блоком.

AES подвергся тщательному исследованию государственного и гражданского криптологического сообщества. Rijndael обладает простым дизайном, облегчающему его реализацию, малым размером исполняемого кода и нетребователен к объему памяти.

### CAST

Представляет собой шифр со 128-битовым ключом и 64-битовым блоком. Дизайн основан на формальной архитектуре DES с доказанной стойкостью. Не имеет слабых ключей. Алгоритм совершенно устойчив к линейному и дифференциальному криптоанализу. Может быть взломан только методом прямого перебора.

S-блоки, используемые в алгоритме для противодействия дифференциальному криптоанализу, не являются фиксированными и не зависят от ключа, но проектируются индивидуально для каждого приложения, используя специальные инструкции авторов шифра. Канада, где был разработан алгоритм, использует его в качестве государственного стандарта шифрования.

### Twofish

Один из пяти финалистов на звание AES. Группу разработчиков возглавлял Брюс Шнайер. В реализации использует 256-битовый ключ и 128-битовый блок данных.

Twofish оказался самым стойким в исследовании, хотя и одним из самых медленных. Однако скорость обычно играет решающую роль только в приложениях реального времени, к которым шифрование файлов и почты не относится.

### Blowfish

Автором алгоритма является Брюс Шнайер. Алгоритм представляет собой блочный шифр с ключом переменной длины (вплоть до 448 бит), оперирующий на 64-битовых блоках.

При проектировании использовалась формальная схема DES (т.е. сеть Файсталя) с 16 раундами. Это упрощает анализ алгоритма и гарантирует отсутствие в нём неочевидных уязвимостей. Blowfish имеет большой запас прочности и поддаётся криптоанализу только в сильно ослабленных вариантах. Имеет небольшое пространство слабых ключей, вероятность выбора которых ничтожно мала.

Автор алгоритма, Брюс Шнайер, рекомендует использовать Blowfish во встраиваемых системах анализа, обработки и преобразования данных.

### Потоковые шифры

RC4

Это потоковый шифр, широко применяющийся в различных системах защиты информации в компьютерных сетях (например, в протоколе SSL и для шифрования паролей в Windows NT). Шифр разработан компанией RSA Security Inc. Для его использования требуется лицензия.

Основные преимущества шифра — высокая скорость работы и переменный размер ключа. Типичная реализация выполняет 19 машинных команд на каждый байт текста.

В США длина ключа для использования внутри страны рекомендуется равной 128 битов. Алгоритм имеет специальный статус, который означает, что разрешено экспортировать реализации RC4 с длинной ключа до 40 бит.

RC4 в 10 раз быстрее DES и устойчив к криптоанализу. S-блок медленно изменяется при использовании.

WAKE

Алгоритм WAKE (Word Auto Key Encryption) выдаёт поток 32-битных слов, которые с помощью XOR могут быть использованы для получения шифротекста. Для генерации следующего слова ключа используется предыдущее слово шифротекста. Это быстрый алгоритм. Алгоритм использует S блок из 256 32-битовых значений. Этот S-блок обладает одним особым свойством: старший байт всех элементов представляет собой перестановку всех возможных байтов, а 3 младших байта случайны.

Самым ценным качеством WAKE является его скорость. WAKE чувствителен к вскрытию с выбранным открытым текстом или выбранным шифротекстом.

### Алгоритмы шифрования с открытым ключом

Назначение их то же, что и у блочных шифров – сделать информацию непонятной всякому постороннему. Основное отличие состоит в использовании для операций шифрования двух разных, но взаимосвязанных ключей однонаправленного действия, один из которых может зашифровать информацию, но расшифровать её может только другой.

Благодаря этой особенности некоторые алгоритмы с открытым ключом совместно с хэш-функцией могут применяться и для другой цели: для выработки имитовставки (электронной цифровой подписи), подтверждающей авторство информации. Асимметричные алгоритмы основаны на ряде математических проблем (т.н. NP-полных задач), на которых и базируется их стойкость. Пока учёные-математики не найдут решение этих проблем, данные алгоритмы будут стойки. В этом заключается ещё одно отличие симметричного и асимметричного шифрования: стойкость первого является непосредственной и научно доказуемой, стойкость второго – феноменальной, т.е. основанной на некоем явлении, и научно не доказана (так же, как не доказана их нестойкость).

### RSA

Это криптографическая система с открытым ключом, обеспечивающая оба механизма защиты: шифрование и цифровую подпись. Криптосистема RSA была разработана в 1977 году и названа в честь авторов: Рональда Ривеста, Ади Шамира и Леонарда Адельмана.

Принцип действия RSA состоит в следующем. Для начала сгенерируем пару ключей:

1. Возьмём два больших случайных простых числа p и q (т.е. числа делящихся только на себя и на 1) приблизительно равной разрядности, и вычислим их произведение

n = p∙q (1.7)

1. Выберем число e, взаимно простое с произведением (p–1)\*(q–1). Взаимно простыми называют числа, у которых нет общих множителей кроме 1 (например, 15 и 28 – являются, 15 и 27 – нет: кроме 1 их общий множитель – 3).
2. Вычисляется число d, взаимно простое с n.

d = e-1mod((p–1)∙(q–1)) (1.8)

Числа e и n становятся открытым ключом. Число d – закрытым. Чтобы создать шифротекст c из сообщения m, необходимо выполнить:

c = me mod n (1.9)

Чтобы расшифровать полученный шифротекст, необходимо выполнить:

m = cd mod n (1.10)

Пока не найдены эффективные методы разложения чисел на множители, невозможно факторизовав n получить p и q, а, следовательно, и показатель закрытого ключа d. Таким образом, надежность криптосистемы RSA базируется на трудноразрешимой задаче разложения n на множители. Несмотря на фактическую сложность разложения больших чисел на множители, научно не доказано, что факторизация является трудной, или NP-полной, задачей. Доказательств обратного тоже никто не представил.

### Выбор алгоритма шифрования

Выбирая алгоритм шифрования, который будет использоваться в создаваемой криптосистеме, прежде всего, необходимо обратить внимание на следующие характеристики алгоритмов:

* Криптостойкость. Алгоритм должен быть тщательно проанализирован мировым криптографическим сообществом в течение длительного времени (не менее пяти лет лет) [1] и признан криптостойким к различным видам атак;
* Длина ключа. Ключ, используемый в алгоритме шифрования, должен быть не короче 256 бит для алгоритмов симметричного шифрования и 2048 бит для алгоритмов с открытым ключом. Это сделано для того, чтобы шифр невозможно было вскрыть методом прямого перебора (грубой силой) в ХХI веке;
* Скорость шифрования. Предполагается взаимодействие устройства с компьютером через полноскоростной интерфейс USB2.0.(12 Мбит/сек). Поэтому скорость шифрования данных по выбранному алгоритму должна быть настолько высокой, чтобы не возникало простоев при передаче данных на максимальной скорости.
* Ресурсоемкость. Алгоритм должен быть оптимизирован для аппаратной реализации. Количество оперативной памяти и необходимая производительность микропроцессора, должны находиться в рамках, которые ограничивают микроконтроллеры общего применения.

### Выбор осуществлялся из следующего набора алгоритмов: RC6, Rijndael(Рэндал), Serpent и Twofish, потоковые шифры RC4 и WAKE, алгоритм Blowfish, разработанный известным криптоаналитиком Брюсом Шнайером, а также алгоритм с открытым ключом – RSA.

Все вышеперечисленные алгоритмы в различной степени отвечают предъявляемым к ним требованиям. Для реализации RSA, необходимы операции (возведение в степень) над большими числами, для быстрого выполнения которых в проекте придется применять специализированные микросхемы. Алгоритмы с открытым ключом не применимы из-за низкой скорости. Потоковые шифры лучше не использовать. Они лучше подходят для шифрования потока информации. Например, пакетов информации в компьютерных сетях, либо разговоров по телефонной линии.

Поскольку разрабатываемое устройство предназначено для шифрования файлов, необходимо использовать блочный шифр, а не потоковый. Два кандидата на эту роль – Rijndael и Blowfish. Создатель Blowfish, Брюс Шнайер рекомендует этот шифр для использования в системах, построенных на основе микроконтроллера. Длина ключа используемого в Blowfish переменная, с верхним пределом 448 бит (в Rijndael максимум – 256 бит). К тому же на алгоритм Blowfish отсутствует лицензия и его можно свободно использовать. Поэтому в устройстве будет реализован алгоритм Blowfish.

## Описание алгоритма Blowfish

Blowfish – это алгоритм, предназначенный для реализации в микроконтроллерах [1]. При проектировании Blowfish использовались следующие критерии:

* Скорость. Blowfish шифрует данные на 32-битовых микропроцессорах со скоростью 26 тактов на байт.
* Компактность. Blowfish может работать менее, чем в 5 Кбайт памяти.
* Простота. Blowfish использует только простые операции: сложение, XOR и выборка из таблицы по 32-битовому операнду. Анализ его схемы несложен, это уменьшает количество ошибок при реализации.
* Настраиваемая безопасность. Длина ключа Blowfish переменна и может достигать 448 битов.

Blowfish оптимизирован для тех приложений, в которых нет частой смены ключей. При реализации на 32-битовых микропроцессорах с большим кэшем данных, Blowfish заметно быстрее DES. Blowfish не подходит для использования в приложениях с частой сменой ключей, например, при коммутации пакетов, или для использования в качестве однонаправленной хэш-функции.

Blowfish представляет собой 64-битовый блочный шифр с ключом переменной длины. Алгоритм состоит из двух частей: развертывание ключа и шифрование данных. Развертывание ключа преобразует ключ длиной до 448 битов в несколько массивов подключей, общим объемом 4168 байтов.

Шифрование по алгоритму Blowfish состоит из функции преобразования данных, последовательно выполняемой 16 раз. Каждый этап состоит из зависимой от ключа перестановки и зависимой от ключа и данных подстановки. Используются только сложения и XOR 32-битовых слов. Единственными дополнительными операциями на каждом этапе являются четыре извлечения данных из индексированного массива.

В Blowfish используется много подключей. Эти подключи должны быть рассчитаны до начала шифрования или дешифрирования данных.

P-массив состоит из 18-ти 32-битовых подключей:



Каждый из четырех 32-битовых S-блоков содержит 256 элементов:



Метод, используемый при вычислении этих подключей, описан в этом разделе ниже.

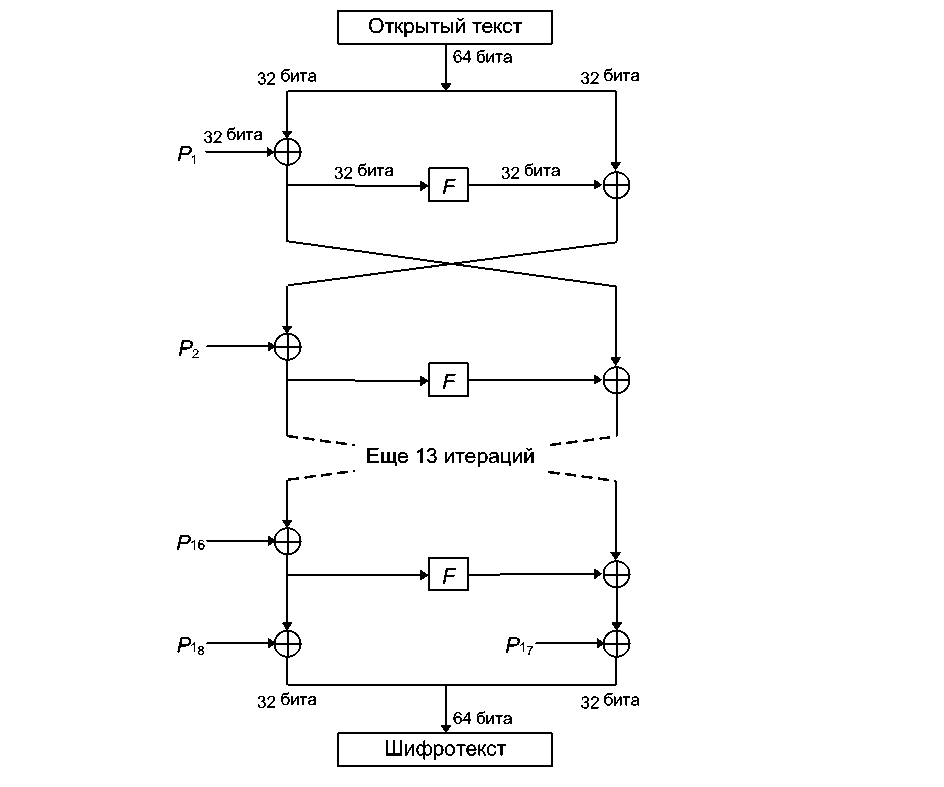


Рис. 1.7 – Алгоритм Blowfish

Blowfish является сетью Фейстела (Feistel), состоящей из 16 этапов. На вход подается 64-битовый элемент данных x.

Алгоритм шифрования:

* Элемент x разбивается на две 32-битовых половины:  и ;
* Для этапов с первого по шестнадцатый, выполняется:

 (1.11)

 (1.12)

Переставить  и (кроме последнего этапа);

* В последнем этапе, производится:

 (1.13)

 (1.14)

* Объединяются элементы  и ;

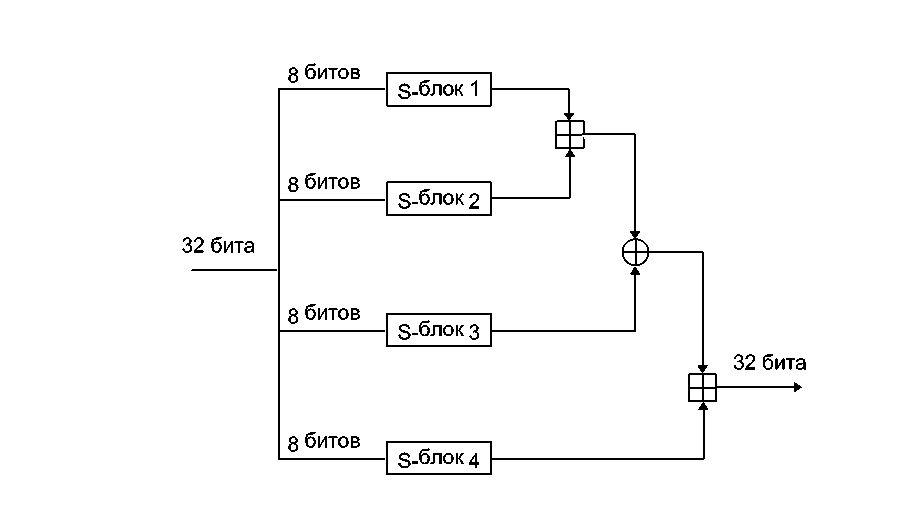


Рис. 1.8 – Функция F

Функция F (рис.1.8) представляет собой последовательность следующих действий:

* Разделить  на четыре 8-битовых части: a, b, c и d;
* Выполнить над a,b,c,d :

 (1.15)

Дешифрирование выполняется также, как и шифрование, но  используются в обратном порядке.

В реализациях Blowfish, для которых требуется очень большая скорость, цикл должен быть развернут, а все ключи должны храниться в КЭШе данных.

Подключи рассчитываются с помощью специального алгоритма. Вот какова точная последовательность действий.

1. Сначала P-массив, а затем четыре S-блока по порядку инициализируются фиксированной строкой. Эта строка состоит из шестнадцатеричных цифр .
2. Выполняется XOR P1 с первыми 32 битами ключа, XOR P2 со следующими 32 битами ключа, и так далее для всех битов ключа (до P18). Используется циклически, пока для всего P-массива не будет выполнена операция XOR с битами ключа.
3. Используя подключи, полученные на этапах (1) и (2), алгоритмом Blowfish шифруется строка из одних нулей.
4. P1 и P2 заменяются результатом этапа (3).
5. Результат этапа (3) шифруется с помощью алгоритма Blowfish и измененных подключей.
6. P3 и P4 заменяются результатом этапа (5).
7. Далее в ходе процесса все элементы P-массива и затем по порядку все четыре S-блока заменяются выходом постоянно меняющегося алгоритма Blowfish.

Серж Воденэ (Serge Vaudenay) исследовал Blowfish с известными S-блоками и r этапами. Дифференциальный криптоанализ может раскрыть P-массив с помощью 28r+1 выбранных открытых текстов. Для некоторых слабых ключей, которые генерируют плохие S-блоки (вероятность выбора такого ключа составляет 1 к 214), это же вскрытие раскрывает P-массив с помощью всего 24r+1. При неизвестных S-блоках это вскрытие может обнаружить использование слабого ключа, но не может определить сам ключ (ни S-блоки, ни P-массив). Это вскрытие эффективно только против вариантов с уменьшенным числом этапов и совершенно бесполезно против 16-этапного Blowfish.

Слабым является ключ, для которого два элемента данного S-блока идентичны. До выполнения развертывания ключа невозможно определить, является ли он слабым.

В устройстве осуществляется проверка ключей на принадлежность к классу слабых. Слабые ключи не используются.

В устройстве реализован Blowfish c количеством этапов, равным 16.

До сих пор неизвестно об успешном криптоанализе Blowfish.

## Выбор функции для хэширования паролей

Целесообразность хэширования паролей была показана в пункте 1.3.3. Ниже приведены используемые на сегодняшний день функции хэширования.

### 

### 1.6.1 MD4

MD4 - это однонаправленная хэш-функция. MD обозначает Message Digest (краткое изложение сообщения). Алгоритм для входного сообщения выдает 128-битовое хэш-значение. MD4 подходит для высокоскоростных программных реализаций. Она основана на простом наборе битовых манипуляций с 32-битовыми операндами.

После первого появления алгоритма Берт Боер и Антон Босселаерс (Antoon Bosselaers) осуществили криптоанализ последних двух из трех этапов алгоритма. Эли Бихам рассмотрел использование дифференциального криптоанализа против первых двух этапов MD4.

1.6.2 MD5

MD5 - это улучшенная версия MD4. Алгоритм хэш-функции сложнее чем в MD4, но их схемы похожи. Результатом MD5 является 128-битовое хэш-значение. Группа китайских ученых показала, что существует простой и быстрый алгоритм подбора коллизий этой хэш-функции.

Алгоритмы MD4 и MD5 в проектах лучше не использовать. Существует возможность вычисления коллизий этих функций.

### 

### 1.6.3 SHA

Алгоритм безопасного хэширования (Secure Hash Algorithm, SHA), разработан для стандарта цифровой подписи (Digital Signature Standard).

Для любого входного сообщения длиной меньше 264 битов SHA выдает 160-битовый результат, называемый кратким содержанием сообщения. SHA является криптостойким алгоритмом и разработан так, чтобы было невозможно найти сообщение, соответствующее данному краткому содержанию сообщения. Любые изменения, произошедшие при передаче сообщения, с очень высокой вероятностью приведут к изменению краткого содержания сообщения.

#### 12.08.2004 найдена полная коллизия SHA-0. На это потребовалось 50 000 часов машинного времени [1].

### 

### 1.6.4 SHA1

Алгоритм разработан в 1995 году в качестве замены более слабого SHA. Длина результата 160 бит.

Группе китайских ученых удалось найти коллизии в SHA1 меньше чем за 269 операций хеширования (для вскрытия грубой силой не обходимо произвести 280 операций). В связи с этим NIST рекомендует пока пользоваться SHA2 – алгоритм, похожий на тот, что используется в SHA1, но с длиной выходного сообщения 256, 384 и 512 бит. Это делает SHA2 более устойчивым к вскрытию полным перебором.

### 1.6.5 SHA2

Алгоритм представляет собой вариант SHA1, с большей длиной выходного сообщения. Существуют варианты SHA-256, SHA-384, SHA-512. Об успешных атаках на SHA2 пока неизвестно.

В устройстве используется хэш-функция SHA2-384, так как:

* это одна из немногих криптостойких функций хэширования на сегодняшний день;
* длина выходного сообщения функции равняется 384 бита. SHA2 будет использоваться для хэширования паролей. Полученный, с помощью SHA2-384, ключ к Blowfish полностью соответствует требованиям, предъявляемым к длине ключей.

## Выбор генератора случайных чисел

Основным требованием, предъявляемым к разрабатываемому устройству, является высокая криптостойкость системы шифрования. Поэтому для генерации сеансовых ключей шифрования необходимо использовать генератор случайных чисел.

### 

### 1.7.1 Методы получения случайных чисел

При использовании как симметричных, так и асимметричных криптосистем необходимо иметь хороший источник случайных чисел для создания ключей. Главная особенность такого источника состоит в том, что создаваемые числа должны быть неизвестными и непредсказуемыми для криптоаналитика.

Лучшим вариантом является создание случайных чисел на основе некоторого физического процесса, так как многие физические процессы действительно случайны. Например, для этого можно использовать аппаратные средства типа “шумящего” диода. Можно использовать какие-либо физические движения пользователя, например, скорость печати в микросекундах, перемещения мыши и т.д.

Практически все методы генерации случайных чисел имеют некоторую корреляцию и не позволяют обеспечить достаточную статистическую хаотичность. Поэтому перед использованием, полученные ключи следует обрабатывать надежной хэш-функцией. (См. пункт 1.6.)

Другой подход состоит в том, чтобы использовать генератор псевдослучайных чисел, запускаемый случайным числом. Главное различие между случайными и псевдослучайными числами в том, что последние обязательно являются периодическими, а действительно случайные числа нет.

Так как генераторы псевдослучайных чисел являются детерминированными алгоритмами, то важно найти среди них криптографически защищенный, а кроме этого использовать хорошее случайное число для запуска генерации.

### 1.7.2 Линейные конгруэнтные генераторы

Линейные конгруэнтные генераторы являются генераторами псевдослучайных чисел, которые вычисляются по формуле:

Xn = (a∙Xn-1 + b) mod m, (1.16)

в которых Xn - это n-ый член последовательности, а Xn-1 - предыдущий член последовательности. Переменные a>0, b>0 и m>0 целочисленные константы: a - множитель, b- инкремент, и m - модуль.

Получаемая последовательность зависит от выбора стартового числа X0 и при разных его значениях получаются различные последовательности случайных чисел.

Период такого генератора не больше, чем m. Если a, b и m выбраны правильно, то генератор будет генератором с максимальным периодом и его период будет равен m.

Преимуществом линейных конгруэнтных генераторов является их быстрота за счет малого количества операций на бит.

Линейные конгруэнтные генераторы нельзя использовать в криптографии, так как они предсказуемы. Впервые линейные конгруэнтные генераторы были взломаны Джимом Ридсом (Jim Reeds), а затем Джоан Бояр (Joan Boyar).

Другие исследователи расширили идеи Джоан Бояр, разработав способы вскрытия любого полиномиального генератора. Таким образом была доказана бесполезность конгруэнтных генераторов для криптографии.

### 

### 1.7.3 Алгоритм Блюма-Блюма-Шуба (Blum Blum Shub, BBS)

Алгоритм представляет собой генератор псевдослучайных чисел, предложенный в 1986 году Ленор Блюм, Мануэлем Блюм и Майклом Шубом.

Алгоритм BBS выглядит так:

xn+1 = (xn)2 mod M, (1.17)

где M=p∙q является произведением двух больших простых чисел p и q. На каждом шаге алгоритма выходные данные получаются из xn путем взятия либо бита четности, либо нескольких наименее значимых бит из xn .

Алгоритм Блюма-Блюма-Шуба рекомендуется использовать только в криптографии. Этот метод имеет необычно высокую стойкость, которая обеспечивается качеством генератора исходя из вычислительной сложности задачи факторизации чисел. Вычисление выходных бит настолько же трудно, как и факторизация M [1].

Генератор Блюма-Блюма-Шуба нецелесообразно использовать в разрабатываемом устройстве из-за низкого быстродействия алгоритма. В этом генераторе используются арифметические операции над большими числами, реализация которых на микроконтроллере займет значительное время.

### 1.7.4 Генератор, используемый в разрабатываемом устройстве

В устройстве будут использоваться случайные числа, полученные от аппаратуры микроконтроллера и преобразовываться в 384 – битное число с помощью хэш-функции SHA2-384. Это необходимо делать для улучшения равномерности распределения случайных чисел.

Для получения случайных чисел, в устройстве используются 32-х разрядный счетчик, на вход которого подается максимально возможная частота. При включении устройства, счетчик инициализируется значением сохраненным ранее в EEPROM. Поскольку операции шифрования файлов инициируются пользователем в случайные промежутки времени, в начале каждой такой операции содержимое счетчика подается на вход функции хэширования. Полученное значение используется в качестве сеансового ключа, а его младшие 32 бита служат для задания нового значения счетчика.

## Структурная схема разрабатываемого устройства

Проанализировав техническое задание, составим структурную схему устройства:

* Устройство предназначено для аппаратного шифрования компьютерных файлов. Поэтому устройство будет использоваться в составе с персональным компьютером.
* Устройство будет производить шифрование данных с большой скоростью (до 12 Мбит/сек). Поэтому основой устройства должен быть высокопроизводительный 32-х разрядный микроконтроллер.
* Устройство связано с компьютером через интерфейс USB на скорости 12 Мбит/сек. Поэтому микроконтроллер, используемый в устройстве, должен быть оснащен full-speed USB контроллером.
* Для предотвращения влияния на устройство высокочастотных помех из линии связи USB интерфейса, в состав устройства необходимо включить фильтр USB сигнала.
* Питание устройства обеспечивается интерфейсом USB. Для обеспечения надежной работы аппаратного шифратора, необходимо предусмотреть стабилизацию и, если необходимо, преобразование полученного от USB напряжения.
* Необходимо предусмотреть индикацию подачи питания на устройство и индикацию нормальной работы устройства.
* Для генерации сеансовых ключей шифрования в устройстве реализован аппаратно-программный генератор случайных чисел.
* В устройстве должна присутствовать энергонезависимая EEPROM память данных для хранения мастер ключей.

Структурная схема устройства для аппаратного шифрования информации, которая соответствует приведенным выше требованиям, изображена на рисунке 1.9.



Рис. 1.9 – Структурная схема устройства аппаратного шифрования

1. РАЗРАБОТКА СХЕМОТЕХНИЧЕСКОЙ РЕАЛИЗАЦИИ АППАРАТНОГО ШИФРАТОРА
   1. Выбор элементной базы для шифратора

Согласно техническому заданию, элементная база для аппаратного шифратора должна состоять из компонентов доступных в Украине.

* + 1. Выбор микроконтроллера
       1. Обоснование выбора

Согласно техническому заданию, устройство должно поддерживать USB. Следовательно, нужен контроллер с поддержкой этого интерфейса.

Устройство должно шифровать файлы с максимально возможной скоростью, поэтому контроллер должен быть быстрым.

Наиболее подходящими контроллерами являются Atmel AT91SAM7S64, Atmel AT89C5131, Philips LPC2141, Philips LPC2142.

Таблица 2.1 – Параметры микроконтроллеров

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Микроконтроллер | Быстродействие, MIPS | Объем flash, Кб | Объем ОЗУ, Кб | Цена, USD |
| AT91SAM7S64 | 50 | 64 | 16 | 5 |
| AT89C5131 | 4 | 32 | 1 | 8 |
| LPC2141 | 55 | 32 | 8 | 5 |
| LPC2142 | 55 | 64 | 16 | 7 |
| Коэф. важности | 0,2 | 0,15 | 0,15 | 0,5 |

Выберем один из них по матрице параметров:

1. ;
2. Составим матрицу приведенных параметров:

* если большее значение параметра  соответствует лучшему качеству ИМС, то ;
* если параметр не удовлетворяет этому условию, то .

;

1. Составим матрицу нормированных параметров A:

, где  – максимальное значение j-го параметра.

;

1. Вычислим оценочную функцию :

;

Т.к.  наименьшее, то AT91SAM7S64 будет оптимальным выбором.

* + - 1. Технические характеристики микроконтроллера AT91SAM7S64.

Характеристики микроконтроллера [2]:

* Содержит ядро процессора ARM7TDMI® ARM® Thumb® ;
* Высокопроизводительная 32-разр. RISC-архитектура;
* Обширный набор 16-разр. инструкций;
* Лидер по соотношению производительность/энергопотребление;
* Встроенное ядро внутрисхемной эмуляции с отладочным коммуникационным каналом;
* Внутренняя высокоскоростная флэш-память размером 64 кбайт и организацией 512 страниц по 128 байт в каждой
* Однотактный доступ при частотах до 30 МГц. Упреждающий буфер оптимизирует выполнение Thumb-инструкций при максимальном быстродействии;
* Время программирования страниц: 4 мс, в т.ч. автоматическое стирание страницы; время полного стирания: 10 мс;
* 10,000 циклов записи, 10-летний срок хранения данных, функции защиты секторов, бит защиты флэш-памяти;
* Интерфейс быстрого программирования флэш-памяти для серийного производства;
* 16 кбайт внутреннего высокоскоростного СОЗУ, однотактный доступ при максимальном быстродействии;
  + Контроллер памяти (MC)
* Встроенный контроллер флэш-памяти, определение некорректного доступа и формирование статуса ошибки;
  + Контроллер сброса (RSTC)
    - Состоит из схемы сброса при подаче питания и схемы детектора снижения напряжения питания с откалиброванным в заводских условиях порогом;
    - Выполняет обработку внешнего сигнала сброса и формирует информацию об источнике сброса;
      * Тактовый генератор (CKGR)
        + Маломощный RC-генератор, встроенный генератор частот от 3 до 20 МГц;
        + Одна схема ФАПЧ;

Контроллер управления энергопотреблением (PMC)

Возможность программной оптимизации энергопотребления, в т.ч. с использованием режимов пониженного быстродействия (Slow Clock), возможно снижение частоты до 500 Гц) и режим холостого хода (Idle);

Три программируемых внешних тактовых сигнала;

Усовершенствованный контроллер прерываний (AIC)

Индивидуальное маскирование, восемь уровней приоритетов, векторизованные источники прерываний;

Два внешних источника прерывания + один внешний источник прерывания с быстрым реагированием, защита от ложных прерываний;

Блок отладки (DBGU);

2-пров. УАПП + поддержка прерывания по отладочному коммуникационному каналу, программируемое предотвращение доступа со стороны внутрисхемного эмулятора;

Интервальный таймер (PIT);

20-разр. программируемый счетчик + 12 разр. счетчик интервалов;

Сторожевой таймер (WDT)

12-разр. программируемый счетчик с защитой ключом;

Выполняет сброс или генерирует запрос на прерывание системы;

Счетчик может быть остановлен, когда процессор находится в состоянии отладки или в режиме холостого хода;

Таймер реального времени (RTT)

* 32-разр. циклический счетчик с сигнализатором;
* Работает от внутреннего RC-генератора;
  + Один контроллер параллельного ввода/вывода (PIOA)
    - 42 программируемые линии ввода-вывода, мультиплексированные с двумя встроенными периферийными модулями;
    - Возможность генерации прерывания по изменению на входе любой линии ввода-вывода;
    - Индивидуально программируемые открытый сток, подтягивающий резистор и синхронизированный выход;
    - 11 канальный контроллер периферийных данных (PDC);
    - Один полноскоростной контроллер USB 2.0 (12 Мбит/сек), режим устройства;
    - Встроенный трансивер, встроенные конфигурируемые буферы FIFO емкостью 328 байт каждый;
    - Один синхронный последовательный контроллер (SSC);
    - Отдельные синхронизация и сигналы синхронизации кадра у каждого приемника и передатчика;
    - Поддержка аналогового интерфейса I2S, поддержка временного уплотнения;
    - Возможность высокоскоростной непрерывной передачи потока данных в 32-разр. формате;
      * Два универсальных синхронных/асинхронных приемопередатчика (УСАПП)
        + Раздельные генераторы скорости связи, инфракрасная модуляция/демодуляция (IrDA);
        + Поддержка смарт-карт ISO7816 T0/T1, аппаратное подтверждение связи, поддержка RS485;
        + Полный интерфейс модема на УСАПП1;

Последовательный периферийный интерфейс SPI с режимами ведущий/подчиненный

Программируемая длина данных от 8 до 16 бит, четыре внешних выхода выбора микросхем;

Один трехканальный 16-разр. таймер-счетчик (TC);

Три внешних тактовых входа, две линии универсального ввода-вывода на каждый канал;

Два ШИМ-генератора, режим захвата и генерации импульсов, возможность реверсирования счета;

Один четырехканальный 16-разр. ШИМ-контроллер (PWMC);

Один двухпроводной интерфейс (TWI);

Работает только в режиме ведущего, поддерживаются все двухпроводные ЭСППЗУ фирмы Atmel;

Один 8-канальный 10-разр. аналогово-цифровой преобразователь, четыре канала мультиплексированы с линиями цифрового ввода-вывода;

Граничное сканирование всех цифровых линий в соответствии со стандартом IEEE 1149.1 через интерфейс JTAG;

Линии ввода-вывода совместимы 5В уровнями и обладают повышенной нагрузочной способностью, до 16 мА каждая;

Источники питания

Встроенный стабилизатор напряжения 1,8 В с нагрузочной способностью до 100 мА для питания ядра и внешних компонентов;

Напряжение питания ввода-вывода VDDIO = 1,8В или 3,3В, отдельное питание флэш-памяти VDDFLASH = 3,3В;

Напряжение питания ядра VDDCORE = 1,8В (с детектором понижения напряжения);

Напряжение питании аналоговой схемы VDDANA = 3,3В;

* Статическая работа на частотах до 55 МГц при наихудших условиях работы: напряжение питания 1,65 В, температура 85°С.
  + 1. Выбор стабилизатора напряжения

Согласно техническому заданию размеры устройства не должны превышать 55х30х15 мм. Из этого следует, что стабилизатор нужно выбирать в миниатюрном корпусе. Немаловажный параметр стабилизатора – падение напряжения на нем. Чем оно меньше - тем лучше. Важным критерием является цена устройства.

На украинском рынке представлены такие стабилизаторы напряжения: LM1117, IRU1117, MC33269.

Выберем один из них по матрице параметров (Таблица 2.2).

Таблица 2.2 – Параметры стабилизаторов напряжения

|  |  |  |  |
| --- | --- | --- | --- |
| Стабилизатор | Падение напряжения, В | Макс. размер, мм | Цена, USD |
| LM1117 | 1 | 7 | 0.9 |
| IRU1117 | 1.2 | 7 | 0.6 |
| MC33269 | 1 | 9 | 0.9 |
| Коэф. важности | 0.4 | 0.4 | 0.2 |

1. ;
2. Составим матрицу приведенных параметров:

* если большее значение параметра  соответствует лучшему качеству ИМС, то ;
* если параметр не удовлетворяет этому условию, то .

;

1. Составим матрицу нормированных параметров A:

,

где  – максимальное значение j-го параметра.

;

1. Вычислим оценочную функцию :

;

Т.к.  наименьшее, то LM1117 будет оптимальным выбором.

* 1. Описание работы схемы

Питание +5 В и сигнал интерфейса USB поступает с разъема XS1. Резистор R1 подтягивает линию USB D+ на +3,3 В (для автоматического определения скорости устройства хостом). Дроссели L1-L5 используются для подавления высокочастотных помех. Конденсаторы С3, С5, С6 и резисторы R2, R3 представляют собой стандартную схему фильтрации сигналов USB. После фильтров сигнал USB поступает в микроконтроллер AT91SAM7S64 DD1.

Схема питания, генерирующая +3,3 В, собрана на линейном стабилизаторе напряжения LM1117-3.3 DA1, диоде Шотки VD1, предотвращающем обратные токи, и блокировочных конденсаторах С1, С2, С4.

Конденсаторы С7, С9 и резистор R4 это цепочка, необходимая для работы ФАПЧ (PLL) микроконтроллера.

Конденсаторы C12, C13 совместно с кварцевым резонатором ZQ1 представляют собой колебательный контур, задающий частоту генератора, встроенного в микроконтроллер.

C8, C10, C11, C14, С15, С16, С17, С18, С19, С20, С21, С22 – это блокировочные конденсаторы по питанию микроконтроллера.

При включении устройства в порт USB, микроконтроллер инициализирует внутренние регистры, настраивает ФАПЧ и проводит процесс энумерации USB устройства. Далее проходит процедура инициализации алгоритма Blowfish и программа микроконтроллера входит в цикл ожидания команд от хоста.

## 2.3 Моделирование в пакете схемотехнического проектирования OrCAD

Для моделирования в системе OrCad была выбрана схема стабилизации напряжения питания +3,3 В. Схема питания состоит из: стабилизатора питания LM1117, диода Шотки MBRS130T3 и блокировочных конденсаторов. На рис. 2.8 представлена схема для моделирования в редакторе Schematics.

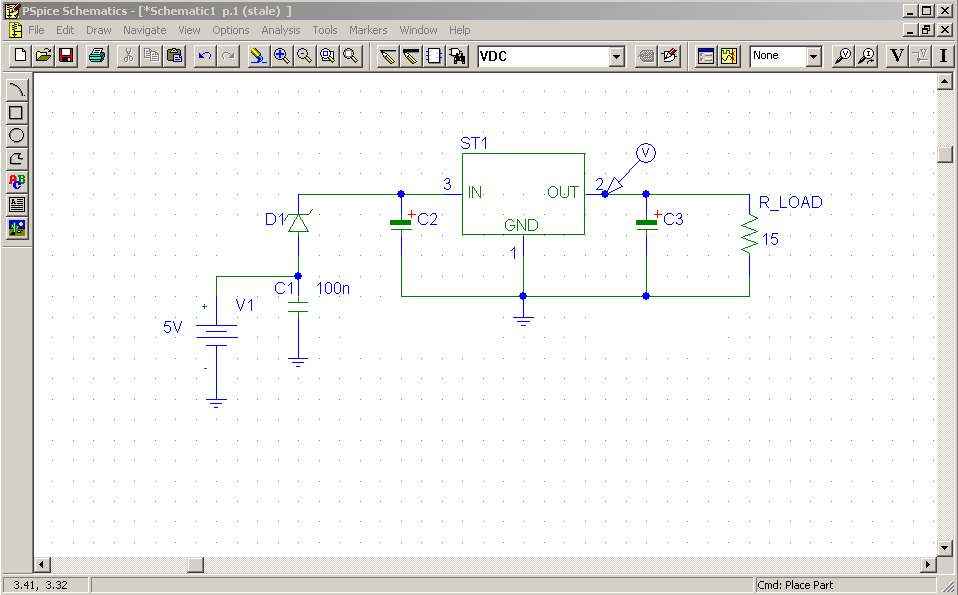


Рис. 2.8 – Схема для моделирования DC Sweep

Для моделирования были использованы специальные библиотеки моделей от производителей, чтобы получить максимально достоверные результаты. Так как на вход этой схемы напряжение подается от USB, то промоделируем ее по напряжению питания от 0 до 6 В с шагом 0,1 В. Для этого в меню Analysis/Setup выбираем вид анализа DC Sweep и проводим анализ схемы по Voltage Source от 0 В до 6 В с шагом 0,1 В.

В результате как показано на рис. 2.9, данная схема стабилизирует напряжение 3,3 В начиная с 4,5 В источника, т.е. подходит для нашего устройства.

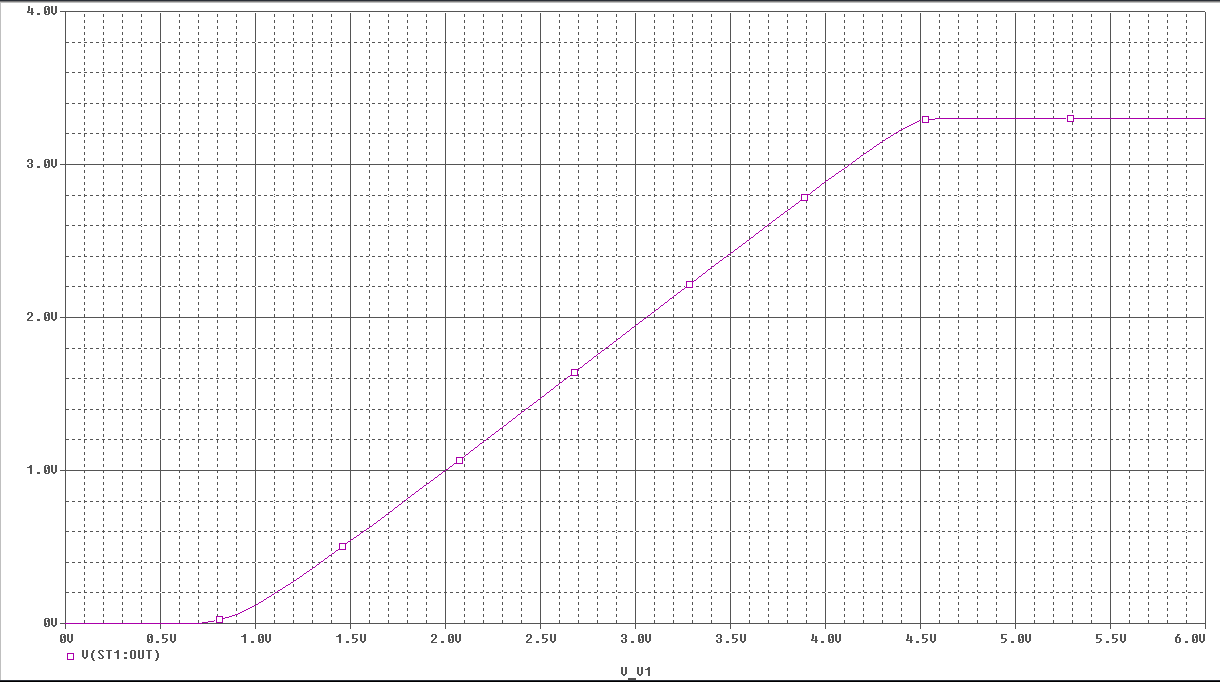


Рис. 2.9 – DC Sweep анализ

Чтобы проверить как стабилизатор справляется с низкочастотной помехой, промоделируем схему, подключив на вход источник синусоидального напряжения 50 Гц со смещением в 5 В и амплитудой колебаний 0,1 В. Для этого вместо источника постоянного напряжения VDC установим источник типа VSIN, как показано на рис. 2.10.

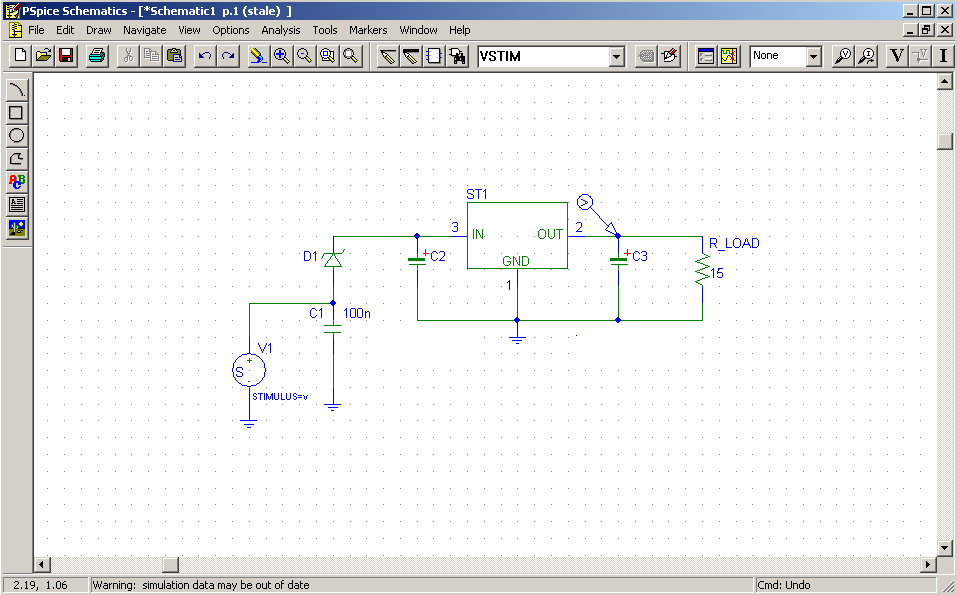


Рис. 2.10 – Схема питания с источником синусоидального сигнала

На рис. 2.11 показан сигнал, подаваемый на вход, а на рис. 2.12 – сигнал получаемый на выходе стабилизатора.

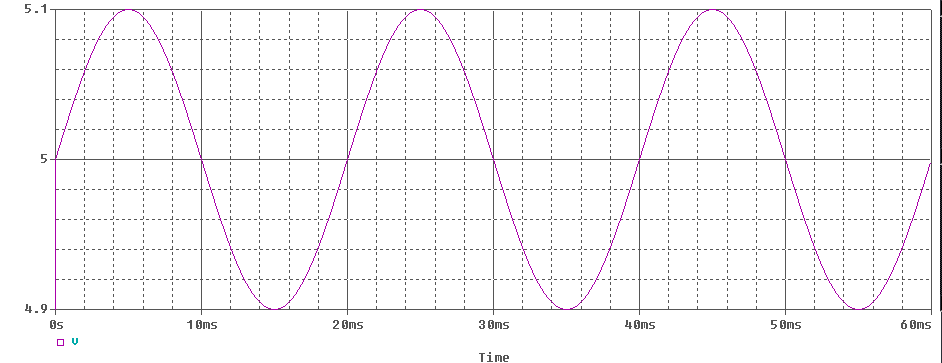


Рис. 2.11 – Входной синусоидальный сигнал

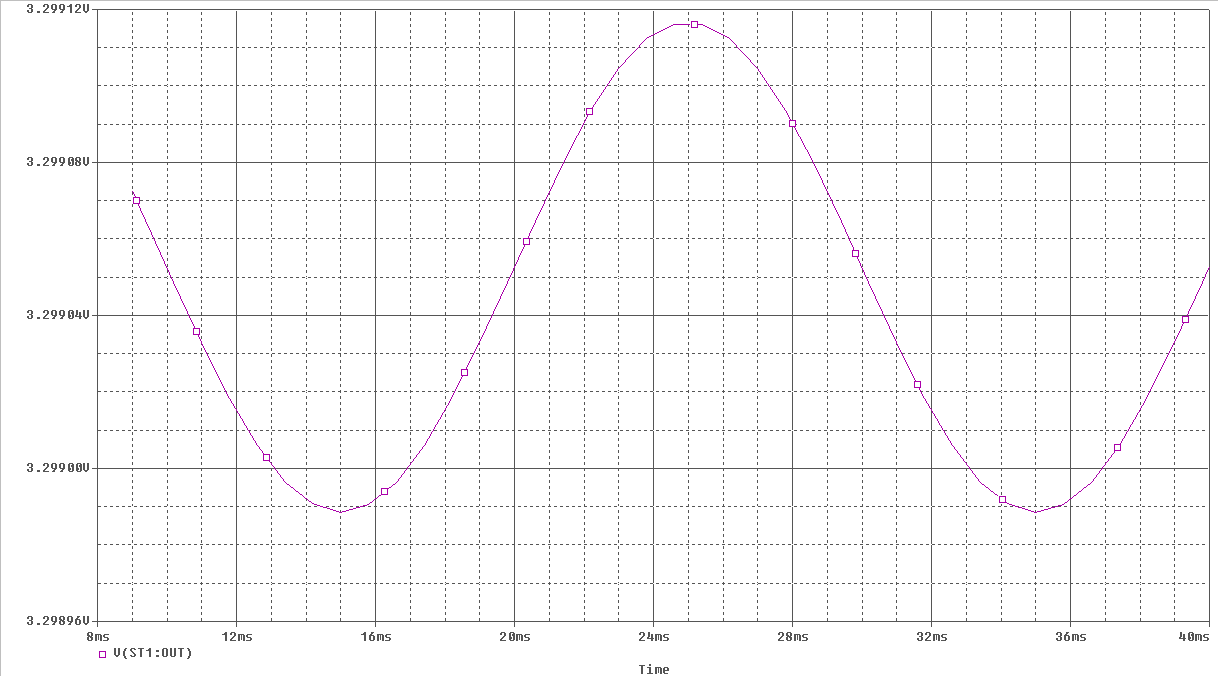


Рис. 2.12 – Выходной сигнал стабилизатора

Как видно из рис. 2.12 амплитуда колебаний составляет 0,00006 В, при входной амплитуде 0,1 В.

Т.е. коэффициент подавления будет:

 (Дб);

1. РАЗРАБОТКА ПЕЧАТНОГО УЗЛА АППАРАТНОГО ШИФРАТОРА
   1. Выбор и обоснование типа печатной платы

Существует 4 типа печатных плат (ПП) [4]:

* односторонние (ОПП);
* двухсторонние (ДПП);
* многослойные (МПП);
* гибкие (ГПП).

ОПП просты в конструировании и экономичны в изготовлении. Они характеризуются: возможностью обеспечить повышенные требования к точности выполнения проводящего рисунка; установки навесных элементов на поверхность платы со стороны, противоположной пайке, без дополнительной изоляции; возможностью использования перемычек из проводникового материала. Монтажная и трассировочная возможности этих плат низкие. Надежность и механическая прочность крепления элементов низкие. Для повышения прочности крепления элементов возможно изготовление ОПП с металлизацией отверстий. Обычно, ОПП применяют для монтажа бытовой ЭВА, в силовой электронике, в НЧ устройствах.

ДПП имеют высокую плотность монтажа и хорошую механическую прочность крепления.

МПП – это совокупность слоев диэлектрика и слоев печатного монтажа, имеющие межслойные соединения или открытый доступ к внутренним слоям. По сравнению с ОПП и ДПП они характеризируются повышенной плотностью монтажа, устойчивостью к климатическим и механическим воздействиям, уменьшением размеров и числа внешних выводов.

Преимущества ГПП:

– толщина d=0,1…0,28 мм → значительное уменьшение веса;

– высокая ударопрочность (удары практически не влияют).

Для своего устройства я решил использовать ДПП, т.к.:

* Применение двухсторонней печатной платы позволяет облегчить трассировку соединений, компактнее разместить элементы, рационально использовать площадь печатной платы, следовательно, уменьшить ее размер и расход материала.
* ДПП с металлизированными монтажными и переходными отверстиями характеризуются: высокими коммутационными свойствами; повышенной прочностью соединения вывода навесного элемента с проводящим рисунком платы; повышенной плотностью монтажа.
  1. Выбор и обоснование материала печатной платы

Основными материалами для изготовления двухсторонних печатных плат являются гетинакс и стеклотекстолит. Гетинакс – спрессованные слои электроизоляционной бумаги, пропитанные фенольной смолой. Стеклотекстолиты – прессование слои стеклотканей пропитанные эпоксидной смолой. Оба материала имеют малую водопроницаемость (0,2-0,8 % при Т = 260 ºС), большое поверхностное сопротивление (104 МОм), термостойкость материала в течении 1000 часов. Но по остальным электромеханическим параметрам стеклотекстолит превосходит гетинакс. Поэтому я выбрал стеклотекстолит фольгированный марки СФ-2-35-1,5 ГОСТ 10316-78. Как видно из обозначения, толщина фольги – 35 мкм, толщина основания – 1,5 мм.

* 1. Выбор и обоснование класса точности печатной платы

По точности изготовления элементов печатного монтажа ПП делят на 5 классов.

Таблица 3.1 – Классы точности печатных плат [4]

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Класс точности | Плотность монтажа | Мин. ширина проводника | Расстояние между краями соседних эл-тов | Разрешающая способность | Предельные размеры |
| 1 | малая | 0,75 | 0,75 | 0,6 | без огр. |
| 2 | средняя | 0,45 | 0,45 | 1,2 | 240 |
| 3 | средняя | 0,25 | 0,25 | 2 | 170-240 |
| 4 | высокая | 0,15 | 0,15 | 3,3 | 170 |
| 5 | высокая | 0,1 | 0,1 | 5 | 100-170 |

Для своего устройства я решил использовать четвертый класс точности, потому что:

* Это самый низкий класс, который позволяет провести печатный проводник к выводам ИМС (шаг 0,5мм);
* Он имеет достаточно высокую плотность монтажа, что позволяет уменьшить габариты.

Таблица 3.2 – Параметры 4-го класса точности печатной платы [4]

|  |  |
| --- | --- |
| Параметр | Значение |
| Минимальная ширина печатного проводника, , мм | 0,15 |
| Минимальное расстояние между проводниками, s, мм | 0,15 |
| Допуск на диаметр отверстий с металлизацией, ∅<1, мм | +0,05; -0,1 |
| , ∅>1, мм | +0,1; -0,15 |
| Допуск на размещение отверстий, , мм | 0,05 |
| Допуск на размещение КП, , мм | 0,05 |
| Допуск на размещение проводников, , мм | 0,03 |
| Минимальная ширина пояска КП, , мм | 0,025 |

* 1. Конструкторско-технологический расчет печатного узла шифратора
     1. Минимальная ширина печатного проводника по постоянному току.

В схеме сканера присутствуют ШП (+3,3 В) и ШЗ. Следовательно, минимальную ширину проводника по постоянному току можно рассчитать только для ШЗ, т. к. в ней будет протекать максимальный ток.

, (3.1)

где– максимальный постоянный ток, протекающий в проводниках;

– допустимая плотность тока ;

– толщина проводника

, (3.2)

где– толщина фольги;

– толщина гальванически осажденной меди (мм);

– толщина химически осажденной меди (мм);

(мм);

Таблица 3.3 – Потребление тока

|  |  |  |
| --- | --- | --- |
| Радиоэлемент | Iпотр, мА | Колич. |
| AT91SAM7S64 | 60 | 1 |
| DMHG-GJS-NP-LM-1 | 20 | 1 |

(А);

Подставим значения в формулу 3.1:

(мм);

* + 1. Минимальная ширина проводника с учетом допустимого падения напряжения.

, (3.3)

где– объемное удельное сопротивление фольги для данного метода изготовления проводника, ;

– самый длинный проводник ();

;

;

– допустимое падение напряжения ;

(мм);

* + 1. Минимальный диаметр монтажных отверстий.

, (3.4)

где– диаметр вывода элемента (для USB разъема );

– нижнее предельное отклонение от номинального диаметра монтажного отверстия .

 – разница между минимальным диаметром отверстия и максимальным диаметром вывода, .

Таким образом, для ИМС (мм).

* + 1. Диаметр контактных площадок.

Минимальный диаметр КП:

, (3.5)

где - толщина фольги;

 – минимальный эффективный диаметр КП:

, (3.6)

где– ширина пояска КП ();

– погрешность расположения центра отверстия ();

– погрешность расположения центра КП ;

– максимальный диаметр просверленного отверстия:

, (3.7)

где – номинальный диаметр отверстия;

 – допуск по диаметру отверстия ();

(мм);

Из формулы 3.6:

(мм);

Подставим значения в формулу 3.5:

(мм);

(мм);

Выбираем диаметр КП равный 1,5 мм.

* + 1. Ширина проводников.

Минимальная ширина проводников:

, (3.8)

где  для печатных плат 4-го класса точности.

(мм);

Максимальная ширина проводников:

(мм);

Выбираем ширину проводников равную 0,25 мм.

* + 1. Минимальное расстояние между проводником и КП

, (3.9)

где– расстояние между центрами элементов ();

– погрешность смещения проводника ().

(мм);

* + 1. Минимальное расстояние между двумя соседними проводниками

(мм);

* + 1. Минимальное расстояние между двумя соседними КП. [5]

(мм);

* + 1. Расстояние между проводником и КП.

(мм);

Рассчитанное значение больше , поэтому проводник, проложенный между КП, не будет касаться ни одной из КП.

* + 1. Соответствие 4-му классу точности.

Проведя КТР, мы убедились, что все элементы печатной платы соответствуют выбранному классу точности.

* 1. Электрический расчет печатной платы
     1. Падение напряжения на печатных проводниках

, (3.10)

где– объемное удельное сопротивление фольги для данного метода изготовления проводника ;

– максимальный постоянный ток, протекающий в проводниках ;

– самый длинный проводник ;

 – ширина проводника ;

– толщина проводника ,

(В);

Рассчитанное падение напряжения не превышает 5% от EП.

* + 1. Мощность потерь.

, (3.11)

где т. к. расчет идет по постоянному току;

– напряжение питания ;

– тангенс диэлектрических потерь материала ПП ;

– собственная емкость платы: ,

где– диэлектрическая проницаемость материала ПП ;

– площадь металлизации ;

– толщина ПП ;

(пФ);

Из формулы 3.4:

(мкВт);

* + 1. Паразитная емкость между двумя соседними проводниками.

, (3.12)

где– длинна взаимного перекрытия двух параллельных проводников ;

– расстояние между проводниками ;

(пФ);

* + 1. Паразитная индуктивность ШП и ШЗ.

, (3.13)

где– суммарная длинна ШП и ШЗ ;

(мкГн);

* 1. Расчет теплового режима

Максимальную мощность рассеивает стабилизатор питания МС33269-D. На данной микросхеме падает напряжение 1 В и протекает суммарный ток потребления всей схемы – 80 мА. Т. е. выделяемая мощность равна:

(Вт);

Из документации:

* допустимая температура кристалла микросхемы : ;
* сопротивление кристалл/корпус ;
* сопротивление корпус/среда .

Для расчета возьмем температуру окружающей среды .

Рассчитаем температуру кристалла [6]:

 (°C);

Данная температура является допустимой для работы стабилизатора. Следовательно, не требуется использование радиатора.

Согласно документации производителя, при  стабилизатор способен рассеять до 600 мВт, что соответствует расчетам.

* 1. Расчет вибропрочности печатной платы

Данные для расчета:

* материал печатной платы – СФ-2-35-1,5.
* габаритные размеры платы – 51 х 26 х 1,5 мм.
* масса элементов на плате – 15 г.
* коэффициент перегрузки – 5.
* частота вибрации 60 Гц.
* параметры стеклотекстолита:

предел текучести – ;

модуль Юнга – ;

коэффициент Пуассона – ;

коэффициент затухания – ;

удельный вес – ;

удельная плотность – ;

коэффициент запаса прочности – .

* тип закрепления: опирание по четырем сторонам.

Рассчитаем собственную частоту колебаний печатной платы [5]:

1. Масса печатной платы:

(г);

1. Коэффициент влияния:

;

1. Коэффициент :

;

1. Цилиндрическая жесткость печатной платы:

(Н∙м);

1. Собственная частота колебаний печатной платы:

(Гц);

Так как собственная частота намного больше 250 Гц, то плата обладает хорошей виброустойчивостью и дальнейшие расчеты можно не проводить.

* 1. Расчет показателей надежности

Надежность – это свойство объекта сохранять во времени в установленных пределах значения всех параметров, характеризующих способность выполнять требуемые функции в заданных режимах и условиях применения, технического обслуживания, ремонтов, хранения и транспортирования (ГОСТ 27.002-83).

Расчет надежности заключается в определении показателей надежности изделия по известным характеристикам надежности составляющих компонентов и условиям эксплуатации. Данные для расчета надежности сведены в Таблице 3.4. Формулы для расчета взяты из [4].

Таблица 3.4 – Параметры надежности элементов [4]

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Наименование элемента | Тип элемента | N |  |  |  |  |  |
| ИМС | AT91SAM7S64 | 1 | 0,2 | 1 | 2 | 10 | 4 |
| LM1117 | 1 | 0,2 | 1 | 2 | 10 | 4 |
| Резистор | RС0805 | 6 | 0,02 | 0,06 | 0,6 | 10 | 0,0432 |
| Конденсатор керамический | СС0805 | 20 | 0,3 | 0,1 | 0,5 | 10 | 3 |
| Конденсатор танталовый | Size B | 2 | 0,5 | 0,5 | 0,5 | 10 | 2,5 |
| Резонатор |  | 1 | 0,25 | 1 | 1 | 10 | 2,5 |
| ПП | ДПП | 2 | 1 | 1 | 1 | 10 | 20 |
| Диод | MBRS130T3 | 1 | 0,2 | 0,2 | 1 | 10 | 0,4 |
| Дроссель | BLM21PG221SN | 5 | 0,3 | 1 | 1 | 10 | 15 |
| Контакты  разъема | USB-PWBK-4A | 5 | 0,2 | 1 | 1 | 10 | 10 |
| Контакты  разъема | Джампер | 2 | 0,2 | 1 | 1 | 10 | 4 |
| Пайка выводов | Печатный монтаж | 148 | 0,005 | 1 | 1 | 10 | 7,4 |

N – количество элементов.

– интенсивность отказов элемента при нормальных условиях работы.

– коэффициент нагрузки:

для резисторов

; (3.14)

для конденсаторов

; (3.15)

– поправочный коэффициент по температуре.

 – поправочный коэффициент на влияние внешних воздействий (для наземной стационарной аппаратуры ).

Результирующая интенсивность отказов равно сумме интенсивностей отказов элементов:

(ч-1);

Определим среднее время наработки на отказ:

(ч);

Рассчитаем вероятность безотказной работы:

; (3.16)

Вероятность безотказной работы за 1 год:

.

Вероятность отказа за 1 год:

.

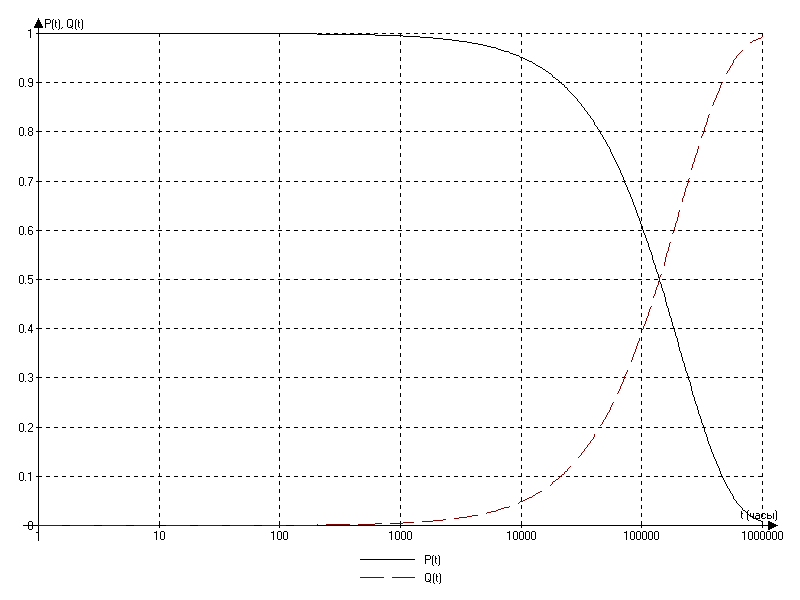


Рис. 3.1 – Графики вероятности безотказной работы P(t) и вероятности отказа Q(t)

* 1. Технология поверхностного монтажа

Особенностью современного производства электронных устройств является все более широкое применение больших и сверхбольших интегральных схем (БИС и СБИС). При этом существенно возрастает количество выводов каждой схемы, расстояния между выводами уменьшаются с 2,5 мм до 0,625 мм и менее.

Установка многовыводных корпусов БИС И СБИС на печатные платы технически и экономически более эффективна не в сквозные отверстия, а на контактные площадки, расположенные на поверхности печатных плат.

Этим объясняется все боле широкий переход от монтажа компонентов в отверстия (PTH - Plated Through Hole) к технологии поверхностного монтажа (SMT - Surface Mount Technology).

Вместе с тем, в большинстве серийных электронных блоков применяют как поверхностный монтаж, так и монтаж в отверстия. Это связано с тем, что конструкции ряда компонентов не пригодны для поверхностного монтажа. В устройствах, работающих в условиях ударных и вибрационных перегрузок, предпочитают монтаж в отверстия из-за более надежного крепления компонентов.

Навесные компоненты для поверхностного монтажа, намного меньше, чем их традиционные эквиваленты, которые монтируются в отверстия. Вместо длинных выводов, как у корпусов, монтируемых в отверстия, они имеют очень короткие выводы или просто внешние контактные площадки. Та­кие компоненты закрепляются на верхней (или нижней) стороне коммутационной платы при совмещении их вы­водов или внешних контактов с контактными площадками.

### Преимущества SMT:

* меньшие размеры компонентов приводят к уменьшению размеров плат. Это уменьшает себестоимость. Типичное SMT преобразование уменьшает пространство на плате до 30 % размера за счет отсутствия отверстий.
* большее количество функциональных возможностей компоновки SMT элементов.
* компоненты могут легко размещаться с обеих сторон платы, что увеличивает плотность размещения.
* меньшая масса изделия и более низкий профиль изделия могут улучшать вибро и ударопрочностные свойства.
* Некоторые более новые компоненты доступны только в SMT корпусах.

### Недостатки SMT:

* платы с SMT компонентами требуют специальной разработки и автоматизированного проектирования;
* у печатных плат SMT высокие требования к допускам и качеству изготовления;
* применение SMT компонентов для изготовления печатных плат является экономически оправданным при наличии оборудования автоматизации сборки;
* Некоторые разработки требуют применения DIP компонентов. Для сборки таких плат приходиться применять автоматическую установку SMT компонентов, что увеличивает издержки на выполнение дополнительных сборочных шагов. В таких случаях, есть такие платы, реализация которых на DIP компонентах имела бы меньшую стоимость сборочной операции.
* При применении SMT появляются дополнительные издержки на программирование процесса автоматизации сборки и изготовление трафаретов.
  + 1. Типы SMT сборок

В электронной промышленности существует шесть общих типов SMT сборки, каждому из которых соответствует свой порядок производства. Когда разработчик выбирает тип сборки, его целью должна быть минимизация числа операций, так как каждая операция увеличивает промышленную стоимость. Существует специальный стандарт (National Technology Roadmap for Electronic), в котором представлены основные виды сборок, разбитые по классам.

Существуют следующие схемы поверхностного монтажа:

* Тип 1 - монтируемые компоненты установлены только на верхнюю сторону;
* Тип 2 - монтируемые компоненты установлены на обе стороны платы;
* Класс А - только through-hole (монтируемые в отверстия) компоненты;
* Класс В - только поверхностно монтируемые компоненты (SMD);
* Класс С - смешанная: монтируемые в отверстия и поверхностно монтируемы компоненты;
* Класс Х - комплексно-смешанная сборка: through-hole, SMD, fine pitch, BGA;
* Класс Y - комплексно-смешанная сборка: through-hole, surface mount, Ultra fine pitch, CSP
* Класс Z - комплексно-смешанная сборка: through-hole, Ultra fine pitch, COB, Flip Chip, TCP;

Варианты схем поверхностного монтажа:

1. SMT - Только верхная сторона



Рис. 3.2 – Установка SMT элементов на одну сторону платы

Этот тип не является общим так как большинство разработок требует некоторых DIP компонентов. Его называют IPC Type 1B.

Порядок проведения процесса:

* нанесение припойной пасты, установка компонентов, пайка, промывка.

### 2. SMT Верхние и нижние стороны



Рис. 3.3 – Установка SMT элементов на обе стороны платы

На нижней стороне платы размещаются чип-резисторы и другие компоненты небольших размеров. При использовании пайки волной, они будут повторно оплавляться за счет верхнего (побочного) потока волны припоя. При размещение больших компонентов с обеих сторон, типа PLCC, увеличивают издержки производства, потому что компоненты нижней стороны должны устанавливаться на специальный токопроводящий клей. Данный тип называется IPC Type 2B.

Порядок проведения процесса:

* нанесение припойной пасты, установка компонентов, пайка, промывка нижней стороны;
* нанесение припойной пасты на верхнюю сторону печатной платы, установка компонентов, повторная пайка, промывка верхней стороны.

3. SMT верхняя сторона в первом случае и верхняя и нижняя во втором, но PTH только верхняя сторона.

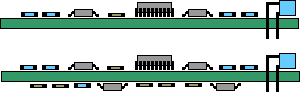


Рис. 3.4 – Установка SMT элементов на обе стороны платы и PTH элементов на одну сторону платы

Этот метод установки используется, когда имеются DIP компоненты, в SMT сборке. Процесс включает размещение DIP компонентов, вставляемых в отверстия перед SMT пайкой. При использовании данного метода убирается лишняя операция пайки волной или ручной пайки PTH компонентов, что значительно уменьшает стоимость изделия. Первое требование - способность компонентов противостоять вторичной пайке. Размеры отверстия платы, контактные площадки и геометрия трафарета должны быть точно совмещены, чтобы достичь качественной пайки. Плата должна иметь сквозные металлизированные отверстия и может быть односторонней или двухсторонний, то есть компоненты могут размещаться как с верхней так и с нижней стороны.

Порядок обработки односторонней печатной платы:

* нанесение припойной пасты, установка SMT компонентов, установка PTH компонентов, пайка, промывка верхней стороны.

Порядок обработки двухсторонней печатной платы:

* нанесение припойной пасты, установка SMT компонентов, SMT пайка, промывка нижней стороны;
* установка PTH компонентов, пайка, промывка верхней стороны.

### 4. Тип 1С: SMT только верхняя сторона и PTH только верхняя сторона



Рис. 3.5 – Установка SMT и PTH элементов на верхнюю сторону платы

Данный метод является смешанной технологией сборки. Все модули SMT и PTH установлены на верхней стороне платы. Допускается установка некоторых компонентов монтируемых в отверстия (PTH) на верхней стороне платы, где размещены SMT компоненты для увеличения плотности. Данный тип сборки называется IPC Type 1C.

Порядок проведения процесса:

* нанесение припойной пасты, установка, оплавление, промывка верхней части SMT;
* автоматическая установка DIP, затем осевых компонентов (такие как светодиоды);
* ручная установка других компонентов ;
* пайка волной PTH компонентов, промывка.

### 5. Тип 2С: SMT верхняя и нижняя стороны или PTH на верхней и нижней стороне



Рис. 3.6 – Установка SMT и PTH элементов на обе стороны платы

Установка поверхностно монтируемых и монтируемых в отверстия (DIP) компонентов с обеих сторон платы не рекомендуется из-за высокой стоимости сборки. Эта разработка может требовать большого объема ручной пайки. Также не применяется автоматическая установка PTH компонентов из-за возможных конфликтов с SMT компонентами на нижней стороне платы. Данный тип сборки называется IPC Type 2C.

Порядок проведения процесса:

* нанесение припойной пасты, установка, пайка, промывка верхней стороны SMT;
* нанесение специального токопроводящего клея через трафарет, установка, фиксация SMT;
* автоматическая установка DIP и осевых компоненты;
* маскирование всей нижней стороны PTH компонентов;
* ручная установка других компонентов;
* пайка волной PTH и SMT компонентов, промывка;
* ручная пайка нижней стороны PTH компонентов.
  + 1. Установка компонентов на плату

Традиционные компоненты, монтируемые в отверстия, являются наиболее узким местом в процессе установки их на печатную плату. Это практически полностью исключает возможность автоматизации процесса. Гораздо проще и быстрее автоматизировать процесс установки поверхностно монтируемых компонентов.

Машины для автоматической установки работают по трем основным принципам:

* поочередная установка компонентов;
* поочередно-одновременная установка компонентов;
* одновременная установка компонентов;

В аппаратах поочередной установки один компонент все время устанавливается одной или двумя установочными головками. Поочередная установка, также может проводиться при помощи револьверной головки. В поочередно-одновременной установке несколько компонентов может быть установлено одновременно. Установочные машины одновременного типа, устанавливают все или возможно-большее количество компонентов за один раз.

Поочередные и поочередно-одновременные машины, также называются последовательными и их основное преимущество в гибкости настройки. Если машина поочередной установки оснащена револьверной головкой, скорость установки компонентов на печатную плату значительно возрастает. Эти машины могут устанавливать компоненты нескольких типов. Место установки компонента может быть легко изменено, а точность установки достаточно высока.

Машины одновременной установки компонентов значительно производительней. Скорость установки компонентов может достигать 300000 компонентов в час, однако эти машины не так просты и гибки в настройке. Если для изменения места установки компонента в машинах поочередного и поочередно-одновременного типа достаточно изменить программы, то для машины одновременной установки требуются значительные более сложные механические изменения. Поэтому, эти машины используются для особо больших партий изделий.

* + 1. Нанесение припойной пасты

Для крепления компонентов на печатную плату используется метод нанесения припойной пасты. Это наиболее широко используемый метод установки компонентов. В этом методе, припойная паста наносится непосредственно на контактные площадки печатной платы.

Припойные пасты представляют собой смесь мелкодисперсного порошка материала припоя со связующей жидкой основой. При этом содержание порошка припоя составляет приблизительно 88 % от веса всей смеси (обычно этот показатель меняется в пределах от 85 до 92 %). Однако чаще всего состав припойных паст выражают через соотношение ингредиентов материала припоя. Так, например, 63/37 означает содержание в составе материала припоя 63 % олова и 37 % свинца, а 62/36/2 – 62 % олова, 36% свинца и 2 % серебра. Оба этих состава используются для приготовления припойных паст в SMT.

Флюс в составе припойных паст служит для активации контактируемых металлических поверхностей, удаления с них окислов и предотвращения окисления припоя в процессе пайки (что необходимо для создания паяного соединения). Флюс обеспечивает требуемую растекаемость (реологию), а также изменение вязкости со временем (тиксотропность) при нанесении припойной пасты на коммутационную плату. Если состав припойной пасты имеет недостаточную вязкость, она будет растекаться, или "расползаться", что, несомненно, приведет к потере точности рисунка, обеспечиваемой трафаретом, а это в свою очередь может послужить причиной образования шариков припоя или перемычек в процессе пайки. Кроме того, количество припойной пасты, нанесенной на плату, в ряде мест может оказаться недостаточным из-за ее растекания по плате.

Промежуток времени между нанесением припойной пасты на коммутационную плату и процессом пайки является еще одним фактором, который нужно учитывать при выборе пасты. Длительный промежуток времени может привести к ухудшению электрофизических параметров пасты. Припойная паста не должна ухудшать свои параметры не только в условиях термообработки при повышенной температуре, но и в условиях циклического воздействия температуры, которым подвергается плата как в процессе пайки, так и на других этапах изготовления изделия.

Наиболее важным в массовом производстве печатных плат, является метод трафаретного нанесения припойной пасты (рис.3.7.), в котором паста продавливается через трафарет (окна) на контактные площадки печатной платы. Припойная паста уже содержит в себе и припой, и флюс, а их пропорция одна из важных характеристик пасты. Материалом трафарета может быть как сплав никеля, так и нержавеющая сталь. Отверстия в трафарете обычно прорезаются лазером или протравливаются. В массовом производстве этот метод эффективен, но не гибок, так как свой собственный трафарет (причем несколько) требуется для каждой платы. Гибкость достигается только за счет быстрой смены трафарета и автоматического распределения пасты.

При проведении скребком по поверхности трафарета припойная паста продавливается сквозь отверстия в трафарете на контактные площадки. Наиболее важной фазой этого процесса является продвижение пасты вдоль поверхности трафарета, она должна продвигаться с правильной силой, углом и скоростью. Трафарет и скребок должны быть чистыми и паста должна иметь строго определенные характеристики для этой силы, угла и скорости. Ошибки в этих параметрах приводят к плохим характеристикам пайки. Больше половины ошибок всего процесса сборки печатных плат приходятся именно на процесс нанесения припойной пасты. Преимуществом метода трафаретного нанесения припойной пасты является то, что паста может быть нанесена слоем до 300 мкм с очень высокой точностью.

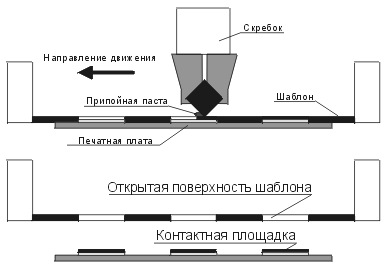


Рис. 3.7 – Нанесение припойной пасты

* + 1. Пайка SMT компонентов

Традиционная техника пайки волной припоя выполняется чаще всего погружением компонента в ванну с припоем. Для пайки на коммутационных платах SMT компонентов обычно применяется метод расплавления дозированного припоя. Пайка расплавлением припоя в парогазовой фазе в настоящее время уступает место пайке с инфракрасным нагревом, лазерная пайка пока не получила распространения. Ведущие поставщики сборочно-монтажного оборудования обычно включают установки для пайки в состав выпускаемых производственных линий.

Появление на коммутационных платах поверхностно монтируемых компонентов существенно изменило технологию пайки. Пайка волной припоя была внедрена в 50-х гг. и до настоящего времени является единственным групповым методом пайки компонентов, устанавливаемых в отверстия коммутационных плат. Для пайки плат со смешанным монтажом (компоненты, монтируемые в отверстия с одной стороны платы и простые, монтируемые на поверхность) был разработан метод пайки двойной волной припоя. Технология пайки поверхностно монтируемых компонентов расплавлением дозированного припоя в парогазовой фазе (ПГФ) появилась в 1973 г., когда фирма DuPont разработала и запатентовала специальные жидкие материалы. В течение нескольких лет Western Electric была единственной фирмой, пользовавшейся преимуществами этой новой разработки. С 1983 г. основным конкурентом пайки в ПГФ стала пайка расплавлением дозированного припоя с помощью инфракрасного нагрева (ИК-пайка).

Для чувствительных к тепловому воздействию и сложных микросборок с поверхностным монтажом тремя ведущими японскими компаниями была разработана и реализована лазерная пайка.

1. РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ АППАРАТНОГО ШИФРАТОРА

Исходный код программного обеспечения приведен в Приложении А.

1. Программное обеспечение микроконтроллера

Основой устройства является высокопроизводительный (50 MIPS) 32-х разрядный ARM микроконтроллер AT91SAM7S64. Программное обеспечение для этого микроконтроллера было написано на языке “C” в программном продукте IAR Embedded Workbench 4.40A. Отладка программы производилась в устройстве через интерфейс JTAG.

После подачи питания на микроконтроллер, программа выполняет инициализацию регистров микроконтроллера и периферийных модулей (схему ФАПЧ, контроллер потребляемой мощности, контроллер прерываний, генератор случайных чисел). Если установка параметров микроконтроллера прошла успешно, программа зажигает красный светодиод – индикация работы устройства.

На следующем этапе работы программа микроконтроллера проходит процедуру энумерации USB устройства (перевод устройства в адресное состояние, выбор конфигурации, выбор интерфейса). Устройство имеет одну конфигурацию и один интерфейс. Если энумерация прошла успешно, программа зажигает зеленый светодиод – индикация распознанного системой USB устройства.

Далее программа переходит в цикл ожидания управляющих команд от драйвера устройства.

Команды, поддерживаемые устройством:

* получение версии устройства;
* загрузка в устройство мастер ключа;
* считывание номера мастер ключа;
* запись номера мастер ключа, с которым будут производится операции шифрования;
* назначение символического имени ключу с указанным номером;
* шифрование файла;
* расшифровка файла;

При получении запроса на шифрование файла, программа ожидает запроса с номером мастер ключа (К1), который хранится в памяти микроконтроллера и будет использоваться для шифрования сеансового ключа.

После получения номера мастер ключа программа с помощью ГСЧ генерирует сеансовый ключ (К2). Далее происходит инициализация алгоритма Blowfish для ключа К2 и шифрование выбранного файла (шифруются получаемые устройством пакеты, размером 64 байта). После того как файл зашифрован ключ К2 шифруется алгоритмом Blowfish с ключом К1 и зашифрованный ключ К2 передается в персональный компьютер.

Для расшифровки файла, программа микроконтроллера считывает зашифрованный ключ К2 и номер ключа К1. Программа расшифровывает ключ К2 и далее расшифровывает файл используя ключ К2.

На рисунке 4.1. представлена блок-схема программы для микроконтроллера.

1. Структура программы микроконтроллера



Рис. 4.1 – Структура программы микроконтроллера

1. Драйвер устройства

Драйвер устройства написан на языке “C++” в программном продукте Compuware Driver Studio 3.2.0. с использованием Microsoft Windows XP DDK.

Драйвер устройства принимает IOCTL запросы на ввод-вывод (Input Output Control Codes – IOCTL) от DLL библиотеки, которая поставляется в комплекте с драйвером.

Запросы драйвера к USB устройству перечислены в пункте 4.1.

На рисунке 4.2. представлена блок-схема, которая демонстрирует работу программы драйвера.

1. Структура программы драйвера устройства



Рис. 4.2 – Структура программы драйвера устройства

1. DLL библиотека драйвера устройства

DLL библиотека предназначена для предоставления стандартного интерфейса для работы с драйвером устройства. DLL библиотека экспортирует функции для:

* получения версии устройства;
* шифрования файла;
* расшифровки файла;
* записи в устройство мастер-ключа;
* считывания номера мастер-ключа, который используется в данный момент;
* записи в устройство номера мастер-ключа, который необходимо использовать в последующих операциях шифрования;
* назначения мастер-ключу символьного имени;

DLL библиотека написана на языке “C++” в программном продукте Microsoft Visual Studio 6.0.

1. Структура программы DLL библиотеки



Рис. 4.3 – Структура программы DLL библиотеки

1. Прикладная программа

Прикладная программа для шифрования файлов позволяет пользователю осуществлять следующие операции:

* шифровать файлы;
* расшифровывать файлы;
* производить операции с мастер-ключами (загружать ключи в устройство, назначать ключу символическое имя, удалять ключи из устройства).

Приложение написано в программе Microsoft Visual Studio 6.0. на языке “С++”, с использованием библиотеки классов MFC.

Алгоритм работы программы приложения показан на рис. 4.4.

1. Структура прикладной программы



Рис. 4.4 – Структура прикладной программы

ВЫВОДЫ

В данном бакалаврском проекте был разработан программно-аппаратный комплекс для шифрования файлов.

При решении задачи был проведен анализ существующих средств шифрования информации и сделан выбор необходимых технических решений. Основным критерием выбора было обеспечение надежной защиты информации.

Для шифрования данных применяется алгоритм Blowfish. Алгоритм устойчив ко всем существующим видам криптоанализа. Ключ, используемый в шифре, имеет переменную длину и его предел равен 448 бит. Это гарантирует устойчивость алгоритма к вскрытию методом грубой силы. Для увеличения криптостойкости системы сеансовые ключи генерируются случайным образом.

Размеры разработанного устройства составляют 52х30х10 мм и соответствуют требованиям технического задания. Для связи с компьютером используется интерфейс USB 2.0. с максимальной скоростью передачи данных 12 Мбит/сек. Питание устройства осуществляется от интерфейса USB. Потребляемый ток составляет 80 мА. Печатный узел устройства разработан с применением технологии поверхностного монтажа.

Для взаимодействия персонального компьютера с устройством был разработан драйвер и оконное приложение. Разработанное программное обеспечение может работать с операционными системами Microsoft Windows 98, NT, 2000, XP.

Конструкторско-технологический, электрический расчеты, расчет теплового режима, расчет на вибропрочность и расчет основных показателей надежности показали соответствие устройства требованиям, изложенным в техническом задании.

В бакалаврском проекте имеется графическая часть, включающая принципиальную схему устройства, сборочный чертеж печатной платы, а так же чертежи разводки печатного узла, которые необходимы для изготовления устройства на производстве.

По разработанной технической документации собран работающий прототип устройства.

Разработанная конструкция может быть усовершенствована, путем увеличения функциональных возможностей. Можно реализовать работу устройства в режиме электронного ключа для авторизации доступа к информации.

Приведенные расчеты и графическая часть показывают, что задание бакалаврского проекта выполнено полностью.

Дальнейшая работа: и отпечатка в контроллер. Это позволит